

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

VAGNER KAEFER DOS SANTOS

**AMBIENTE DE EXECUÇÃO DISTRIBUÍDA UTILIZANDO SISTEMA
DE DIAGNÓSTICO INSTANTÂNEO E CONTÊINERES**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2017**

VAGNER KAEFER DOS SANTOS

**AMBIENTE DE EXECUÇÃO DISTRIBUÍDA UTILIZANDO SISTEMA
DE DIAGNÓSTICO INSTANTÂNEO E CONTÊINERES**

Trabalho de Conclusão de Curso como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação, do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Bruno César Ribas

PATO BRANCO
2017



TERMO DE APROVAÇÃO

Às 20 horas e 20 minutos do dia 28 de junho de 2017, na sala de reuniões do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Bruno Cesar Ribas (orientador), Eden Ricardo Dosciatti e Ives Rene Venturini Pola para avaliar o trabalho de conclusão de curso com o título **Ambiente de Execução Distribuída Utilizando Sistema de Diagnóstico Instantâneo e Contêineres**, do aluno **Vagner Kaefer dos Santos**, matrícula 01376942, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Prof. Bruno Cesar Ribas
Orientador (UTFPR)

Prof. Eden Ricardo Disciatti
(UTFPR)

Prof. Ives Rene Venturini Pola
(UTFPR)

Beatriz Terezinha Borsoi
Coordenador de TCC

Pablo Gauterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Dedico este trabalho a minha mãe, Noeli Maria Kaefer. Por seu incansável apoio perante todas as necessidades e dificuldades que surgiram nestes anos de graduação. Sem ela este trabalho provavelmente nunca teria existido.

AGRADECIMENTOS

A minha querida mãe, Noeli Maria Kaefér e a toda família que, com muito carinho e apoio, não mediram esforços para que eu concluísse a graduação.

Ao professor orientador, Dr. Bruno Cesar Ribas, que além de ter sido um grande professor na graduação, também se mostrou um grande orientador durante o desenvolvimento deste projeto, com uma pontualidade e participação esplêndida em todas as etapas do trabalho.

Ao Vinicius Dalla Corte e ao Ekuikui Vanilson pela parceria e irmandade durante todos os períodos do curso, sem vocês esta caminhada teria sido bem mais complicada.

A todos os professores que fizeram parte nesta jornada de UTFPR, cada disciplina teve seu papel na preparação perante as necessidades que a vida proverá.

Ao Edevino Baldo e a sua esposa Ana Lúcia B. Baldo por todo apoio oferecido. Trabalhar com vocês durante os quatro primeiros anos da graduação foi uma excelente experiência profissional e pessoal.

A todos os amigos da UTFPR, de Pato Branco e de São Lourenço do Oeste, gostaria de poder nomear a todos e agradecer por tudo.

A todos que de qualquer forma fizeram parte deste capítulo da minha vida, minha mais profunda gratidão.

RESUMO

KAEFER, Vagner. Ambiente de execução distribuída utilizando sistema de diagnóstico instantâneo e contêineres. 2017. 107f. Trabalho de Conclusão de Curso de bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

Nas últimas décadas a tecnologia computacional vem mostrando avanços referente *hardware* e *software*. Juntamente com o avanço tecnológico o número de desafios da área também aumentaram, não somente com problemas de processamento mais complexos, mas também com problemas relacionados com o consumo de energia de todos esses equipamentos. Esse trabalho se baseia na utilização de virtualização com contêineres, juntamente com sistemas distribuídos para criar um ambiente de execução de processos genéricos. Sua estrutura busca utilizar, de forma mais efetiva, equipamentos computacionais através das vantagens oferecidas pela virtualização com contêineres e pelo uso de sistemas distribuídos.

Palavras-chaves: sdi. execução distribuída. virtualização. docker. cluster.

ABSTRACT

KAEFER, Vagner. Ambiente de execução distribuída utilizando sistema de diagnóstico instantâneo e contêineres. 2017. 107f. Trabalho de Conclusão de Curso de bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

In last decades, computer technology has been showing advances in hardware and software. Along with technological advancement, the number of challenges in the area also increased, not only with more complex processing problems, but also with problems related to the energy efficiency of all these equipments. This work is based on the use of container virtualization and distributed systems to create a generic process execution environment. Its structure seeks to more effectively use computer equipment through the advantages offered by container virtualization and the use of distributed systems.

Key-words: sdi. distributed execution. docker. virtualization. cluster.

LISTA DE FIGURAS

Figura 1	– Um particionamento de rede.....	17
Figura 2	– Modelo básico de arquitetura para gerenciamento de dados replicados	18
Figura 3	– Camadas de arquivos sobre múltiplos diretórios.....	20
Figura 4	– Diagrama de uma máquina hospedeira executando máquinas virtuais	21
Figura 5	– a) Estrutura de uma máquina tipo I; b) Estrutura de uma máquina tipo II....	23
Figura 6	– Estrutura da virtualização total	24
Figura 7	– Estrutura da virtualização por paravirtualização.....	25
Figura 8	– Máquina virtual completa x Virtualização leve	26
Figura 9	– Modelo de estrutura de um contêiner	27
Figura 10	– Processo de autenticação através de troca de chaves	30
Figura 11	– Diagrama de execução do comando start	34
Figura 12	– Diagrama de execução do comando stop	34
Figura 13	– Diagrama de execução do comando restart	34
Figura 14	– Estrutura de funcionamento de envio e instalação de uma imagem do Docker	41
Figura 15	– Estrutura de funcionamento de recebimento de contêineres para execução.	42
Figura 16	– Estrutura de funcionamento de envio de contêineres para execução	43
Figura 17	– Diagrama do início de uma execução de um contêiner em uma máquina cliente	44
Figura 18	– Diagrama de acompanhamento de execução de contêineres em uma máquina cliente	45
Figura 19	– Estrutura de funcionamento do sistema de controle de execuções	46
Figura 20	– Estrutura de funcionamento do sistema de controle de máquinas livres e vinculação dessas máquinas com os contêineres da fila de espera	47
Figura 21	– Estrutura de funcionamento da varredura de rede e inclusão automática de máquinas	49
Figura 22	– Distribuição problemática de contêineres entre as máquinas	61
Figura 23	– Distribuição de contêineres entre as máquinas após ajustes	62
Figura 24	– Custo do aluguel de máquinas na <i>Digital Ocean</i>	63
Figura 25	– Estrutura utilizada durante o desenvolvimento	64
Figura 26	– Painel de controle geral	64
Figura 27	– Painel de controle de uma máquina	65
Figura 28	– Criação de uma máquina baseada em uma imagem.....	66
Figura 29	– Transferência de rede durante execução do experimento	70
Figura 30	– Distribuição de contêineres entre as máquinas clientes	71
Figura 31	– Utilização do disco durante o experimento	72
Figura 32	– Tráfego de rede gerado durante o experimentos com os pacotes de execução	73
Figura 33	– Distribuição de pacotes de execução entre as máquinas clientes	75
Figura 34	– Utilização do disco durante o experimento	76
Figura 35	– Diferença de utilização de armazenamento as execuções	78
Figura 36	– Diferença de utilização da rede durante as execuções.....	79

LISTA DE LISTAGENS

Listagem 2.1	– Dockerfile para criação de uma imagem	28
Listagem 3.1	– Download do SDI utilizando GIT	31
Listagem 3.2	– Listagem dos arquivos do SDI.....	31
Listagem 3.3	– Comando para instalar pacotes	31
Listagem 3.4	– Comando para criação do par de chaves e configuração com o SSH	32
Listagem 3.5	– Comando ssh-copy-id para instalar chave pública em host remoto	32
Listagem 3.6	– Baixando pacotes e configurando o servidor SSH.....	33
Listagem 3.7	– Desabilitando login através de senha.....	33
Listagem 3.8	– Comando de memória sendo enviado para a máquina 10.132.111.1 e em seguida para todas as máquinas conectadas	35
Listagem 4.1	– Pseudo código da verificação do suporte ao Docker	39
Listagem 4.2	– Pseudo código para envio de uma imagem do Docker	40
Listagem 4.3	– Código BASH para instalação de uma imagem do Docker	40
Listagem 4.4	– Consulta do resultado de execução do contêiner representado com o ticket IPRRoFgeD	48
Listagem 4.5	– Consulta geral sobre as execuções que se encontram em andamento ou já foram finalizadas	48
Listagem 4.6	– Listagem dos arquivos do SDI depois de realizadas as novas alterações	50
Listagem 4.7	– Download da imagem de um repositório oficial, exportação da imagem, envio e instalação em todas as máquinas clientes.....	52
Listagem 4.8	– Listagem de imagens instaladas em uma máquina clientes.....	52
Listagem 4.9	– Programação do arquivo inicia.sh para a criação do contêiner	53
Listagem 4.10	– Programação do arquivo Dockefile para a criação da imagem	53
Listagem 4.11	– Criação da imagem, criação e exportação do contêiner.....	54
Listagem 4.12	– Programação do arquivo command para a criação de um pacote de execução para testes	54
Listagem 4.13	– Criação da imagem, criação e exportação do contêiner.....	55
Listagem 4.14	– Acompanhamento da execução do contêiner de teste	55
Listagem 4.15	– Acompanhamento da execução do contêiner de teste	56
Listagem 4.16	– Informações gerais da execução e conteúdo do arquivo de saída.....	56
Listagem 4.17	– Criação da imagem, criação e exportação do contêiner.....	57
Listagem 4.18	– Acompanhamento da execução do pacote de execução.	57
Listagem 4.19	– Acompanhamento da execução do pacote de execução.	58
Listagem 4.20	– Informações gerais da execução e conteúdo do arquivo de saída.....	58
Listagem 4.21	– Acompanhamento da execução dos contêineres.....	59
Listagem 4.22	– Acompanhamento da execução dos contêineres.....	60
Listagem 4.23	– Envio do contêiner de teste e acompanhamento de sua execução	61
Listagem 5.1	– Dockerfile para a geração da imagem de execução das fórmulas	68
Listagem 5.2	– Programação do arquivo inicia.sh para execução das fórmulas	68
Listagem 5.3	– Programação do arquivo command para a criação dos pacotes de execução	69
Listagem 5.4	– Verificação dos resultados	70
Listagem 5.5	– Verificação dos resultados	75
Listagem 5.6	– Criação do Swap nas máquinas clientes	77

LISTA DE TABELAS

Tabela 1	–	Tempos de execuções disponíveis	36
Tabela 2	–	Estatísticas da execução por contêineres fechados	72
Tabela 3	–	Estatísticas da execução por pacotes de execução	75
Tabela 4	–	Informações da execução por contêineres fechados e por pacotes de execução	77

LISTA DE ABREVIATURAS E SIGLAS

ACM	Association for Computing Machinery
FIFO	First In First Out
GNU	General Public License
ILP	Integer Linear Programming
ICPC	International Collegiate Programming Contest
LXC	Linux Container
NFS	Network File System
OpenVZ	Open Virtuozzo Containeres
OS	Sistema Operacional
PBO	pseudo-Boolean Optimization
SAT	Satisfiability
SDI	Sistema de Diagnóstico Instantâneo
SSH	Secure Shell
TMPFS	Temporary File System
UFPR	Universidade Federal do Paraná
UFS	Union File System
VCS	Sistema de Controle de Versão

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONSIDERAÇÕES INICIAIS	14
1.2	OBJETIVOS	14
1.3	OBJETIVOS ESPECÍFICOS	14
1.4	JUSTIFICATIVA	15
2	REFERENCIAL TEÓRICO	16
2.1	SISTEMAS DISTRIBUÍDOS	16
2.2	UNION FILE SYSTEM	19
2.3	VIRTUALIZAÇÃO COMPLETA	21
2.4	DOCKER	25
3	SISTEMA DE DIAGNÓSTICO INSTANTÂNEO	29
3.1	INSTALAÇÃO	30
3.1.1	Instalação da máquina principal	30
3.1.2	Configuração da máquina principal	31
3.1.3	Instalação e configuração da máquina cliente	32
3.2	UTILIZAÇÃO	33
3.3	FUNCIONAMENTO INTERNO	35
4	DESENVOLVIMENTO	37
4.1	PLANEJAMENTO	37
4.2	EXECUÇÃO	38
4.2.1	Verifica Docker	39
4.2.2	Envio de imagens	39
4.2.3	Recebimento de contêineres e geração de tickets	41
4.2.4	Distribuição de contêineres	43
4.2.5	Controle no cliente	43
4.2.6	Regras de uso	45
4.2.7	Controle de execução e estatísticas	46
4.2.8	Acompanhamento das execuções	47
4.2.9	Varredura de rede	49
4.3	DOCUMENTAÇÃO	49
4.4	DISCUSSÕES DAS IMPLEMENTAÇÕES	52
4.4.1	Envio de imagens	52
4.4.2	Criação do contêiner fechado e do pacote de execução	52
4.4.3	Execução por contêiner fechado	55
4.4.4	Execução por pacote de execução	57
4.4.5	Múltiplas execuções simultâneas	59
4.4.6	Reexecução de contêineres	59
4.4.7	Execução com tempo limite	60
4.4.8	Problema de distribuição	61
5	EXPERIMENTOS	63
5.1	AMBIENTE DE EXECUÇÃO	63
5.2	EXECUÇÃO DE FÓRMULAS	67
5.2.1	Preparação do ambiente	67
5.2.2	Execução por contêineres fechados	69
5.2.3	Execução por pacotes de execução	73
5.2.4	Discussão	76

6 CONCLUSÃO	80
--------------------------	-----------

APÊNDICES	84
APÊNDICE A – EXECUÇÃO POR CONTÊINERES FECHADOS...	85
APÊNDICE B – EXECUÇÃO POR PACOTES DE EXECUÇÃO	88
APÊNDICE C – RESULTADOS DAS FÓRMULAS.....	91
APÊNDICE D – TEMPO DE ESPERA DAS FÓRMULAS.....	100

1 INTRODUÇÃO

O último meio século tem sido um tempo de rápido crescimento da tecnologia, especificamente com relação à computadores (SHIREY et al., 2015). O avanço tecnológico do *hardware* dos computadores atuais possibilitou virtualizar várias máquinas virtuais em um único *hardware*. Segundo Reeser, Jonkowski e Kemper (2015. p 2558-2564), a virtualização de *hardware* consiste na “execução de uma ou mais máquinas virtuais em um computador físico hospedeiro”. Neste ambiente, uma máquina executa várias máquinas em uma camada superior, podendo elas ser de diferentes estruturas e/ou sistemas.

O crescimento na capacidade do *hardware* computacional também gerou a situação em que nem todos os recursos são utilizados em todo momento, sendo que em algumas vezes, chega-se a utilizar somente de 5% a 10% dos recursos disponíveis em data centers (RIBAS et al., 2012). Esta porcentagem de utilização gera desperdício de máquinas e energia, já que estes recursos poderiam ser aproveitados de forma mais eficiente com o auxílio de sistemas distribuídos, que possibilitam que as máquinas trabalhem com um processamento mais próximo dos 100%, levando em consideração que foram projetadas para possuírem uma melhor eficiência energética quando utilizadas na região próxima de 100% (RIBAS et al., 2012).

Um sistema distribuído consiste na utilização de várias máquinas, em conjunto, para realizar a execução de algoritmos que podem ser executados de maneira assíncrona. Ao final de todas as execuções os resultados são associados. Para os usuários e programas aparenta que os mesmos estão utilizando um único sistema (COULOURIS et al., 2013).

Dhakate e Godbole (2015) atestam que, nos dias atuais, alguns serviços como sistemas bancários, financeiros, de telecomunicações, de saúde, científicos, de viagens, de órgãos governamentais, entre outros, exigem servidores com alta capacidade de processamento e armazenamento. Levando em consideração que estas infraestruturas possuem um elevado custo de montagem e manutenção, a virtualização se mostra interessante para utilizar de maneira mais eficiente o *hardware* de servidores. O compartilhamento do *hardware* por várias aplicações pode possibilitar a economia de energia e recursos.

Existem vários meios para tratar cada tipo de demanda computacional, algumas necessitam uma elevada quantidade de processamento e/ou de memória. A junção de sistemas distribuídos e virtualização pode fornecer uma forma de processar os mais diversos tipos de problemas, desde os mais simples até os mais complexos, através da utilização de recursos de várias máquinas.

A união desses recursos requer um sistema que realize sua gerência. Para realizar a comunicação e controle existem alguns sistemas de código aberto, como é o caso do *Sistema de Diagnóstico Instantâneo* (SDI), desenvolvido por Ribas et al. (2009).

1.1 CONSIDERAÇÕES INICIAIS

O SDI proporciona um ambiente de fácil utilização e personalização, permitindo a execução de *scripts* de coleta de informações e verificação com ação preventiva (RIBAS et al., 2009).

A junção do SDI com suporte à virtualização e sistemas distribuídos pode disponibilizar um ambiente de execução com várias máquinas que não se limita com sua estrutura e ambiente, somente sendo necessário que exista o suporte para execução de contêineres.

Como resultado será possível realizar a execução de vários contêineres de maneira ágil e com um controle total do que está acontecendo. O ambiente de execução poderá ser utilizado para tratar os mais diversos tipos de problemas, desde processar dados, realizar simulações e qualquer tarefa que possa necessitar de uma elevada quantidade de processamento e/ou memória.

As máquinas utilizadas pelo sistema não necessitam ser adquiridas, mas somente alugadas pelo tempo de uso necessário. O serviço de Cloud Computing aumentou o nível de eficiência e agilidade em datacenters (VERAS, 2012). Expandindo e remodelando o setor de TI (CARR, 2008). Como as máquinas podem ser alugadas e utilizadas somente pelo período necessário, é gerada uma grande economia de recursos financeiros, já que não será necessário realizar a aquisição de equipamentos, como computadores, servidores e demais dispositivos necessários para a criação de uma infraestrutura, a qual pode facilmente custar milhões de reais somente na etapa de aquisição, ignorando os custos posteriores de manutenção, enquanto alugar uma máquina em uma infraestrutura de Cloud Computing chega a custar menos de um centavo de dólar por hora (OCEAN, 2017).

1.2 OBJETIVOS

O objetivo geral deste trabalho é incrementar o Sistema de Diagnóstico Instantâneo para suportar a execução distribuída de contêineres em diversas máquinas clientes. As máquinas clientes poderão utilizar qualquer *Sistema Operacional* (OS), desde que o mesmo suporte a virtualização de contêineres e seja capaz de realizar comunicação de dados através do protocolo *Secure Shell* (SSH).

1.3 OBJETIVOS ESPECÍFICOS

- Incrementar o sistema de controle na interface do SDI.
- Desenvolver módulo para suporte à contêineres Docker.

- Criar o conjunto de regras para envio de requisições e admissão de respostas.
- Desenvolver uma forma escalável de transmissão de contêineres na rede.
- Disponibilizar opções de garantia de integridade das execuções.
- Utilizar o sistema para executar um problema de teste genérico.

1.4 JUSTIFICATIVA

O Docker (2016b) é um sistema que utiliza virtualização através de containerização, também conhecido como virtualização leve.

A virtualização leve pode ser vista como uma máquina virtual, que contém um sistema de arquivos completo, além de todos os arquivos necessário para que o contêiner possa ser executado, como: códigos, ferramentas de sistema, bibliotecas de sistema, ou qualquer *software* que possa ser instalado no sistema operacional Linux. Desta forma, o contêiner pode ser executado em qualquer máquina que possua suporte a containerização, independentemente de sua estrutura. Para Sallou e Monjeaud (2015), essa tecnologia chama a atenção, pois oferece desempenho e escalabilidade eficaz tanto na nuvem quanto na arquitetura física.

A utilização do Docker, juntamente com o sistema de gerenciamento do SDI, disponibiliza um ambiente de execução distribuído entre várias estações, utilizando os recursos das mesmas para a solução dos mais diversos problemas que possam ser executados separadamente de forma assíncrona, além de aproveitar todas as vantagens que a virtualização de contêineres disponibiliza.

O trabalho foi organizado através dos seguintes capítulos: o segundo capítulo foi destinado à fornecer o referencial teórico necessário para entendimento deste trabalho; o terceiro capítulo explana de maneira detalhada o funcionamento e estrutura interna do SDI; o quarto capítulo aborda o planejamento e execução do desenvolvimento do novo sistema; o quinto capítulo descreve os experimentos realizados com o sistema desenvolvido; o último capítulo discute os resultados obtidos durante o trabalho, juntamente com as possibilidades de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento teórico sobre as tecnologias e ferramentas que fazem parte deste trabalho.

2.1 SISTEMAS DISTRIBUÍDOS

Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente (TANENBAUM; STEEN, 2007). Os componentes são localizados em computadores interligados em redes e se comunicam e coordenam suas ações através da troca de mensagens. Dessa forma as máquinas podem estar na mesma sala, no mesmo prédio ou em outro continente.

Coulouris et al. (2013) comentam que os sistemas distribuídos fornecem um alto grau de disponibilidade perante falhas de *hardware* ou *software*. A disponibilidade de um sistema é a medida da proporção de tempo em que ele está pronto para uso, quando uma máquina falha, apenas o trabalho que essa máquina estava realizando sofre uma perda, porém, o sistema pode detectar a falha e transferir o trabalho da máquina defeituosa para outra máquina que esteja disponível, não comprometendo o funcionamento completo do sistema.

Conforme definido por Coulouris et al. (2013), os sistemas distribuídos podem ser classificados como assíncronos ou síncronos. Em um sistema assíncrono, não existe nenhuma suposição referente a temporização da transmissão de mensagens, diferentemente dos síncronos, onde é possível supor limites de tempo máximo de transmissão. Esse teto máximo é baseado no tempo necessário para que uma tarefa seja executada. Sabendo-se do tempo limite de uma transmissão, pode-se utilizar desta informação para detectar possíveis falhas de processo.

Segundo Chandra e Toueg (1996), um detector de falha é um serviço que os processos consultam para saber se um processo em particular falhou. Ele pode ser implementado através de um processo paralelo, o qual executa um algoritmo de detecção de falha em conjunto com seus outros processos correlatados.

Um objetivo importante é considerar falhas e projetar algoritmos para lidar com elas. Problemas de rede podem deixar um ou vários nós inacessíveis. A Figura 1 mostra um exemplo onde um roteador danificado isolou parte da rede de nós. O sistema deve ser capaz de tratar esses erros e manter sua execução.

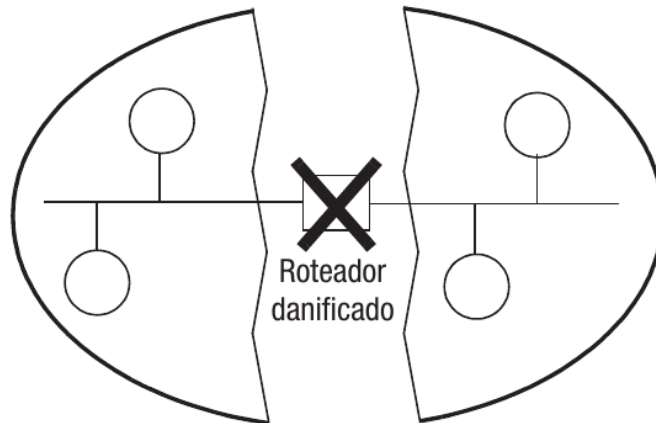


Figura 1 – Um particionamento de rede

Fonte: Adaptado de Coulouris et al. (2013)

Coulouris et al. (2013) comenta duas questões importantes em relação a tolerância a falhas: O que acontece quando mensagens são perdidas e o que acontece quando um processo falha. O detector de falhas deve identificar essas situações e usar algoritmos para tratá-las.

Segundo Coulouris et al. (2013), a replicação é a chave para se prover alta disponibilidade e tolerância a falhas em sistemas distribuídos. Ela é uma técnica para melhorar os serviços, aumentando seu desempenho, sua disponibilidade e/ou tornando-o tolerante a falhas. Existem três motivações para se fazer uso da replicação:

- **Melhoria de desempenho**

Criar caches dos dados é uma maneira conhecida de melhorar o desempenho. Por exemplo, navegadores e servidores *proxies* colocam no cache cópias de recursos para evitar a latência de busca desses recursos diretamente no servidor de origem (COULOURIS et al., 2013).

- **Maior disponibilidade**

Usuários de sistemas exigem que a proporção do tempo que um serviço está acessível deve ser próxima a 100% (COULOURIS et al., 2013). A replicação de dados em várias máquinas possibilita que o sistema continue operando normalmente, mesmo com a situação de falha de uma ou mais destas máquinas.

- **Tolerância a falhas**

Dados de alta disponibilidade podem não ser rigorosamente corretos. Podem estar, por exemplo, desatualizados. Na situação em que dois usuários isolados por um problema de rede realizem alterações em um mesmo arquivo, é gerado um conflito que necessita ser corrigido.

Coulouris et al. (2013) atentam que um requisito comum quando os dados são replicados é a transparência da replicação, ou seja, os usuários não devem saber que existem cópias físicas dos dados, para eles, os dados são organizados como objetos lógicos individuais e identificam apenas um item em cada caso, quando solicitam a execução de uma operação.

Além da transparência, a consistência das informações não pode ser deixada de lado. As operações executadas sobre um conjunto de objetos replicados devem produzir resultados que satisfaçam as especificações das correções desses objetos.

Os dados de um sistema consistem em objetos, um objeto pode ser um arquivo por exemplo. Porém, cada objeto lógico é implementado por um conjunto de cópias físicas, denominadas réplicas (COULOURIS et al., 2013).

A Figura 2 demonstra um modelo básico de arquitetura para o gerenciamento de dados replicados.

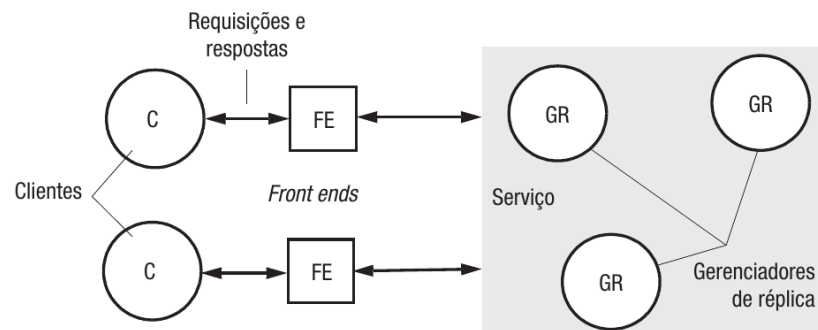


Figura 2 – Modelo básico de arquitetura para gerenciamento de dados replicados
 Fonte: (COULOURIS et al., 2013)

Os processos indicados na Figura 2 não implicam em uma implementação por processos (ou *hardware*) distintos. Uma máquina pode realizar uma ou mais etapas do processo geral.

Um gerenciador de réplicas aplica operações em suas réplicas de maneira recuperável, ou seja, caso ocorra alguma falha em um processo, sua operação pode ser desfeita, não deixando os dados inconsistentes.

Segundo Zampieri (2001), existem fundamentalmente dois tipos de gerenciadores de réplicas, os gerenciadores dinâmicos e os estáticos. Os membros de um grupo estático não são alterados durante a vida do sistema, mesmo com membros que se encontram em estado de falha, enquanto que nos gerenciadores dinâmicos ocorrem mudanças no seu conjunto de membros,

quando um membro apresenta falhas, o mesmo é removido do grupo até que seja normalizado novamente.

As requisições dos clientes são gerenciadas por um componente chamado *front-end*, sua função é se comunicar, por passagem de mensagens, com um ou mais gerenciadores de réplica.

2.2 UNION FILE SYSTEM

Tanenbaum (2010) cita que todos as aplicações precisam armazenar e recuperar informações. Os arquivos são gerenciados pelo sistema operacional, o qual define como a informação é estruturada, nomeada, acessada, usada, protegida e implementada. A parte do sistema operacional que trata dos arquivos é conhecida como sistema de arquivos.

Conforme Gupta e Saini (2015), o sistema de arquivos é utilizado para armazenar e organizar arquivos de forma permanente, para que possam ser encontrados e acessados com facilidade, representando uma das partes mais importantes de um sistema operacional (HEIDEMANN; POPEK, 1994).

O *Union File System* (UFS) é um sistema de arquivos empilhável que funde um conjunto de diretórios (chamados de ramos) em uma única visão lógica, mesmo que cada ramo se origine de diferentes tipos de sistemas de arquivos. Cada ramo possui uma atribuição de prioridade, além de poder ser configurado como somente leitura ou de leitura e escrita (QUIGLEY et al., 2006).

Um arquivo pode existir em vários ramos, porém, o usuário visualiza apenas a cópia no ramo de prioridade mais alta. O UFS permite que alguns ramos sejam somente de leitura, enquanto ramos de maior prioridade são de leitura e escrita, utilizando a semântica de cópia em gravação para fornecer a ilusão de que todos os ramos são graváveis (QUIGLEY et al., 2006).

A partir da raiz principal, o UFS cria camadas de montagens através de uma pilha. Todo acesso ao armazenamento se inicia nas camadas mais elevadas e segue para as camadas inferiores, caso um arquivo solicitado não seja encontrado na primeira camada, a busca segue para as próximas camadas.

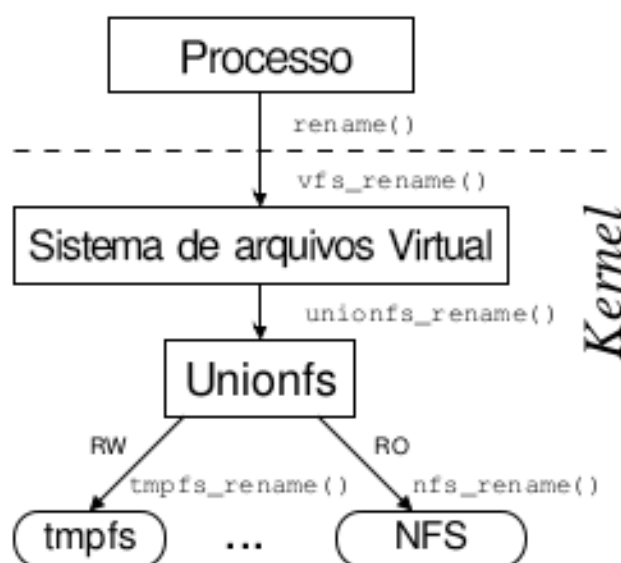


Figura 3 – Camadas de arquivos sobre múltiplos diretórios

Fonte: Adaptado de Quigley et al. (2006)

A Figura 3 demonstra uma montagem de diretórios realizada pelo UFS, o conteúdo do sistema de arquivos *Temporary File System* (TMPFS) e *Network File System* (NFS) são virtualmente mesclados em um único sistema de arquivos. O sistema NFS está configurado como somente leitura e o sistema TMPFS está definido como leitura e escrita. Todas as instruções de escrita destinadas ao sistema NFS retornarão um erro de permissão, logo o próprio UFS é responsável por utilizar a camada do TMPFS para realizar a instrução. O usuário não tem conhecimento que seu arquivo foi salvo na camada TMPFS, pois o UFS realiza esta operação de forma transparente.

O UFS possui esquemas complexos de alocação e faz mais uso de *kmalloc* do que outros sistemas de arquivos, mas o uso do UFS simplifica a administração de máquinas sem armazenamento. Um conjunto de máquinas pode compartilhar um único sistema de arquivos raiz, através de um sistema NFS configurado como somente leitura, permitindo que os administradores mantenham uma imagem comum para todas as máquinas. Esse sistema de arquivos raiz é então unificado com um ramo que possui permissão de escrita, este novo ramo é configurado como sendo de maior prioridade, assim os usuários podem personalizar sua máquina e/ou salvar seus dados de forma transparente, sem alterar nenhum dado do sistema de arquivos raiz. Se a persistência de dados não for necessária, o sistema de arquivos TMPFS pode ser usado como o ramo de maior prioridade. Caso a persistência seja necessária, a montagem de um armazenamento local podem ser usada para salvar os arquivos do usuário (QUIGLEY et al., 2006).

O UFS também pode fornecer semântica de cópia durante a escrita, marcando logicamente um ramo de leitura e escrita como somente leitura. Isso permite que o UFS possa ser utilizado como um sistema de arquivos que fornece imagens de sistemas (QUIGLEY et al., 2006).

2.3 VIRTUALIZAÇÃO COMPLETA

Esta seção do trabalho é baseada no trabalho de Laureano (2006). O conceito de máquinas virtuais surgiu no início de 1960 buscando centralizar os sistemas de computador utilizados pela IBM, onde cada máquina virtual simulava uma réplica de uma máquina física, desta forma cada usuário tinha a impressão de possuir um sistema disponível exclusivamente para ele. Sua utilização se tornou uma alternativa para vários sistemas de computação, por possuir várias vantagens em relação a custos, portabilidade e segurança .

Os computadores atuais são formados por praticamente três componentes: o *hardware* físico, o sistema operacional que gerencia e controla o *hardware* e por fim, suas aplicações. Como pode ser visto na Figura 4, o sistema operacional é responsável por gerenciar o acesso ao *hardware*, principalmente quando existe o compartilhamento do dispositivo entre mais de uma aplicação. Com o passar do tempo surgiram vários tipos de *hardwares* e sistemas operacionais, cada um criado através de projetos independentes, resultando em uma grande variedade de plataformas que não possuem compatibilidade entre si. A utilização de uma máquina virtual se mostra como uma solução para este problema, criando uma camada que permite a compatibilização de aplicações entre plataformas.

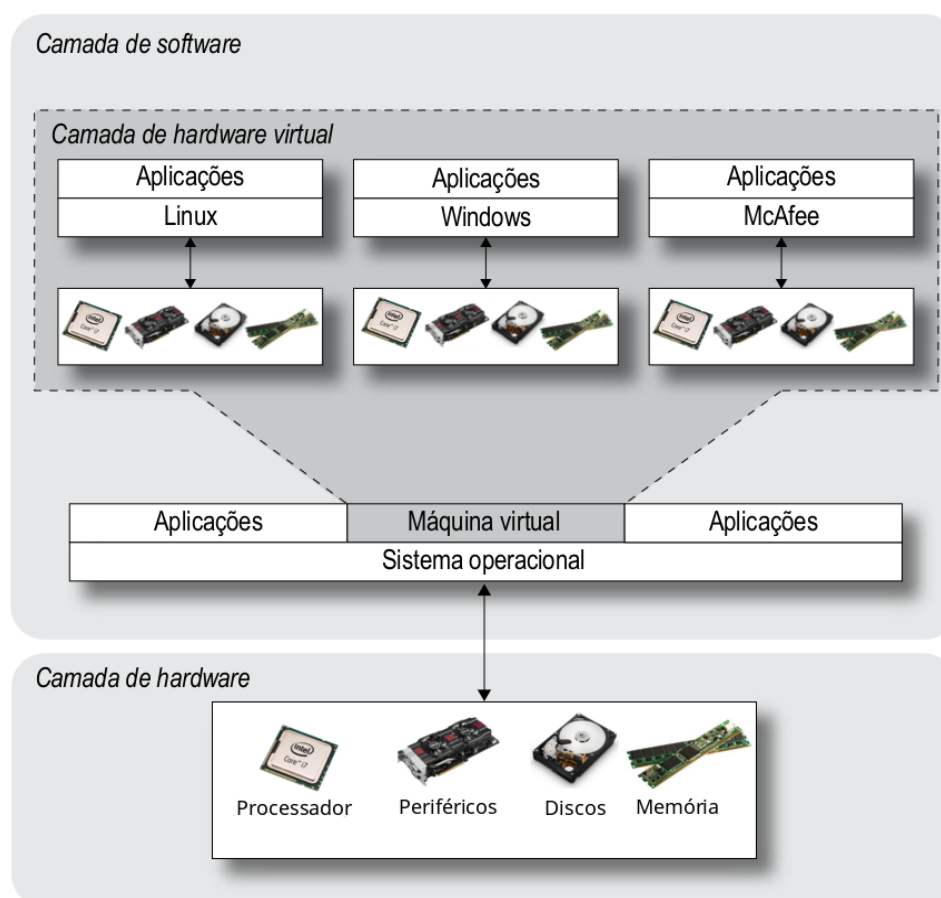


Figura 4 – Diagrama de uma máquina hospedeira executando máquinas virtuais
 Fonte: Adaptado de Laureano (2006)

Segundo Laureano (2006), uma máquina real é formada por um *hardware*, que fornece a plataforma para o funcionamento de um sistema operacional e suas aplicações através de um determinado conjunto de instruções e outros elementos fundamentais para o processamento de dados, alocação de memória e gerenciamento de entrada e saída de dados. Já a máquina virtual cria uma camada de *hardware* virtual dentro da camada de *software* da máquina hospedeira, implementando todas as instruções realizadas por uma máquina real em um ambiente totalmente abstrato de *software*. Essa virtualização é possível através de monitor de máquina virtual (*hypervisor*), o monitor de máquina virtual pode criar N máquinas virtuais em uma máquina virtual, desde que seu *hardware* suporte todas estas aplicações.

O monitor de máquina virtual não consiste em um simples emulador, um emulador é um *software* que simula um computador real com uma camada de abstração completa entre o sistema em execução e o *hardware*, traduzindo instruções de um processador virtual para o processador real, infelizmente essa transcrição gera um custo elevado em termos de eficiência, já que toda instrução deve ser traduzida, além disso, emuladores são *softwares* complexos, pois necessitam simular todas as instruções do processador e demais características do *hardware* emulado. Já o monitor de máquina virtual proporciona uma interface através da multiplexação do *hardware*, fornecendo uma cópia virtual do computador hospedeiro para cada máquina virtual, fazendo com que o sistema virtualizado acredite estar sendo executado em um ambiente com acesso total e direto ao *hardware*.

Existem dois tipos de máquinas virtuais, o sistema monitor das máquinas virtuais do tipo I trabalha diretamente entre o *hardware* e os sistemas convidados, possuindo total controle sobre o *hardware* e proporcionando um ambiente para cada máquina virtual, as quais se comportam como uma máquina física completa que pode executar seu próprio sistema operacional, enquanto nas máquinas virtuais tipo II o monitor é implementado como um processo no sistema operacional hospedeiro. O monitor é executado por um sistema operacional anfitrião. Seu funcionamento é semelhante a virtualização tipo I, porém, sua maior diferença consiste na existência de um sistema operacional abaixo do *hypervisor*. Buscando um melhor desempenho, ambos os tipos de monitores permitem que uma máquina virtual faça o uso direto do *hardware*, desde que esta ação não altere o correto funcionamento do sistema e das demais máquinas virtuais. A Figura 5 demonstra a diferença entre os dois tipos de máquinas virtuais.

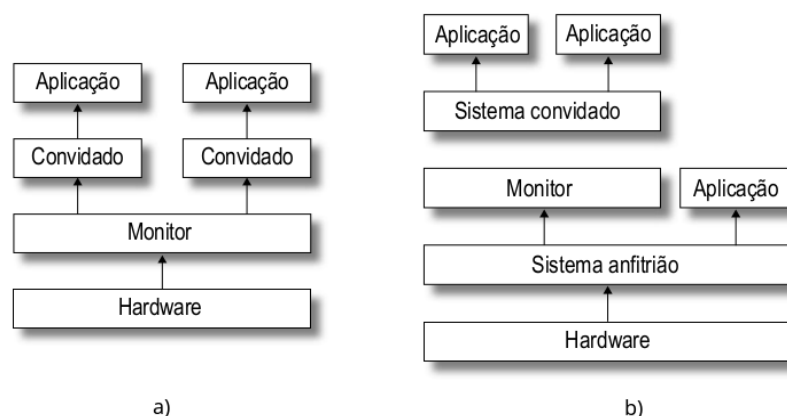


Figura 5 – a) Estrutura de uma máquina tipo I; b) Estrutura de uma máquina tipo II

Fonte: Adaptado de Laureano (2006)

Dessa forma, o uso das máquinas virtuais e emuladores possibilitam: a execução de um sistema operacional e suas aplicações sobre outro; a utilização de aplicações de diversas plataformas em uma única máquina; a execução de múltiplos sistemas operacionais ao mesmo tempo e flexibilizam a criação de um ambiente de trabalho complexo.

Laureano (2006) comenta que os monitores de máquina virtual possuem algumas propriedades de segurança e outras aplicações, como:

- **Isolamento:** Esta é uma das principais propriedades de uma máquina virtual, ela garante que um *software* em execução não acesse nem modifique qualquer informação do *hypervisor* ou de outra máquina virtual. Essa propriedade possibilita que erros e atividades maliciosas fiquem contidos dentro da máquina virtual, além disso, o isolamento possibilita a limitação do uso de recursos de uma máquina virtual, evitando que o excesso de uso de uma máquina prejudique o desempenho de outras máquinas virtuais.
- **Inspeção:** O *hypervisor* tem acesso e controle total sobre todas as máquinas virtualizadas, possibilitando o acompanhamento do estado da CPU, do conteúdo da memória e de todos os recursos de *hardware* disponíveis.
- **Interposição:** O monitor de privilégio pode intercalar ou acrescentar instruções em operações da máquina virtual.
- **Eficiência:** Quando possível, operações podem ser executadas diretamente no *hardware* físico, desde que não afete as demais máquinas virtualizadas e/ou aplicações.
- **Gerenciabilidade:** A partir do monitor de máquina virtual é possível gerenciar e administrar todas as instâncias de forma simplificada e centralizada.
- **Compatibilidade do software:** A abstração da máquina virtual pode mascarar diferenças nas camadas do *hardware* e do *software* abaixo da máquina virtual. Um exemplo desta

propriedade é a máquina virtual Java, onde um *software* em Java pode ser executado em qualquer outra máquina virtual Java.

- **Desempenho:** A camada de *software* do *hypervisor* em um sistema, afeta o desempenho do *software* que funciona na máquina virtual, porém, os benefícios proporcionados pelo uso da virtualização compensam esta perda de desempenho.

Segundo Laureano (2006) as técnicas mais utilizadas para virtualização são: paravirtualização (*paravirtualization*) e virtualização total (*full virtualization*).

Na virtualização total, o sistema convidado não precisa sofrer qualquer tipo de alteração, pois a estrutura completa do *hardware* é virtualizada. Sendo esta a sua principal vantagem, porém, o sistema virtualizado é executado de forma mais lenta e o *hypervisor* precisa trabalhar para que operações privilegiadas possam ser executadas em processadores que não suportem a virtualização de forma nativa. A Figura 6 exibe a representação da virtualização total.

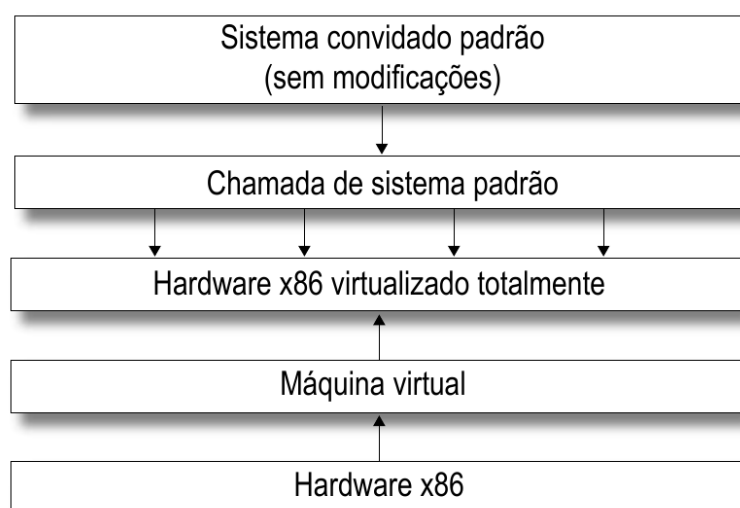


Figura 6 – Estrutura da virtualização total

Fonte: Laureano (2006)

Na paravirtualização o sistema convidado passa por alterações, visando uma maior interação com o monitor de máquinas virtuais, resultando em uma melhor eficiência. Estas alterações diminuem a portabilidade do sistema, porém, permitem que recursos do *hardware* sejam acessados diretamente. O trabalho de adaptação do sistema convidado é compensado com a melhora de performance obtida. A Figura 7 mostra a representação da virtualização por paravirtualização.

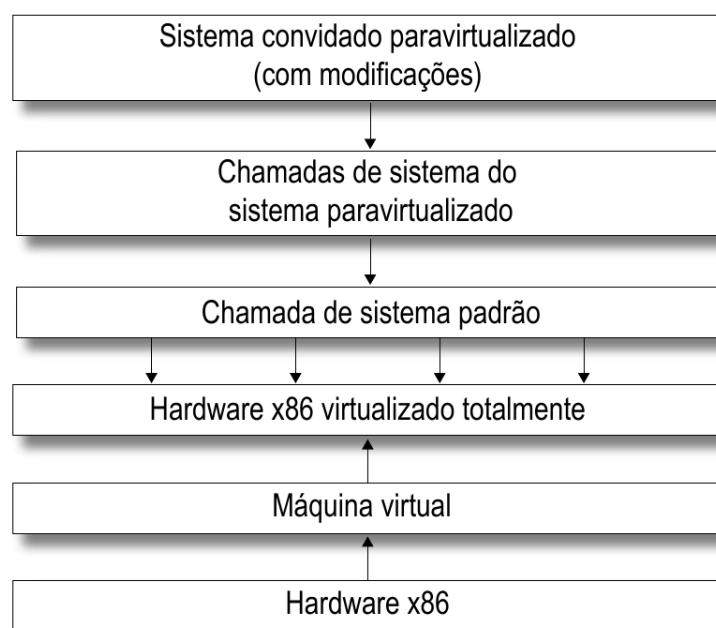


Figura 7 – Estrutura da virtualização por paravirtualização
Fonte: Laureano (2006)

2.4 DOCKER

Uma nova geração de virtualização minimiza pontos negativos da virtualização completa: A virtualização em contêineres, ela é baseada na execução de contêineres e fornece desempenho próximo de aplicações nativas não virtualizadas, pois funciona no mesmo nível do sistema operacional hospedeiro (VASCONCELOS; FREITAS, 2014). Segundo Stubbs, Moreira e Dooley (2015), enquanto máquinas virtuais fornecem uma abstração em nível de *hardware*, o modelo de virtualização em contêineres virtualiza as chamadas do sistema operacional, possibilitando que muitos contêineres compartilhem a mesma instância do *kernel* do sistema operacional hospedeiro.

Docker (2016c) comenta que geralmente as aplicações são construídas usando três tipos de infraestrutura: computação, rede e armazenamento. A virtualização em contêineres através do Docker fornece um meio de abstrair todos estes componentes, aumentando a agilidade e o controle das aplicações. Como resultado, é possível criar, testar e implementar aplicações rapidamente (SERVICES, 2017).

O *Open Virtuozzo Containers* (OpenVZ) é uma tecnologia de virtualização baseada em contêineres que funciona em nível de sistema operacional (OPENVZ, 2016). O OpenVZ é um dos sistemas de contêineres mais utilizados em ambientes de produção, possuindo uma versão estável. Stubbs, Moreira e Dooley (2015) comentam que os contêineres são atrativos, pois fornecem isolamento completa de sistemas independentes que rodam em um ambiente compartilhado. Este isolamento fornece segurança e auxilia na mitigação de ataques tradicionais através

da isolação da rede, do sistema de arquivos e pelo monitorando do consumo de recursos de cada contêiner.

O *Linux Container (LXC)* é uma interface que torna o sistema operacional Linux compatível para implantação de um ou mais contêineres através do compartilhamento do próprio *kernel* (LXC, 2016). Diferentemente do OpenVZ, o LXC não necessita de modificações do kernel do sistema hospedeiro, pois são utilizados os *namespaces* do próprio sistema para tornar disponível a virtualização em contêineres. O LXC ainda está em fase de desenvolvimento, mas já é utilizado como *back-end* nas versões mais recentes do Docker (RIZKI; RAKHMATSYAH; NUGROHO, 2016).

O sistema Docker compartilha suas bibliotecas com diferentes aplicações e utiliza o sistema de camadas oferecido pelo UFS, possibilitando utilizar um único sistema operacional e compartilhar seus recursos entre diversas máquinas. Caso um contêiner necessite de alguma dependência específica, ela estará instalada diretamente no próprio contêiner. O isolamento e a segurança fornecidos pelo Docker permitem que muitos contêineres sejam executados simultaneamente em um único *hardware* hospedeiro (PREETH et al., 2015).

A Figura 8 mostra que o Docker elimina o *hypervisor* das máquinas virtuais e o sistema operacional virtualizado de cada instância.

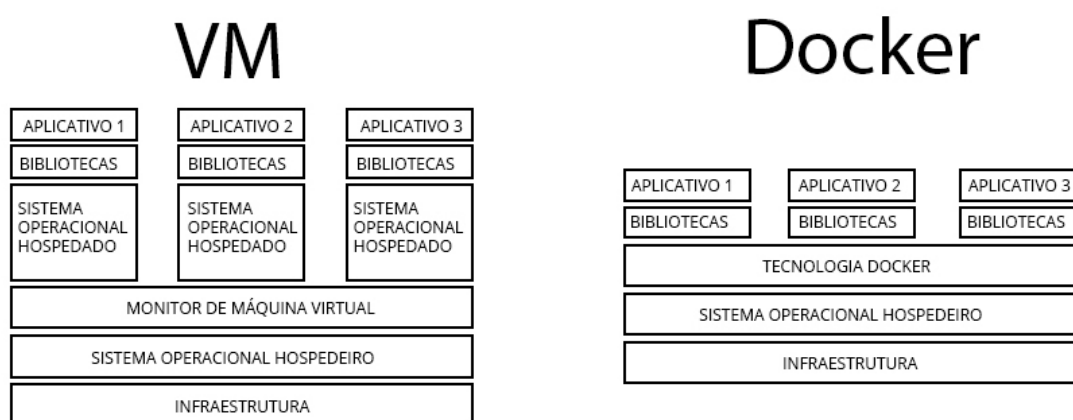


Figura 8 – Máquina virtual completa x Virtualização leve

Fonte: Adaptado de Docker (2016b)

Dhakate e Godbole (2015) demonstraram que, como o sistema operacional é compartilhado com os contêineres, a inicialização de um contêiner torna-se muito leve e leva apenas alguns segundos. Compartilhar este contêiner entre máquinas também pode levar apenas alguns segundos, enquanto a clonagem de uma máquina virtual completa pode levar vários minutos.

Os desenvolvedores podem criar e replicar seus ambientes de produção e comandar ações para vários contêineres em questão de segundos (DHAKATE; GODBOLE, 2015). Para auxiliar nessa replicação e armazenamento, utilizasse o *Docker Registry*.

Segundo Docker (2016a), o *Docker Registry* é um aplicativo de servidor sem escalonamento, altamente escalável que armazena e permite distribuir imagens do Docker. O *Docker*

Registry está disponível através de código aberto, sob a licença permissiva do *Apache*.

Sua utilização pode ser feita pelo serviço oferecido pelo próprio Docker, chamado de *Docker Hub*, o qual atende as necessidades para a transmissão de imagens entre várias máquinas, desde que estas possuam conexão via Internet.

Caso o usuário tenha o desejo de criar seu próprio distribuidor de imagens, independentemente do motivo, ele necessita somente instalar o serviço na máquina que será utilizada como servidor. Desta forma, várias máquinas podem se conectar e baixar imagens necessárias.

A transmissão de contêineres tem um bom desempenho por conseguir visualizar as camadas do contêiner e realizar a transmissão somente de partes específicas. A Figura 9 mostra um modelo de estrutura de um contêiner.

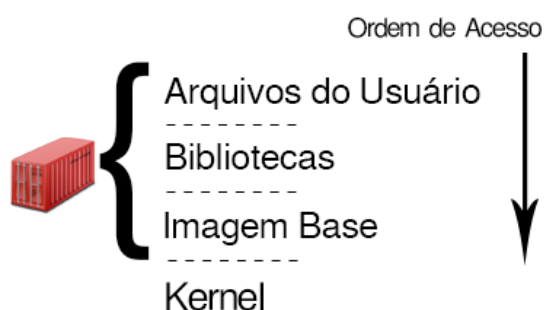


Figura 9 – Modelo de estrutura de um contêiner

Fonte: Autoria própria

Na necessidade da transferência de um contêiner para outra máquina, a virtualização leve verifica qual será a imagem base do contêiner, suas bibliotecas e arquivos do usuário. Caso a máquina de destino já possua a imagem base desejada e suas bibliotecas, somente será necessário enviar os arquivos do usuário. Desta forma, uma transferência de dados pode ter um tamanho reduzido. Esta característica oferecida pelo UFS que utiliza arquivos em camadas é a chave para a transmissão de contêineres realizada pela virtualização leve.

O Docker também disponibiliza ferramentas para auxiliar na criação de imagens. Uma imagem pode ser visualizada como um sistema operacional, na imagem é definido quais aplicativos e bibliotecas estarão disponíveis para utilização. Para criar uma imagem é utilizado um arquivo *Dockerfile* que contém todas as instruções para geração. A Listagem 2.1 exibe um exemplo de utilização do *Dockerfile*.

```
# informa que a imagem será baseada na última versão da distribuição Ubuntu
FROM ubuntu:latest

# executa um comando no container
# neste caso instala o programa clasp
RUN apt-get update && apt-get install -y apt-utils clasp

# copia um arquivo para dentro do contêiner
COPY arquivo_exemplo.txt /home/arquivo.txt

# informa que programa deve ser inicializado com o contêiner
CMD /bin/bash executa.sh
```

Listagem 2.1 – Dockerfile para criação de uma imagem

O Dockerfile irá criar uma imagem baseada na distribuição Ubuntu e irá instalar o aplicativo clasp e suas dependências. Todos os contêineres criados através dessa imagem poderão utilizar o aplicativo e todas as demais funcionalidades da distribuição Ubuntu. As imagens e contêineres são identificados por um código de identificação e por um nome, também definido como *tag*.

Os contêineres são executados de forma isolada do sistema operacional hospedeiro, porém, é possível realizar o compartilhamento de pastas entre a máquina hospedeira e o contêiner, este compartilhamento é chamado de *Volume*. Um *Volume* consiste basicamente de um diretório compartilhado, facilitando a transferência de informações e oferecendo persistência de dados, uma vez que não há risco dos dados serem deletados quando um contêiner terminar sua execução e/ou for removido. Porém, existem pontos desfavoráveis em sua utilização, principalmente em relação à segurança em nível de arquivos do sistema, pois o contêiner possuirá acesso físico ao armazenamento, impossibilitando o controle dos comandos que serão executados neste ambiente. Um *Volume* também pode ser utilizado para realizar o compartilhamento de arquivos entre vários contêineres.

3 SISTEMA DE DIAGNÓSTICO INSTANTÂNEO

O *Sistema de Diagnóstico Instantâneo* (SDI) (RIBAS et al., 2009) é um sistema prático e escalável para diagnosticar e gerenciar redes de forma instantânea, independentemente do seu porte. Ele fornece uma maneira de executar scripts de diagnósticos e comandos em máquinas remotas. O sistema foi inicialmente desenvolvido utilizando a linguagem *Bash*. Com o desenvolvimento de novas funcionalidades, as novas versões receberam implementações de scripts escritos na linguagens *Python* e *C*, sendo essas duas linguagens necessárias para o correto funcionamento do sistema original. O SDI é um projeto de código aberto disponibilizado sobre a *General Public License v2* (GNU).

O SDI é utilizado pela Secretaria de Educação do Paraná (SEED, 2017), informando dados relevantes para a gestão dos laboratórios das escolas do estado, como por exemplo: problemas de armazenamento; problemas de energia; informações de uso de memória; informações de processamento e tempo de utilização. Permitindo uma gerência completa de mais de 2000 escolas do projeto Paraná Digital, facilitando a fiscalização, controle e prevenção de problemas.

A comunicação do sistema é realizada de forma criptografada, através do protocolo *Secure Shell* (SSH). Conforme definido na norma técnica RFC4251 de Ylonen (2006), o SSH é um protocolo que utiliza criptografia para possibilitar a comunicação com máquinas remotas de maneira segura, mesmo que seu tráfego seja realizado em redes que não sejam seguras e/ou confiáveis. Segundo a norma técnica RFC4344 de Bellare, Kohno e Namprempre (2006), o SSH foi concebido para fornecer privacidade e integridade na transmissão de dados.

A autenticação entre máquinas faz uso da troca de chaves criptográficas, que são desenvolvidas com base em cálculos matemáticos e garantem que a requisição está sendo realizada por uma máquina autorizada. Neste processo, existem duas chaves, uma pública que pode ser distribuída (compartilhada sem restrições) e uma chave privada, que deve ser mantida em segredo, somente o proprietário da chave privada pode ter acesso a mesma.

Uma mensagem criptografada com uma chave pública somente pode ser descriptografada através de sua chave privada, dessa forma, qualquer remetente consegue criptografar informações através da chave pública do destinatário, garantindo que somente o destinatário poderá ter acesso aos dados (MORENO; PEREIRA; CHIARAMONTE, 2005). Em suma, a chave pública é utilizada para criptografar a informação e a chave privada é responsável pela descriptografia.

Para realizar o acesso a uma máquina esse conceito é utilizado. O usuário solicita acesso a uma máquina e informa sua identificação pública, a máquina irá utilizar a chave pública do solicitante para gerar uma informação criptografada, em seguida essa mensagem deve ser descriptografada pelo usuário. Caso o usuário esteja de posse da chave privada ele será capaz de realizar a descriptografia e devolver o conteúdo para verificação. Caso a informação recebida seja igual a gerada garante-se que o solicitante realmente é autêntico, liberando a conexão e o

acesso para utilização. A Figura 10 mostra esse processo de autenticação:

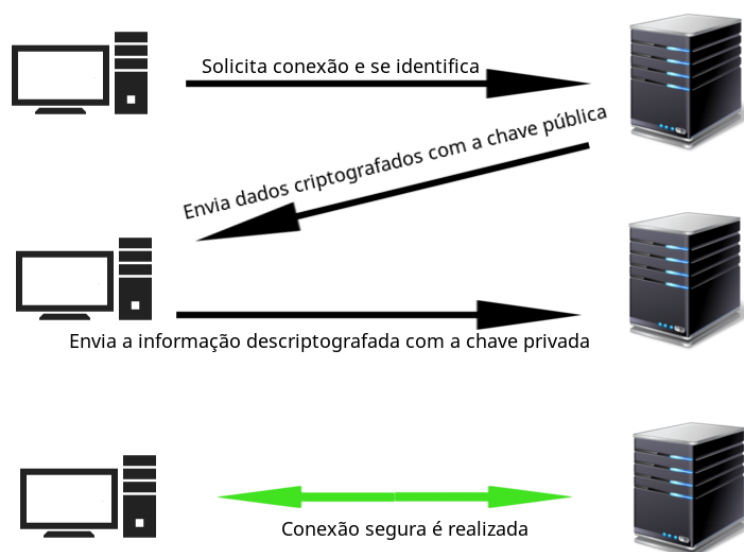


Figura 10 – Processo de autenticação através de troca de chaves

Fonte: Adaptado de MORENO, PEREIRA e CHIARAMONTE (2005)

3.1 INSTALAÇÃO

A seguir estão descritos os passos para instalar e configurar o SDI.

3.1.1 Instalação da máquina principal

Para realizar a instalação do sistema de forma prática é necessário realizar a instalação de alguns aplicativos e suas dependências. Em todos os exemplos e comandos apresentados a seguir foram utilizadas máquinas executando o sistema operacional Ubuntu, versão 16.04.2, x64 bits. Porém, o SDI pode ser instalado e executado em qualquer distribuição Linux.

Como pasta de instalação do sistema será utilizada a pasta `/root/sdi`. Depois de criada a pasta pode-se utilizar o gerenciador Git para baixar o sistema dos repositórios oficiais utilizando os comandos disponíveis na Listagem 3.1. Serão clonados também o histórico de alterações e todas as versões desenvolvidas do sistema, possibilitando que, caso o usuário deseje, possa realizar alterações e/ou incrementos no sistema.

```

root@sdi:~# mkdir /root/sdi
root@sdi:~# cd /root/sdi
root@sdi:~# git clone https://github.com/sdi-ng/sdi.git

```

Listagem 3.1 – Download do SDI utilizando GIT

Dentro dessa pasta criada estão todos os arquivos necessários para utilizar o SDI. A Listagem 3.2 mostra todos os arquivos que estão dentro da pasta.

```

root@sdi:/root/sdi/sdi# ls -F
configsdiparser.py*      summaries-available/    launchsdi.sh*
parser.sh*               sdictl*                 socketclient.c*
states-enabled/         summaries-enabled/     commands-available/
sdi.conf*               launchsditunnel.sh*    misc.sh*
socketdaemon.py*        states.sh*              sync.sh*
commands-enabled/       launchscripts.sh*      launchsendfile.sh*
modules/                sendfile.sh*            states-available/
sdiweb/

```

Listagem 3.2 – Listagem dos arquivos do SDI

Essa primeira etapa não necessita de mais passos, como o sistema utiliza a linguagem *Bash*, não é necessário realizar nenhuma compilação, em seguida verifica-se se o sistema possui todos os pré requisitos e realiza-se algumas configurações.

3.1.2 Configuração da máquina principal

Os seguintes aplicativos devem estar instalados no sistema para um correto funcionamento do SDI:

- **ssh**: Necessário para realizar a comunicação entre as máquinas.
- **bc**: Necessário para realizar alguns cálculos de precisão.
- **cron**: Necessário para realizar o agendamento de tarefas em relação ao tempo.
- **apache2**: Esse programa é opcional, sua instalação é necessária para o funcionamento do *sdiweb*, o qual não será utilizado nem explicado neste trabalho.

Caso algum programa não esteja instalado o comando exibido na Listagem 3.3 pode ser utilizado para instalar todos os pacotes:

```

root@sdi:~# apt-get install -y ssh bc cron apache2

```

Listagem 3.3 – Comando para instalar pacotes

Para realizar a autenticação nas máquinas remotas é necessária a realização da troca de chaves conforme apresentado anteriormente. Após a criação de um par de chaves na máquina do SDI, a chave pública deve ser inserida nas máquinas remotas. Os comandos descritos na Listagem 3.4 irão criar um par de chaves e deixá-las configuradas para utilização com o SSH:

```
root@sdi:~# mkdir ~/.ssh/
root@sdi:~# ssh-keygen -t rsa -f ~/.ssh/id_rsa -P ''
root@sdi:~# chmod 600 ~/.ssh/id_rsa.pub
root@sdi:~# chmod 600 ~/.ssh/id_rsa
```

Listagem 3.4 – Comando para criação do par de chaves e configuração com o SSH

A chave pública deve ser enviada para as máquinas remotas, possibilitando que seja realizada a autenticação. Uma maneira de enviar a chave para as máquinas remotas é através do comando 'ssh-copy-id', conforme mostrado na Listagem 3.5. Para utilizar o comando ssh-copy-id a máquina remota deve possuir um servidor SSH instalado e configurado para aceitar autenticação através de chaves.

```
root@sdi:~# ssh-copy-id /sdi/.ssh/id_rsa.pub root@IP_REMOTO
```

Listagem 3.5 – Comando ssh-copy-id para instalar chave pública em host remoto

Para finalizar é necessário adicionar a(s) máquina(s) clientes(s) no SDI. Por padrão, pode-se criar grupos (classes) de máquinas para um melhor gerenciamento, as classes ficam dentro da pasta CLASSES, separadas por arquivos. Cada arquivo representa uma classe e possui endereços das máquinas clientes separados por linha.

Depois de inseridas as máquinas no sistema e enviadas as chaves públicas o SDI está pronto para ser utilizado.

3.1.3 Instalação e configuração da máquina cliente

Para configurar uma máquina para ser utilizada como máquina cliente do SDI é necessária a instalação de algumas dependências e aplicativos. Em todos os exemplos e comandos apresentados a seguir foram utilizadas máquinas executando o sistema operacional Ubuntu, versão 16.04.2, x64 bits. Porém, qualquer distribuição Linux pode ser utilizada.

Como a máquina cliente irá ser utilizada somente para execução não é necessário realizar muitos procedimentos, diferentemente da máquina principal que deverá gerenciar todos esses equipamentos. A sequência de comandos exibidos na Listagem 3.6 é suficiente para deixar uma máquina totalmente preparada para ser utilizada pelo SDI.

```

root@sdi:~# apt-get update
root@sdi:~# apt-get install -y openssh-server
root@sdi:~# sed -i 's/#AuthorizedKeysFile/AuthorizedKeysFile/' \
/etc/ssh/sshd_config
root@sdi:~# mkdir -p /root/.ssh
root@sdi:~# systemctl restart sshd

```

Listagem 3.6 – Baixando pacotes e configurando o servidor SSH

Os comandos da Listagem 3.6 realizam a instalação do servidor SSH, necessário para receber conexões SSH de hosts remotos. Em seguida altera as configurações do servidor SSH para permitir a autenticação do usuário root e ativa a autenticação através da troca de chaves.

Depois que a chave pública do SDI for instalada não é mais necessário deixar habilitado o login através de senha, para desabilitar esse tipo de login, permitindo somente a autenticação através da troca de chaves, o comando mostrado na Listagem 3.7 realiza essa alteração.

```

root@sdi:~# sed -i 's/#PasswordAuthentication yes/PasswordAuthentication \
no/' /etc/ssh/sshd_config
root@sdi:~# systemctl restart sshd

```

Listagem 3.7 – Desabilitando login através de senha

Com essas configurações a máquina está pronta para ser utilizada pelo SDI.

3.2 UTILIZAÇÃO

O controle do SDI é realizado através do arquivo `sdictl`, por meio dele são inseridos comandos para o controle geral do sistema, a versão original do sistema possui as seguintes opções:

`--start`

Realiza a inicialização das funcionalidades do sistema e lança todos os túneis de comunicação com as máquinas clientes. O fluxograma apresentado na Figura 11 mostra como a função `start` é executada.

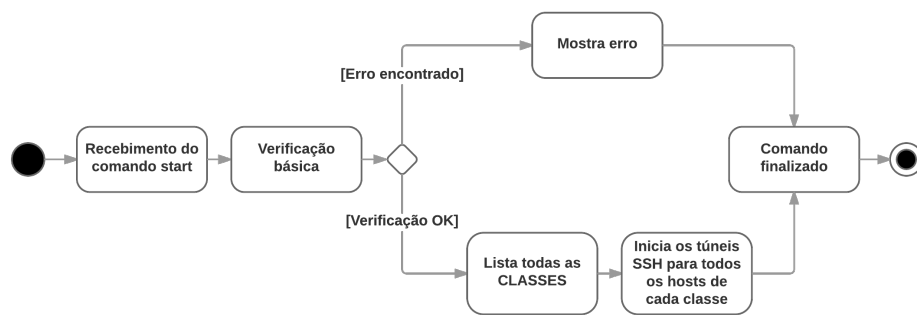


Figura 11 – Diagrama de execução do comando start

Fonte: Autoria própria

--stop

O comando stop é o inverso do comando start, ele encerra todos os túneis de comunicação SSH e desliga todas as funcionalidades do sistema. O fluxograma exibido na Figura 12 mostra como a função stop é executada.



Figura 12 – Diagrama de execução do comando stop

Fonte: Autoria própria

--restart

O comando restart é uma junção dos comandos start e stop, ele finaliza todas as conexões ativas e executa o lançamento de todos os túneis com os hosts clientes. O fluxograma da Figura 13 mostra como a função restart é executada.

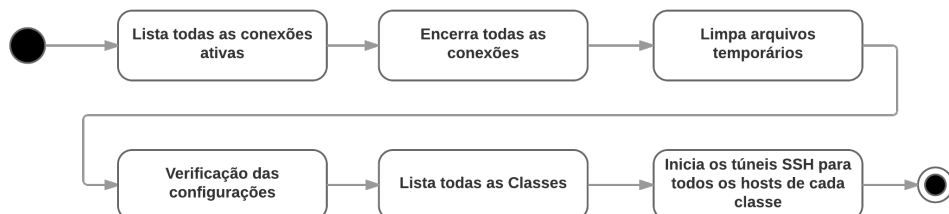


Figura 13 – Diagrama de execução do comando restart

Fonte: Autoria própria

--close

O comando `close` recebe como parâmetro um host específico para ser finalizado, ele irá fechar o túnel SSH e a máquina não irá mais ser utilizada pelo SDI até que seja iniciada novamente.

3.3 FUNCIONAMENTO INTERNO

A seguir é descrito mais detalhadamente o funcionamento interno do SDI.

O SDI possibilita que sejam enviados comandos para as máquinas, esse envio de comando é realizado através da escrita do comando em um arquivo específico. Na estrutura do SDI existe um diretório chamado `cmds` que contém um arquivo chamado `general`, além de um arquivo para cada máquina cliente conectada. O nome do arquivo é o endereço IP da máquina remota. Para enviar um comando para um único cliente basta escrever o comando no arquivo da respectiva máquina, caso seja desejado enviar um comando para todas as máquinas do sistema o comando deve ser escrito no arquivo `general`. A Listagem 3.8 mostra um comando enviado para a máquina 10.132.111.1, que faz a leitura da quantidade de memória livre. Em seguida, o mesmo comando é enviado para todas as máquinas.

```
root@sdi-master:~# echo "free -m -t" >> cmds/10.132.111.1
root@sdi-master:~# echo "free -m -t" >> cmds/general
root@sdi-master:~#
```

Listagem 3.8 – Comando de memória sendo enviado para a máquina 10.132.111.1 e em seguida para todas as máquinas conectadas

O SDI possui várias configurações que podem alterar várias opções do sistema, como os arquivos de recebimento de comandos, os locais das pastas, nomes, limites de tempo, quantidade de conexões, configurações da conexão SSH, entre outros. Todas elas podem ser visualizadas e alteradas através do arquivo `sdi.conf`. Todos os arquivos do sistema realizam o carregamento das informações com base nesse arquivo e o utilizam como guia das operações.

É possível criar comandos para serem enviados em períodos de tempo específicos e ainda filtrar as informações que são recebidas. Além disso, conforme a resposta de um comando, é possível realizar gatilhos de disparos para iniciar outros comandos. Na base do sistema se encontra a pasta `commands-available`, na pasta estão alguns exemplos de comandos, como: `loadavg` (porcentagem de utilização do processador); `rammemory` (informações da memória RAM); `status` (*online* ou *offline*) e `uptime` (tempo desde a última inicialização). Para que um desses comandos seja executado é necessário que o mesmo seja copiado ou inserido através de link simbólico dentro da pasta `commands-enabled`, esta pasta contém algumas subpastas com um período de execução pré-definido, os períodos disponíveis são exibidos na Tabela 1.

Tabela 1 – Tempos de execuções disponíveis

minutely.d	Scripts dentro dessa pasta são executados a cada minuto
hourly.d	Scripts dentro dessa pasta são executados de hora em hora
daily.d	Scripts dentro dessa pasta são executados uma vez por dia
weekly.d	Scripts dentro dessa pasta são executados uma vez por semana
monthly.d	Scripts dentro dessa pasta são executados uma vez por mês
onconnect.d	Scripts dentro dessa pasta são executados quando o SDI se conecta

Os resultados da execução de todos os comandos pré-programados ou enviados ficam salvos na pasta data. Para manter a organização das informações, cada máquina contém sua própria pasta, a qual possui os arquivos de todos os dados recebidos.

A estrutura do SDI foi bem planejada, resultando em um sistema sólido e confiável. Esta estrutura bem definida, possibilita e facilita que sejam realizadas alterações e incrementos no sistema. Sua estrutura possibilita o gerenciamento de N máquinas simultaneamente. Suas opções de funcionalidade fornecem amplas possibilidades de utilização, atendendo a grande maioria das necessidades de seus utilizadores.

A comunicação entre o sistema e as máquinas clientes é realizada de forma segura, garantindo a segurança e integridade das informações transmitidas. Além de utilizar a autenticação por troca de chaves para garantir acesso privilegiado somente a remetentes autênticos.

4 DESENVOLVIMENTO

Este capítulo explana o planejamento do sistema proposto, o seu desenvolvimento e a documentação de como utilizar o sistema final.

4.1 PLANEJAMENTO

A primeira etapa deste projeto foi entender o funcionamento do SDI. Sem esse conhecimento de base se tornaria impossível realizar alterações e incrementos para criar o suporte para execução de contêineres no sistema, logo, o entendimento de como o SDI trabalha é obrigatório. O SDI foi desenvolvido originalmente em *Bash*, porém, com o passar do tempo foram inseridas novas funcionalidades em *Python* e *C*.

O uso de mais de uma linguagem de programação gerou dificuldades para executar o sistema, pois além de existir a necessidade de instalação de alguns aplicativos e dependências, também houve dificuldades referente a execução dos códigos desenvolvidos em *Python*. Os códigos foram desenvolvidos utilizando a versão 2 da linguagem, porém, atualmente existe a versão 3 do *Python*, que é utilizada por padrão nas máquinas atuais e possui várias alterações de sintaxe em relação a sua antecessora. Estas alterações de linguagem impedem que os códigos desenvolvidos na versão 2 sejam executados corretamente na versão 3. Dessa forma, buscando deixar o sistema mais organizado e amenizar as dificuldades citadas, foi decidido migrar todas as funcionalidades desenvolvidas em *Python* e *C* para a linguagem *Bash*. Todas as alterações foram registradas em um repositório GIT para gerenciamento e auxílio de futuros trabalhos.

O repositório público do SDI está disponível em um repositório do *Github*¹. No repositório também está disponibilizada a documentação do sistema através da *Wiki*². A documentação atual do SDI foi totalmente reescrita durante o estudo do sistema, pois a documentação original não pode ser localizada, já que sua única versão estava hospedada em um servidor da *Universidade Federal do Paraná* (UFPR) e foi excluída com o passar dos anos.

A não existência da documentação original aumentou o nível de complexidade para entendimento dos arquivos e funções do sistema, sendo necessário abrir arquivo por arquivo, olhar os códigos e entender o que estava acontecendo e qual a finalidade daquele código. Este estudo mais aprofundado de cada arquivo do sistema foi mais custoso, porém, fez com que o entendimento final do sistema fosse maior se comparado a um estudo mais simples realizado através da documentação.

O planejamento do sistema foi dividido em vários blocos. Cada bloco irá possuir uma tarefa específica e a junção de todos os blocos irá resultar no sistema funcionando conforme

¹ <https://github.com/sdi-ng/>

² <https://github.com/sdi-ng/sdi/wiki>

desejado. Essa divisão em blocos possui algumas vantagens: facilita a manutenção do sistema, pois é mais fácil encontrar um bloco que esteja falhando em algum procedimento do que procurar um problema em um único grande sistema responsável por várias tarefas; torna mais prático o desenvolvimento de novas funcionalidades futuras e auxilia no entendimento do funcionamento do sistema. Os seguintes blocos foram inicialmente planejados:

- **Verifica Docker:** Verifica as máquinas conectadas ao SDI e marca as que possuem suporte a execução de contêineres Docker;
- **Tickets:** Recebe os contêineres para serem executados, armazena em uma lista de envio e gera um Ticket único para acompanhamento da execução;
- **Envio:** Acompanha os contêineres da fila de espera e os envia para as máquinas clientes.
- **Controle de execução:** Acompanha e gerencia as execuções em andamento e detecta problemas de execução.
- **Imagens:** Recebe imagens e instala nas máquinas conectadas ao SDI que possuam suporte ao Docker.
- **Status:** Consulta e mostra todas as informações de uma execução através do seu Ticket.
- **Estatísticas:** Mostra informações gerais sobre o sistema, como o número de execuções sendo realizadas, quantidade de execuções finalizadas e número de máquinas disponíveis.
- **Controle no cliente:** Conjunto de comandos instalados e executados nas máquinas clientes, utilizados para realizar algumas funções de execuções e controle.

Com esses blocos desenvolvidos e em pleno funcionamento, o sistema estará apto a receber contêineres, realizar sua execução e retornar todas as informações para o utilizador.

4.2 EXECUÇÃO

O primeiro trabalho realizado no sistema consistiu na remoção dos arquivos da linguagem *Python* e *C*. Os arquivos da linguagem *C* foram inseridos por um mestrando (RUOSO, 2013) que utilizou o sistema para realizar a comunicação entre as máquinas através de *sockets*. Como esse trabalho utiliza a comunicação através dos túneis *SSH*, a parte de *sockets* não é necessária e foi totalmente desabilitada, desta forma não é mais necessário compilar códigos da linguagem *C* para que seja possível utilizar o SDI.

A funções em *Python* estavam mais mescladas ao sistema, realizando funções importantes, como leitura de configurações, verificações de código durante a execução e algumas outras tarefas. Todos esses códigos foram reescritos na linguagem *Bash*. Nessas alterações, 63 arquivos

foram modificados e 13 foram excluídos. A partir deste ponto o SDI estava configurado e pronto para ser utilizado conforme as demandas deste projeto.

4.2.1 Verifica Docker

O primeiro bloco foi desenvolvido para verificar se a máquina cliente possui suporte a virtualização Docker, essa verificação é o ponto inicial das execuções, pois a máquina de destino obrigatoriamente necessita do Docker executando corretamente para que possa receber os contêineres.

A verificação do Docker é realizada logo após iniciada a conexão e posteriormente a cada hora. O script envia um comando solicitando informações sobre a versão do Docker instalada. Se essas informações retornarem, o script registra que a máquina possui suporte a execução de contêineres, caso contrário, a máquina é classificada como sem suporte a execução de contêineres. Na Listagem 4.1 se encontra o pseudocódigo que verifica o suporte ao Docker.

```
Para toda máquina online faça
  Solicita informações da versão do docker instalada
  Se as informações forem enviadas
  então
    Docker se encontra instalado, a máquina é marcada com suporte
    a execução de contêineres
  se não
    Docker não se encontra instalado ou possui problemas de configuração
    a máquina é marcada sem suporte a execução de contêineres
  fim
fim
```

Listagem 4.1 – Pseudo código da verificação do suporte ao Docker

Essa informação é essencial para o funcionamento do sistema e a criação do bloco não necessitou que fossem realizadas alterações no sistema, sendo necessário somente criar o script de verificação e incluí-lo na pasta de comandos disponíveis e ativos do próprio SDI.

4.2.2 Envio de imagens

O segundo bloco foi desenvolvido para enviar e instalar as imagens do Docker em todas as máquinas clientes. Para o processo de envio de imagens é necessário enviar o arquivo da imagem para as máquinas e em seguida realizar a instalação. Antes de iniciar o desenvolvimento dos códigos responsáveis por esse envio, foi conferido se o próprio sistema de envio de arquivos disponível na versão original do SDI poderia realizar essa função. Porém, a estrutura de envio já

criada no sistema não poderia verificar se a máquina remota possui suporte ao Docker. Para evitar a alteração dessa funcionalidade, que pode ser utilizada em outras situações, foi criada a função `--sendimage` no `sdictl`, realizando o envio de imagens exclusivamente para as máquinas que possuem suporte a execução de contêineres. Mesmo sendo uma funcionalidade nova, foi possível aproveitar alguns códigos do sistema de envio de arquivos do SDI.

A função de enviar imagens segue o seguinte protocolo de execução: Primeiramente todas as imagens recebidas para instalação são enviadas para a pasta `/tmp/sdiimages/` em todas as máquinas clientes. Na sequência existe uma função que é responsável por acompanhar os arquivos da pasta `/tmp/sdiimages/` e detectar se alguma imagem deve ser instalada. Quando uma nova imagem é recebida ela é instalada automaticamente. Esse comando de verificação e instalação é executado a cada minuto pelo próprio SDI.

A Listagem 4.2 demonstra o pseudocódigo que é realizado para enviar uma imagem para todas as máquinas que possuem suporte ao Docker.

```
Para toda máquina conectada faça
  se suporta docker faça
    envia imagem;
  fim
fim
```

Listagem 4.2 – Pseudo código para envio de uma imagem do Docker

O código que verifica quando uma imagem é recebida funciona da seguinte forma: são listados todos os arquivos que se encontram na pasta `/tmp/sdiimages/`, para cada arquivo encontrado é realizada a execução do comando de importação do próprio Docker. Após a instalação o arquivo da imagem é deletado, pois todas as informações necessárias para a utilização da imagem já estão disponíveis no repositório de imagens do Docker. Ao instalar a imagem no Docker é seguido um padrão para a nomenclatura das imagens. Todas são instaladas com o nome ‘sdi’ e suas versões são nomeadas conforme a imagem enviada. Por exemplo, se for recebida uma imagem chamada `debian-teste`, ela será instalada com o nome ‘sdi’ e sua versão se chamará `debian-teste`. Os contêineres que irão fazer uso dessa imagem precisam dessa informação para funcionar corretamente. A Listagem 4.3 exibe o código que realiza essa função.

```
#lista os arquivos da pasta /tmp/sdiimages/ na variável arq
for arq in `ls /tmp/sdiimages/`; do
  # para cada arquivo encontrado executa o comando de
  # importação e remove o arquivo
  docker import /tmp/sdiimages/$arq sdi:$arq
  rm /tmp/sdiimages/$arq
done
```

Listagem 4.3 – Código BASH para instalação de uma imagem do Docker

A Figura 14 mostra o diagrama completo do processo de envio e instalação de uma imagem.

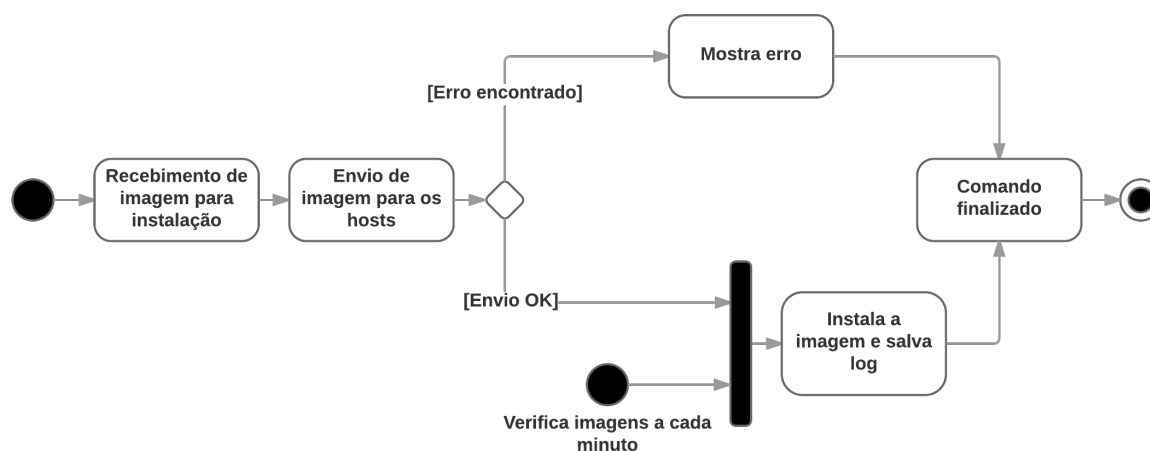


Figura 14 – Estrutura de funcionamento de envio e instalação de uma imagem do Docker

Fonte: Autoria própria

A melhor maneira de enviar os contêineres pela rede depende da estrutura onde o sistema estará sendo utilizado e de alguns fatores externos. A transmissão poderá ser realizada diretamente entre o servidor do SDI e a máquina cliente através da conexão SSH. Também é possível realizar a distribuição das imagens através de máquinas intermediárias (visando não sobrecarregar a máquina do SDI) ou também através do *Docker Registry*, podendo esse último ser local ou público através do serviço oferecido pelo próprio Docker, o (*Docker Hub*). Caso a opção do *Docker Hub* seja escolhida, as máquinas clientes deverão, obrigatoriamente, possuir uma conexão ativa com a Internet. A última opção de envio não foi integrada neste projeto, podendo ser realizada como trabalho futuro.

4.2.3 Recebimento de contêineres e geração de tickets

Com o sistema de envio de imagens finalizado, passou-se ao desenvolvimento da estrutura para recebimento de contêineres. O planejamento foi realizado conforme mostrado no diagrama da Figura 15.

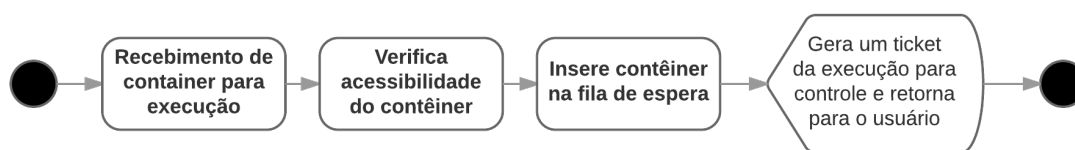


Figura 15 – Estrutura de funcionamento de recebimento de contêineres para execução

Fonte: Autoria própria

Como pode ser visto na Figura 16 o envio e execução de contêineres não realiza muitas tarefas. Ele também poderia ser responsável por escolher a máquina cliente que seria responsável pela execução do contêiner recebido, porém, essa tarefa foi transferida para o bloco de controle de execuções, pois o mesmo possui controle e acesso a informação completa das máquinas clientes, identificando máquinas disponíveis para recebimento de contêineres e detectando problemas. Desta forma, pode ser realizado o despacho da fila de espera de maneira mais efetiva.

Quando um contêiner é recebido pelo sistema, é verificado se ele existe e está acessível. Se não forem encontrados problemas, será gerado um ticket pseudoaleatório que irá seguir junto ao contêiner durante toda a sua execução. Ele é o identificador que será utilizado pelo controle do sistema e pelo usuário utilizador, possibilitando a verificação do estado atual da execução e todas as informações complementares como: registros de horários de ações; máquina responsável pela execução e quantidade de tentativas de execução realizadas.

O sistema de recebimento de contêineres foi desenvolvido para aceitar duas formas de recebimento, cada uma programada no arquivo `sdict1` para um tipo específico de execução:

- container: Esta opção irá receber um contêiner fechado e realizar sua execução direta. O SDI e o Docker não irão gerenciar nenhuma instalação e/ou transferência de imagens e/ou arquivos. Entende-se que o contêiner está completo, com sua imagem e todos os arquivos necessários para sua execução. Ele será recebido e enviado diretamente para execução.
- contar: Esta opção irá receber um pacote de execução. Este pacote conterá arquivos necessários para a execução e um arquivo de comando, o qual irá funcionar como gatilho da execução. Esta forma de uso utiliza as propriedades do sistema de camadas do Docker, pois o arquivo de comando irá indicar qual imagem será utilizada e sua localização, sendo o Docker responsável pela aquisição da imagem, geração e execução do contêiner.

Ao receber um contêiner o SDI deve receber, além do endereço do contêiner, o tempo limite de execução em segundos. Caso o usuário não informe o tempo limite, o próprio SDI irá definir o tempo limite automaticamente. Atualmente o tempo limite automático está definido como 3600 segundos (uma hora).

4.2.4 Distribuição de contêineres

Quando o bloco de verificação de execuções define qual máquina cliente irá executar um contêiner da fila de espera, o bloco de envio de contêineres é responsável por iniciar o envio do contêiner. O processo de envio dos contêineres funciona conforme exibido no diagrama da Figura 16.

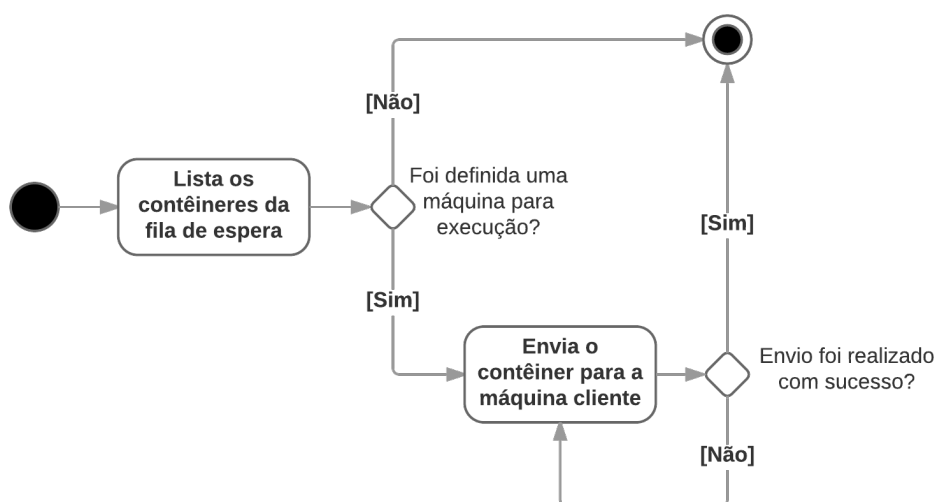


Figura 16 – Estrutura de funcionamento de envio de contêineres para execução

Fonte: Autoria própria

O envio dos contêineres possui uma regra de ordenação de envio, a qual é realizada através de um sistema de fila, onde o primeiro contêiner a ser inserido no SDI será o primeiro contêiner a ser enviado para execução. Esse modelo é conhecido como *First In First Out* (FIFO).

A ordenação do envio busca garantir que os contêineres serão executados o mais rápido possível, com o mínimo de tempo de espera possível.

4.2.5 Controle no cliente

Quando um contêiner é enviado para uma máquina cliente, ele é salvo no diretório /containerstoexecute/ ou /tarstoexecute/, conforme o tipo de execução. Um conjunto de comandos instalados na máquina cliente são responsáveis por detectar os contêineres e realizar o início de suas execuções. Esse conjunto de comandos foi nomeado de *coresdi-client*. Este conjunto de comandos poderia ser instalado na máquina do SDI e ser executado periodicamente para todas as máquinas, porém, atualmente o tempo mínimo de execução de scripts do SDI é de um minuto, nos piores casos um contêiner poderia ficar um minuto aguardando sua execução, seria um minuto de desperdício de recursos da máquina, esta foi a principal motivação para

a criação do bloco nas máquinas cliente, desta forma a detecção de novos contêineres é realizada a cada 10 segundos, podendo este valor ser alterado nas configurações. O `coresdi-client` é automaticamente enviado e configurado em todas as máquinas clientes durante a inicialização do SDI.

O bloco não faz uso de nenhum limitador de execuções nas máquinas clientes. Cada contêiner recebido é automaticamente executado. Seria interessante para trabalhos futuros, incrementar esse sistema para verificar se a utilização da máquina está abaixo de um determinado limiar antes de iniciar a execução de um novo contêiner, porém, essa versão do sistema é totalmente controlada pelo SDI, que é responsável por realizar o controle de quantos contêineres cada máquina está executando.

O *script* de verificação consulta o conteúdo das pastas `/containerstoexecute/` e `/tarstoexecute/` a cada 10 segundos, quando a pasta possui contêineres, os mesmos são importados no sistema do Docker e sua execução é iniciada. Depois de iniciada, o contêiner recebido é excluído da pasta e um arquivo de controle contendo o *ticket* do contêiner é criado na pasta `/containersrunning/`. A Figura 17 exhibe o diagrama de recebimento e execução dos contêineres em uma máquina cliente.

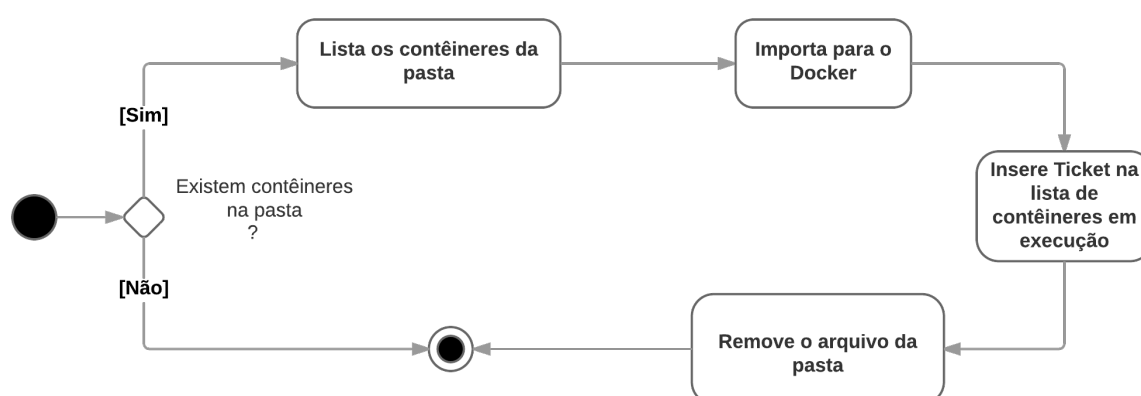


Figura 17 – Diagrama do início de uma execução de um contêiner em uma máquina cliente

Fonte: Autoria própria

Visando realizar o controle das execuções, um segundo *script* de verificação também é executado a cada 10 segundos na máquina cliente, sendo responsável por verificar o estado dos contêineres que se encontram em execução. Para cada *ticket* na pasta `/containersrunning/` é verificado se o contêiner ainda se encontra em execução, caso o mesmo tenha sido finalizado, verifica-se se foi gerado o arquivo de saída. Caso o contêiner não esteja mais sendo executado e nenhum arquivo de saída tenha sido gerado, o fato será inserido no registro de erros. A Figura 18 mostra o diagrama deste processo.

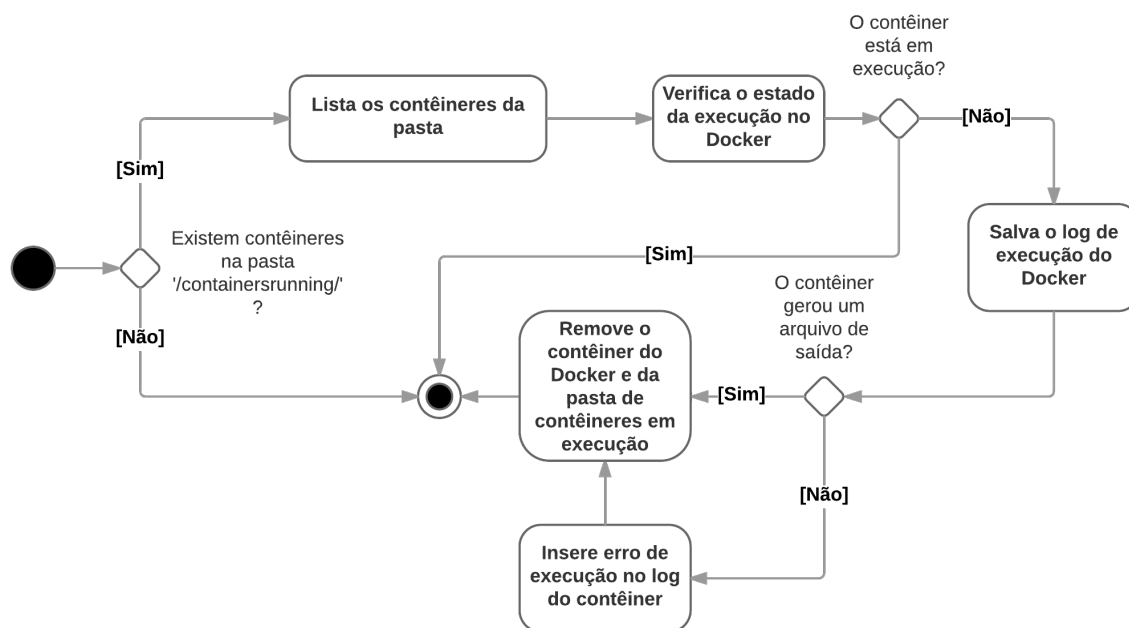


Figura 18 – Diagrama de acompanhamento de execução de contêineres em uma máquina cliente

Fonte: Autoria própria

Como pode ser visto no diagrama da Figura 18, o sistema verifica se o contêiner gerou um arquivo de saída contendo os resultados da execução, o endereço onde este arquivo deve ser salvo segue um conjunto de regras do sistema. Desta forma, o utilizador do sistema deve estar ciente que existem regras de nomeação que devem ser seguidas durante a criação do contêiner ou do pacote de execução. Sem essa configuração o contêiner pode ser executado corretamente, porém, seu resultado não chegará corretamente ao SDI.

4.2.6 Regras de uso

O arquivo de resultado de uma execução não necessita possuir uma extensão específica. O utilizador pode salvar o resultado da execução em qualquer formato, podendo ser um arquivo de texto ou vários arquivos compactados, somente é exigido que o nome deste arquivo seja igual ao ticket do contêiner. O arquivo de saída deve ser salvo na pasta /compartilhada, a qual é criada automaticamente pelo SDI.

O contêiner visualiza o endereço /compartilhada como sendo uma pasta normal do sistema de arquivos, porém, ela é um *Volume* do Docker criado juntamente com o contêiner. Através deste volume é criada uma união entre o endereço /compartilhada do contêiner e a pasta /data da máquina hospedeira. Desta forma, os arquivos são salvos na máquina que está executando o contêiner e não dentro do contêiner, facilitando o acesso do SDI aos arquivos de resultados e logs de execução.

Além da regra do endereço do arquivo de resultados, existe uma segunda regra para a

criação dos contêineres. Segundo a documentação do Docker é necessário informar o programa que será executado no contêiner. Este programa inicial também deve ser nomeado de `inicia.sh` e deve estar localizado no diretório raiz do contêiner. Mesmo sendo um arquivo da linguagem *Bash*, não existem limitações das linguagens que podem ser executadas nos contêineres, o arquivo `inicia.sh` é somente o gatilho de qualquer execução, podendo por exemplo: iniciar um arquivo de outra linguagem, compilar um programa em C ou inicializar qualquer operação que o contêiner possa executar.

As duas regras valem tanto para a execução de contêineres fechados, quanto para a execução por pacotes de execução. A utilização dos pacotes de execução requer um arquivo adicional em relação aos contêineres fechados. Este arquivo adicional é responsável por indicar ao sistema como o contêiner deve ser criado e executado. Este arquivo adicional é nomeado como `command` e deve se encontrar no diretório raiz do pacote.

4.2.7 Controle de execução e estatísticas

Cada contêiner possui seu próprio tempo limite de execução, seus arquivos de resultados, registros de datas e *logs*. Desta forma foi criado um bloco responsável pelo controle das execuções de contêineres, que é responsável por acompanhar os contêineres em execução, verificar se ainda estão dentro do prazo limite de execução, se já foram executados, além de detectar falhas de execução. O sistema responsável por esse controle de execuções funciona conforme mostrado na Figura 19.

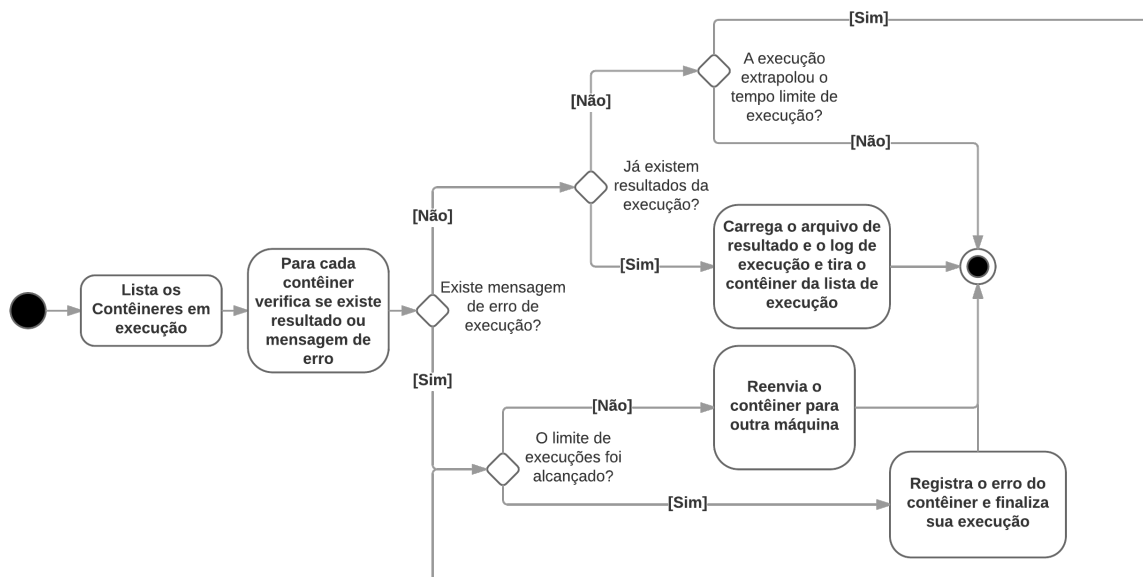


Figura 19 – Estrutura de funcionamento do sistema de controle de execuções

Fonte: Autoria própria

O bloco de controle das execuções também realiza uma segunda tarefa, que consiste

em gerenciar a quantidade de contêineres que as máquinas clientes estão executando, além de escolher as máquinas responsáveis pela execução dos contêineres que se encontram na fila de espera. Esse processo paralelo é realizado conforme o diagrama da Figura 20.

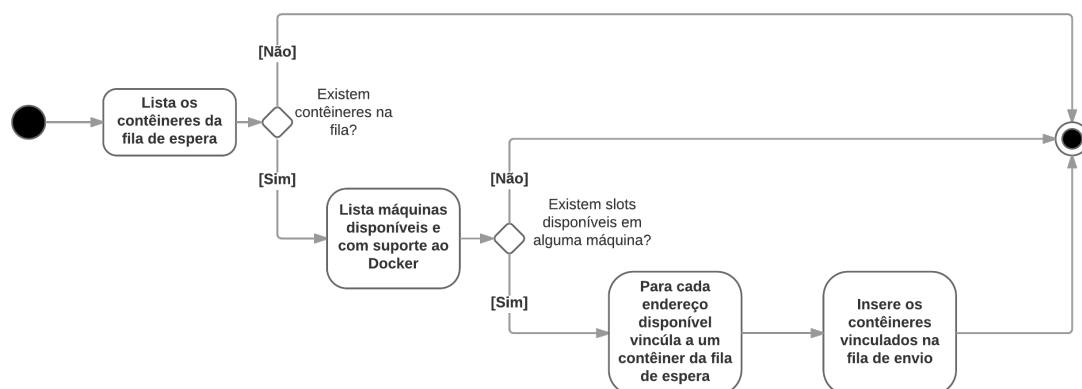


Figura 20 – Estrutura de funcionamento do sistema de controle de máquinas livres e vinculação dessas máquinas com os contêineres da fila de espera

Fonte: Autoria própria

Com esses módulos o sistema já é capaz de receber contêineres e realizar a execução dos mesmos.

4.2.8 Acompanhamento das execuções

Para auxiliar no acompanhamento das execuções e geração de estatísticas, foram criados dois blocos para consulta de dados. O primeiro recebe um ticket de um determinado contêiner e retorna todas as informações de sua execução. Sua utilização é realizada através da opção `--status=ticket`, no qual `ticket` representa o número de identificação do contêiner. Como pode ser visto na Listagem 4.4, são mostradas as seguintes informações: quando o contêiner foi recebido pelo sistema; quando o contêiner foi enviado para execução; quando sua execução foi finalizada; o tempo limite de execução; quantas tentativas de execução foram realizadas; informações de logs da execução, normalmente utilizados para gravar registros de erros de execução; o tamanho do arquivo de resposta e sua localização no sistema; qual máquina foi responsável pela execução do contêiner e o estado de execução do contêiner.


```

root@sdi-master:~/sdi-docker# ./sdictl --status=IPRRoFgeD
Ticket: IPRRoFgeD
Recebido em: Thu Jun  8 01:15:21 UTC 2017
Inicializado em: Thu Jun  8 01:16:54 UTC 2017
Finalizado em: Thu Jun  8 01:17:40 UTC 2017
Tempo limite de execução (seg): 3600
Status: EXECUTED-OK
Tentativas de Execucao: 1
Log: SDI FINISHED: Done...
Arquivo de resposta: ./data_containers/IPRRoFgeD/result (44 bytes)
Executado pelo cliente: 10.132.111.184

```

Listagem 4.4 – Consulta do resultado de execução do contêiner representado com o ticket IPRRoFgeD

O sistema possui atualmente os seguintes estados de execução:

IN-QUEUE: O contêiner foi recebido e está na fila de espera para execução.

SENT-RUNNING: O contêiner foi enviado para uma máquina cliente e está sendo executado.

ERROR-TRYING-AGAIN: Ocorreu algum problema de execução e está sendo realizada mais uma tentativa de execução em outra máquina cliente.

FATAL-ERROR: Contêiner não foi executado corretamente em todas as tentativas realizadas, informações auxiliares podem ser encontradas no log de erros.

EXECUTED-OK: Contêiner foi executado com sucesso.

TIMEOUT: O tempo limite de execução foi alcançado.

O segundo bloco pode ser utilizado para visualizar as informações gerais de todas as execuções. Sua utilização é realizada através da opção `--resume` como é mostrado na Listagem 4.5. São informações simples, mas que fornecem um bom entendimento do que está ocorrendo no sistema.

```

root@sdi-master:~/sdi-docker# ./sdictl --resume
Numero total de contêineres: 2
Na fila de envio: 1
Em execução: 0
Em reexecução: 0
Executados com sucesso: 1
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0

```

Listagem 4.5 – Consulta geral sobre as execuções que se encontram em andamento ou já foram finalizadas

Durante o desenvolvimento do sistema surgiram momentos em que a elevada quantidade de informações das máquinas clientes dificultava o entendimento do que estava acontecendo, principalmente nas buscas de problemas ocorridos durante o desenvolvimento e durante os testes preliminares. Para amenizar este problema, foi incluída a opção `--cleardata` no sistema de controle do `sdict1`, a função limpa todos os dados já recebidos de todas as máquinas clientes.

4.2.9 Varredura de rede

A inclusão de várias máquinas clientes requer informar o endereço IP de cada máquina nas classes do SDI. Visando auxiliar este serviço, foi criado um *script* de busca que varre uma rede informada por parâmetros e tenta encontrar máquinas disponíveis. A busca é baseada em uma faixa de endereços IP informada nos parâmetros. A verificação é realizada através da tentativa de login no IP remoto, caso a conexão seja realizada com sucesso, a máquina é inserida automaticamente na classe `autoinsert`. A Figura 21 demonstra o sistema de operação desta função.

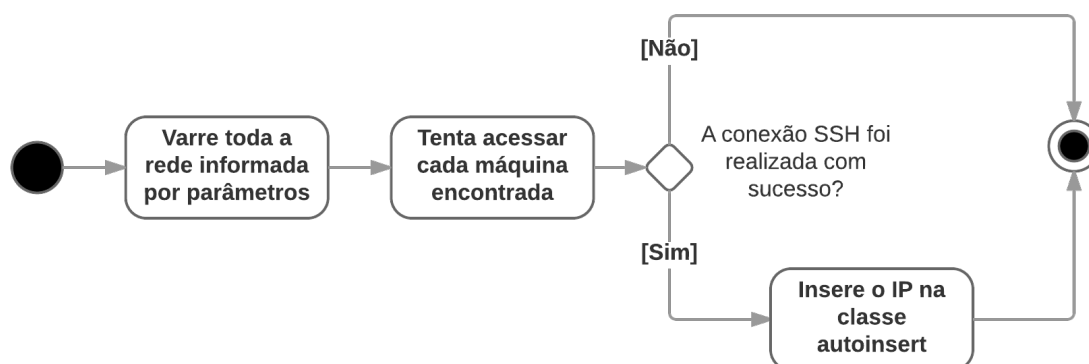


Figura 21 – Estrutura de funcionamento da varredura de rede e inclusão automática de máquinas

Fonte: Autoria própria

Com a estrutura montada com estes blocos, o SDI se tornou um ambiente genérico que pode ser utilizado para realizar a execução de contêineres das mais diversas finalidades.

4.3 DOCUMENTAÇÃO

Após a realização do desenvolvimento das novas funções do sistema, alguns arquivos foram alterados, outros foram incluídos, novas opções de configuração foram disponibilizadas e novos comandos podem ser realizados através do arquivo de controle `sdict1`. Nessa seção serão

introduzidos os novos comandos e configurações disponíveis, juntamente com suas formas de utilização.

A instalação do sistema permanece sendo realizada através do gerenciador Git, conforme descrito na subseção 3.1.1. As principais alterações perceptíveis são os arquivos e pastas do sistema, como pode ser visualizado na Listagem 4.6.

```

root@sdi-master:~/sdi-docker# ls -F
commands-available/      createcore.sh* (novo)      launchsditunnel.sh*
networksearch.sh* (novo) resume.sh* (novo)         sdiweb/
sendimage.sh* (novo)    states.sh*                AUTHORS*
commands-enabled/      data/                     launchsendfile.sh*
parser.sh*             runningcheckdaemon.sh* (novo) www/
status.sh*            sendcommandtar.sh* (novo) CLASSES/
COPYING*              data_containers/ (novo)  launchscripts.sh*
misc.sh*              sdi.conf*                states-available/
sendcontainer.sh* (novo) summaries-available/    cmds/
coresdi-client/ (novo) launchsdi.sh*            modules/
README*               sdictl*                  sendfile.sh*
states-enabled/      sync.sh*

```

Listagem 4.6 – Listagem dos arquivos do SDI depois de realizadas as novas alterações

O arquivo `createcore.sh` é responsável por gerar a pasta `coresdi-client`, que é enviada para execução nas máquinas clientes. O arquivo `networksearch.sh` é responsável pelo sistema de varredura de rede e inclusão automática de máquinas clientes. O arquivo `resume.sh` é o bloco que mostra o resumo geral das execuções. O arquivo `sendimage.sh` é o bloco responsável pelo envio de novas imagens que serão instaladas em todas as máquinas. O arquivo `runningcheckdaemon.sh` é o bloco que realiza o acompanhamento das execuções em todas as máquinas clientes e define as máquinas livres para a execução dos contêineres da fila de espera. A pasta `data_containers` possui todos os contêineres executados pelo sistema, com seus arquivos de resultado, de datas, de *logs* e demais arquivos de informações.

O arquivo `sdi.conf` recebeu cinco novas configurações para a execução de contêineres:

`CONTAINER_POOL`: (string) Define a pasta na qual serão salvos todos os contêineres e seus respectivos dados de execução.

`MAX_CONTAINERS_BY_HOST`: (int) Define quantos contêineres podem ser executados simultaneamente por cada máquina cliente.

`DOCKER_REGISTRY`: (true/false) Define se o SDI deve iniciar um serviço de repositório de imagens.

`DOCKER_REGISTRY_IP`: (IP) Define o endereço IP do repositório de imagens. Este IP deve ser acessível de todas as máquinas clientes.

`EXECUTION_ATTEMPTS`: (int) Define quantas tentativas de execução devem ser realizadas para cada contêiner. Esta opção é utilizada somente quando algum problema é encontrado ou quando o tempo limite de execução é alcançado.

Para realizar o controle das novas funcionalidades foram inseridos os seguintes comandos ao `sdictl`:

```
--container=/pasta/completa/container --tl=TEMPO
```

Recebe o local que se encontra um contêiner fechado e o seu tempo limite de execução em segundos.

```
--contar=/pasta/completa/pacote --tl=TEMPO
```

Recebe o local que se encontra um pacote de execução e o seu tempo limite de execução em segundos.

```
--status=ticket
```

Mostra todas as informações de uma determinada execução através do seu ticket.

```
--resume
```

Mostra um resumo de todas execuções que estão em andamento, as que já foram finalizadas e o número de máquinas clientes disponíveis.

```
--cleardata
```

Remove todas as informações recebidas das máquinas clientes.

```
--sendimage=/pasta/completa/imagem
```

Recebe o local que se encontra uma imagem que será instalada em todas as máquinas clientes.

A configuração das máquinas clientes continua sendo realizada da mesma forma que foi instruído na subseção 3.1.3. Mesmo tendo sido criados vários blocos com novas funcionalidades, a utilização do sistema continua sendo realizada de forma simples e sem maiores dificuldades.

4.4 DISCUSSÕES DAS IMPLEMENTAÇÕES

Esta seção descreve a realização de testes das funcionalidades do SDI com suporte a execução de contêineres.

4.4.1 Envio de imagens

O envio de imagens do Docker é responsável por receber uma imagem, enviá-la para todas as máquinas e realizar a instalação nas mesmas. Para realizar um teste de funcionamento da opção foram seguidos os seguintes procedimentos: foi realizado o download de uma imagem oficial da distribuição Ubuntu; a imagem foi exportada para um arquivo local e a função de envio e instalação de imagens foi executada. A Listagem 4.7 exibe os comandos realizados para a realização do processo.

```
root@master:/sdi-docker# docker pull ubuntu
root@master:/sdi-docker# docker save -o tmp/ubuntu ubuntu
root@master:/sdi-docker# ./sdictl --sendimage=/root/sdi-docker/tmp/ubuntu
Iniciando envio da imagem...
Inserindo imagem na fila de envio dos hosts da classe clientes (1/1)...done
Fila criada com sucesso, aguarde a transferência...
```

Listagem 4.7 – Download da imagem de um repositório oficial, exportação da imagem, envio e instalação em todas as máquinas clientes.

A verificação do processo é realizada através da verificação das imagens instaladas nas máquinas clientes. A Listagem 4.8 mostra que a imagem enviada foi instalada em uma máquina cliente.

```
root@sdi-client1:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
sdi                  ubuntu             eec7ec205bac       50 seconds ago    121MB
```

Listagem 4.8 – Listagem de imagens instaladas em uma máquina clientes.

Todas as máquinas clientes foram acessadas e verificadas. Concluiu-se que a opção de envio e instalação de imagens estava realizando sua função corretamente.

4.4.2 Criação do contêiner fechado e do pacote de execução

O segundo teste do sistema consistiu na execução de contêineres genéricos. Testando simultaneamente várias funcionalidades do sistema, como: recebimento e envio de um contêiner;

bloco de informações gerais; bloco de informações de ticket e bloco de controle de execuções. O contêiner de teste realiza a escrita da frase "**O SDI já possui suporte a execução de contêineres!**" no arquivo de saída padrão após três minutos de execução. Este tempo de três minutos possibilita o teste do bloco de controle de execução, pois pode-se informar ao SDI que o contêiner deve entregar uma resposta em até 1 minuto. Como o contêiner não é executado em menos de três minutos, o SDI deve cancelar a execução do contêiner na máquina remota e registrar o erro.

A imagem do contêiner será criada através de um Dockerfile e baseada na imagem do Ubuntu baixada no teste anterior. O primeiro passo para a criação do contêiner consistiu em programar o arquivo `inicia.sh`, que é o arquivo iniciado automaticamente na execução de um contêiner, como explicado na subseção 4.2.6. A Listagem 4.9 mostra o código do arquivo `inicia.sh` com comentários explicando cada comando utilizado.

```
#!/bin/bash
# A linha acima informa ao sistema a linguagem que deve ser utilizada na
# execução deste arquivo

# o seguinte comando irá bloquear a execução do arquivo por 3 minutos
sleep 3m

# após 3 minutos a frase de saída será escrita no arquivo de saída padrão
printf "O SDI já possui suporte a execução de contêineres!" > /compartilhada/$1

# o seguinte comando informa ao sistema que a execução foi concluída com
# sucesso
exit 0
```

Listagem 4.9 – Programação do arquivo `inicia.sh` para a criação do contêiner

Em seguida é necessário configurar o arquivo Dockerfile para a criação da imagem. A Listagem 4.10 possui as configurações mínimas e suas descrições.

```
# indica que a imagem base é a última versão do ubuntu
FROM ubuntu:latest

# copia o arquivo inicia.sh para a raiz
COPY inicia.sh /

# da permissões de execução ao arquivo inicia.sh
RUN chmod 755 /inicia.sh

# indica que o contêiner é iniciado pelo arquivo inicia.sh
CMD /bin/bash inicia.sh
```

Listagem 4.10 – Programação do arquivo Dockerfile para a criação da imagem

A programação do arquivo Dockerfile pode fornecer a ideia de que o usuário do sistema pode definir o arquivo de início dos contêineres, ignorando a norma da documentação. Porém, esta ideia não é válida, pois ao receber um contêiner para execução, a máquina cliente irá executar o contêiner recebido através do arquivo `inicia.sh` e irá informar o ticket do contêiner. O ticket é informado no primeiro parâmetro e é utilizado para salvar a saída da execução no local correto.

Com o arquivo `inicia.sh` e Dockerfile programados, é realizada a criação da imagem; criação do contêiner e exportação do contêiner criado. Todos esses passos estão descritos na Listagem 4.11 ³.

```
# cria a imagem e a define com o nome sdi:testecontainer
root@sdi-master:~/sdi-docker/tmp/cont_teste# docker build -t sdi:testecontainer .

# cria o contêiner e define seu nome como 'containerteste'
docker create --name containerteste -ti sdi:testecontainer /inicia.sh

# exporta o contêiner 'containerteste' para o arquivo 'container'
docker export -o container containerteste
```

Listagem 4.11 – Criação da imagem, criação e exportação do contêiner

Ao final da execução dos comandos a pasta conterá o arquivo `inicia.sh`, o Dockerfile e o contêiner exportado. Após a criação e exportação do contêiner, iniciou-se a criação de um pacote de execução.

A criação de um pacote de execução requer a criação do arquivo `command`, exibido e comentado na Listagem 4.12. Este arquivo é o gatilho para a execução, que será responsável por criar o contêiner baseado na imagem do Ubuntu e inicializar sua execução.

```
#!/bin/bash
PREFIX=$(dirname $0)

# cria o container baseado na imagem do ubuntu
docker create -v /data:/compartilhada --name $1 -ti \
ubuntu:latest /inicia.sh $1

# executa o container
docker start $1
```

Listagem 4.12 – Programação do arquivo command para a criação de um pacote de execução para testes

³ Os comandos são executados na pasta onde se encontram os dois arquivos

O arquivo `command` e o arquivo `inicia.sh` devem ser compactados em um arquivo da extensão `.tar.gz` antes de serem inseridos no SDI.

4.4.3 Execução por contêiner fechado

A execução por contêiner fechado consiste em inserir um contêiner completo no SDI e aguardar sua execução. Após a execução foi realizada a verificação do conteúdo do arquivo de saída, comparando seu valor em relação com o valor desejado. A Listagem 4.13 exibe o comando de inserção do contêiner no SDI e o ticket gerado para acompanhamento.

```
root@sdi-master:~/sdi-docker# ./sdictl --container=/root/sdi-docker/ \
tmp/container_teste/container
WARNING: Time limit option was not set! The default limit will be used (1 hour).
du9HUAc03fRBYXW4HbjUMu7b2gJcYaCwKUjiFoUePDifxfR02cNw5Dw3HZ8NqyUb
```

Listagem 4.13 – Criação da imagem, criação e exportação do contêiner

Além do ticket de acompanhamento o SDI exibiu um alerta indicando que não foi inserido o tempo limite de execução, quando este parâmetro não é informado, o SDI configura o tempo limite conforme definido em seu arquivo de configuração, neste caso, 60 minutos. As Listagens 4.14 e 4.15 mostram o acompanhamento da execução. Após o contêiner ser inserido foi solicitado um resumo das operações do sistema, mostra-se um contêiner em execução. Aproximadamente 6 minutos depois é solicitado um novo resumo e o sistema informa que um contêiner foi executado com sucesso.

```
root@sdi-master:~/sdi-docker# date
Sat Jun 3 03:50:42 UTC 2017

root@sdi-master:~/sdi-docker# ./sdictl --resume
Numero total de contêineres: 1
Na fila de envio: 0
Em execução: 1
Em reexecução: 0
Executados com sucesso: 0
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
-----
Máquinas Online: 1
Máquinas com suporte ao Docker: 1
```

Listagem 4.14 – Acompanhamento da execução do contêiner de teste


```

root@sdi-master:~/sdi-docker# date
Sat Jun 3 03:56:06 UTC 2017

root@sdi-master:~/sdi-docker# ./sdictl --resume
Numero total de contêineres: 1
Na fila de envio: 0
Em execução: 0
Em reexecução: 0
Executados com sucesso: 1
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
-----
Máquinas Online: 1
Máquinas com suporte ao Docker: 1

```

Listagem 4.15 – Acompanhamento da execução do contêiner de teste

A informação que o contêiner foi executado com sucesso atende o esperado, em sequência foi solicitado informações específicas da execução e visualizado o conteúdo do arquivo de resposta. A Listagem 4.16 exibe os dados da execução e a resposta da execução.

```

root@sdi-master:~/sdi-docker# ./sdictl --status=du9HUAc03fRBYXW4HbjUMu7b2gJ\
cYaCwKUjiFoUePDifxfR02cNw5Dw3HZ8NqyUb
Ticket: du9HUAc03fRBYXW4HbjUMu7b2gJcYaCwKUjiFoUePDifxfR02cNw5Dw3HZ8NqyUb
Recebido em: Sat Jun 3 03:49:54 UTC 2017
Inicializado em: Sat Jun 3 03:50:01 UTC 2017
Finalizado em: Sat Jun 3 03:53:13 UTC 2017
Tempo limite de execução (seg): 3600
Status: EXECUTED-OK
Tentativas de Execucao: 1
Log: SDI FINISHED: Done...
Arquivo de resposta: ./data_containers/du9HUAc03fRBYXW4HbjUMu7b2gJcYaCwKUji
FoUePDifxfR02cNw5Dw3HZ8NqyUb/result (54 bytes)
Executado pelo cliente: 165.227.76.211

root@sdi-master:~/sdi-docker# cat ./data_containers/du9HUAc03fRBYXW4HbjUMu7\
b2gJcYaCwKUjiFoUePDifxfR02cNw5Dw3HZ8NqyUb/result
0 SDI já possui suporte a execução de contêineres!

```

Listagem 4.16 – Informações gerais da execução e conteúdo do arquivo de saída

A execução foi realizada com sucesso e o arquivo de saída possuía o valor desejado. Concluiu-se que esta funcionalidade estava realizando sua função corretamente.

4.4.4 Execução por pacote de execução

A execução por pacotes de execução consiste em inserir um pacote de dados no SDI, que se encarregará de criar o contêiner, conforme especificado, e realizar sua execução. Após a execução foi realizada a verificação do conteúdo do arquivo de saída, comparando seu valor em relação ao valor desejado. A Listagem 4.17 exibe o comando de inserção do pacote no SDI e o ticket gerado para acompanhamento.

```
root@sdi-master:~/sdi-docker# ./sdictl --sentar=/root/sdi-docker/ \
tmp/container_teste/pacote.tar.gz --tl=3600
PCwcWYvV5s9YZA0uhm5SQczZwLsdEVqp5PtHU5QY8YEvq65ZKe1DS3v0Z2siYRVo
```

Listagem 4.17 – Criação da imagem, criação e exportação do contêiner

Além do ticket de acompanhamento o SDI exibiu um alerta indicando que não foi inserido o tempo limite de execução, quando este parâmetro não é informado, o SDI configura o tempo limite conforme definido em seu arquivo de configuração, neste caso, 60 minutos. As Listagens 4.18 e 4.19 mostram o acompanhamento da execução. Após o contêiner ser inserido foi solicitado um resumo das operações do sistema, mostrando um contêiner em execução. Aproximadamente 6 minutos depois é solicitado um novo resumo e o sistema informa que um contêiner foi executado com sucesso.

```
root@sdi-master:~/sdi-docker# date
Sat Jun 3 06:01:13 UTC 2017

root@sdi-master:~/sdi-docker# ./resume.sh
Numero total de contêineres: 1
Na fila de envio: 0
Em execução: 1
Em reexecução: 0
Executados com sucesso: 0
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
[...]
```

Listagem 4.18 – Acompanhamento da execução do pacote de execução.

```

root@sdi-master:~/sdi-docker# date
Sat Jun 3 06:07:20 UTC 2017

root@sdi-master:~/sdi-docker# ./resume.sh
Numero total de contêineres: 1
Na fila de envio: 0
Em execução: 0
Em reexecução: 0
Executados com sucesso: 1
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
[...]

```

Listagem 4.19 – Acompanhamento da execução do pacote de execução.

A informação que o contêiner foi executado com sucesso atende o esperado, em sequência foi solicitado informações específicas da execução e visualizado o conteúdo do arquivo de resposta. A Listagem 4.20 exibe os dados da execução e a resposta da execução.

```

root@sdi-master:~/sdi-docker# ./sdictl --status=PCwcWYvV5s9YZA0uhm5SQCzZwLs\
dEVqp5PtHU5QY8YEvq65ZKe1DS3v0Z2siYRVo
Ticket: PCwcWYvV5s9YZA0uhm5SQCzZwLsdEVqp5PtHU5QY8YEvq65ZKe1DS3v0Z2siYRVo
Recebido em: Jun 3 06:00:50 UTC 2017
Inicializado em: Jun 3 06:01:10 UTC 2017
Finalizado em: Jun 3 06:03:01 UTC 2017
Tempo limite de execução (seg): 3600
Status: EXECUTED-OK
Tentativas de Execucao: 1
Log: SDI FINISHED: Done...
Arquivo de resposta: ./data_containers/PCwcWYvV5s9YZA0uhm5SQCzZwLs
dEVqp5PtHU5QY8YEvq65ZKe1DS3v0Z2siYRVo/result (54 bytes)
Executado pelo cliente: 165.227.76.211

root@sdi-master:~/sdi-docker# cat ./data_containers/PCwcWYvV5s9YZA0uhm5SQCzZwLs\
dEVqp5PtHU5QY8YEvq65ZKe1DS3v0Z2siYRVo/result
0 SDI já possui suporte a execução de contêineres!

```

Listagem 4.20 – Informações gerais da execução e conteúdo do arquivo de saída

A execução foi realizada com sucesso e o arquivo de saída possuía o valor desejado. Concluiu-se que esta funcionalidade estava realizando sua função corretamente.

4.4.5 Múltiplas execuções simultâneas

A execução de um contêiner ocorreu de forma desejada, passou-se ao segundo teste, que consistiu em verificar se o limite de contêineres em execução por máquina estava sendo respeitado. O arquivo de configurações `sdi.conf` foi configurado para limitar a execução de um contêiner por máquina. Em sequência, cinco contêineres foram inseridos ao SDI, o qual possuía somente uma máquina cliente disponível. A Listagem 4.21 exibe que os outros quatro contêineres ficaram na fila enquanto a execução do primeiro contêiner era realizada.

```
root@sdi-master:~/sdi-docker# ./resume.sh
Numero total de contêineres: 5
Na fila de envio: 4
Em execução: 1
Em reexecução: 0
Executados com sucesso: 0
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
-----
Máquinas Online: 1
Máquinas com suporte ao Docker: 1
```

Listagem 4.21 – Acompanhamento da execução dos contêineres

O teste foi realizado com sucesso. Concluiu-se que esta configuração do sistema estava funcionando corretamente.

4.4.6 Reexecução de contêineres

O bloco de controle de execuções também possui a tarefa de realizar a detecção de falhas de execução. Quando uma falha é detectada o bloco é responsável por analisar se o contêiner com problemas deve ser reenviado para outra máquina ou sua execução deve ser registrada como falha.

Visando testar esta funcionalidade, serão disponibilizadas duas máquinas clientes e será inserido um contêiner de teste. Enquanto o contêiner se encontrar em execução a máquina cliente será desligada, tornando-se indisponível pelo SDI. Automaticamente o contêiner deve ser enviado para a outra máquina disponível realizar sua execução. A Listagem 4.22 exibe o andamento deste teste.

```
root@sdi-master:~/sdi-docker# ./sdictl --resume
Numero total de contêineres: 1
Na fila de envio: 0
Em execução: 0
Em reexecução: 1
Executados com sucesso: 0
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
-----
Máquinas Online: 1
Máquinas com suporte ao Docker: 1

root@sdi-master:~/sdi-docker# ./sdictl --resume
Numero total de contêineres: 1
Na fila de envio: 0
Em execução: 0
Em reexecução: 0
Executados com sucesso: 1
Não executados (erro fatal): 0
Tempo de execução excedido (timeout): 0
-----
Máquinas Online: 1
Máquinas com suporte ao Docker: 1
```

Listagem 4.22 – Acompanhamento da execução dos contêineres

O teste de reenvio de contêiner foi realizado com sucesso. Concluiu-se que o bloco de controle de execuções estava funcionando conforme desejado.

4.4.7 Execução com tempo limite

O bloco de controle de execuções também é responsável por analisar se os contêineres em execução já extrapolaram seu tempo limite. Caso o tempo limite seja alcançado, o bloco deve analisar se o contêiner deve ser reenviado para outra máquina, caso seu limite de tentativas de execução também tenha sido extrapolado, o contêiner será finalizado e marcado com a informação de tempo limite excedido. A Listagem 4.23 exibe a inserção de um contêiner que demora 3 minutos para gerar a saída, porém, foi informado ao SDI que o contêiner deve retornar sua execução no tempo máximo de um minuto.

```

root@sdi-master:~/sdi-docker# ./sdictl --container=/root/sdi-docker/ \
tmp/container_teste/container --tl=60
Oocv6ByYgi0AhCyYVpWA3yGzZDpPciVr89SHammAxdxlAzflXeDtuggWDTM5lrDe

root@sdi-master:~/sdi-docker# ./sdictl --status=Oocv6ByYgi0AhCyYV\
pWA3yGzZDpPciVr89SHammAxdxlAzflXeDtuggWDTM5lrDe
Ticket: Oocv6ByYgi0AhCyYVpWA3yGzZDpPciVr89SHammAxdxlAzflXeDtuggWDTM5lrDe
Recebido em: Jun 3 06:15:13 UTC 2017
Inicializado em: Jun 3 06:15:42 UTC 2017
Finalizado em: Jun 3 06:16:57 UTC 2017
Tempo limite de execução (seg): 60
Status: TIMEOUT
Executado pelo cliente: 165.227.76.211

```

Listagem 4.23 – Envio do contêiner de teste e acompanhamento de sua execução

As informações da execução demonstraram que o contêiner teve sua execução finalizada após um minuto de execução. O bloco de controle de execuções estava realizando sua tarefa conforme desejado.

4.4.8 Problema de distribuição

A execução dos testes revelou um problema na distribuição dos contêineres para as máquinas clientes. Algumas máquinas estavam sendo muito utilizadas, executando diversos contêineres, enquanto outras não executaram nenhum contêiner. A Figura 22 mostra a distribuição das execuções.

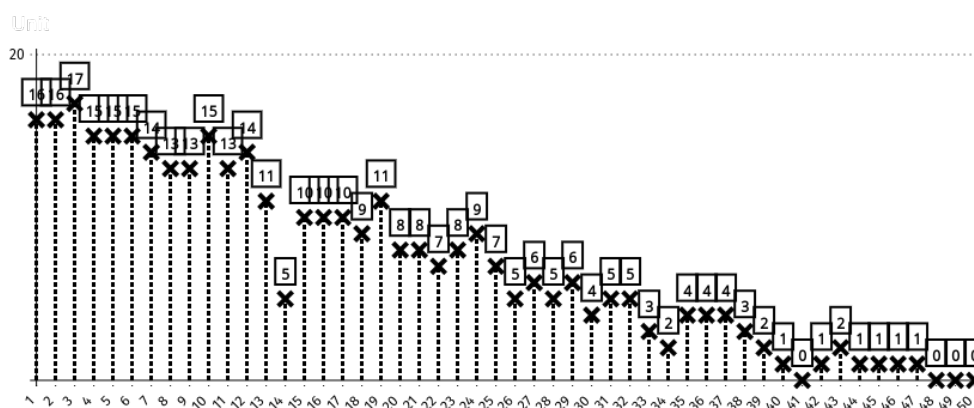


Figura 22 – Distribuição problemática de contêineres entre as máquinas

Fonte: Autoria própria

A análise da Figura 22 exibe que a seleção das máquinas era realizada conforme a sequência de inserção no SDI. As máquinas de número maior somente seriam utilizadas se todas as máquinas anteriores de sua sequência estivessem sendo totalmente utilizadas. Essa escolha era realizada através da listagem ordenada das máquinas. Constatado o problema, a programação do bloco de controle de execuções precisou ser alterada. A alteração do código buscou deixar a distribuição de contêineres de forma sequencial e não por varredura. A busca era inicializada na máquina 1, se a máquina estivesse disponível e houvessem contêineres para execução, ela seria utilizada. Caso a máquina não estivesse disponível a busca seguiria para a próxima máquina e verificaria se existem contêineres aguardando execução, e assim seguiria até a última máquina. Quando não existem contêineres disponíveis, a busca fica parada na máquina atual e não é mais reiniciada em cada envio. A Figura 23 exibe a distribuição das execuções após a alteração.

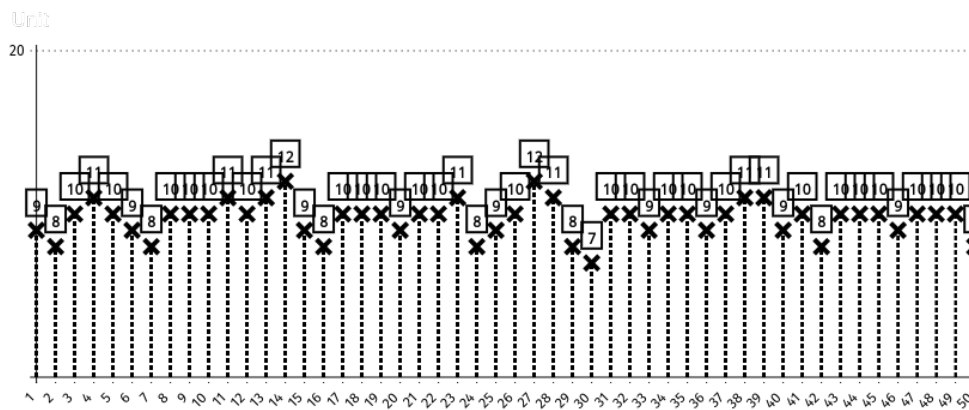


Figura 23 – Distribuição de contêineres entre as máquinas após ajustes

Fonte: Autoria própria

O novo código de distribuição de contêineres estava realizando uma distribuição mais justa entre as máquinas, fazendo uso de todas as máquinas disponíveis e não desperdiçando recursos.

5 EXPERIMENTOS

Este capítulo explana os experimentos realizados para teste do trabalho e seus resultados.

5.1 AMBIENTE DE EXECUÇÃO

A realização de experimentos requer um ambiente de execução com no mínimo duas máquinas, a primeira controlando o SDI e a segunda sendo utilizada como cliente. Foi estudada a possibilidade de utilizar as máquinas das salas de aula da UTFPR para a execução dos experimentos deste trabalho, porém, foram encontradas algumas dificuldades técnicas: Os horários de uso das salas são complexos; a configuração da rede de internet deveria sofrer algumas alterações; as máquinas deveriam passar por várias configurações para estarem disponíveis para o SDI e deveriam ser fornecidas as senhas de acesso de vários equipamentos, que não podem ser fornecidas para alunos por questões de segurança; A falta de acesso total aos equipamentos dificultaria qualquer correção de eventuais problemas, entre outros fatores. Além disso, as alterações de configurações da rede poderiam comprometer o correto funcionamento da infraestrutura da instituição, dessa forma foi decidido utilizar máquinas alugadas em uma infraestrutura de *Cloud Computing*.

Durante todo o desenvolvimento do trabalho foram utilizadas algumas máquinas alugadas na *Digital Ocean*. A *Digital Ocean* foi escolhida por oferecer um bom serviço e o menor custo de mercado em comparação com outras empresas do ramo, como *Google Cloud Computing* e *Amazon AWS*. A Figura 24 mostra os custos de aluguel de máquinas, por hora, conforme sua quantidade de memória, processamento e armazenamento.

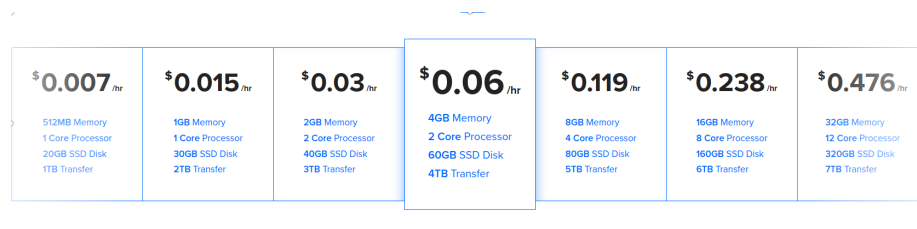


Figura 24 – Custo do aluguel de máquinas na *Digital Ocean*

Fonte: *Digital Ocean*

A máquina mais simples oferecida pela empresa tem um custo de \$0.007 dólares por hora, totalizando \$5 dólares por mês. Durante todo o desenvolvimento foram utilizadas 3 máquinas para a realização de testes básicos e auxílio no desenvolvimento do sistema. A primeira máquina era a máquina principal, a qual executava o SDI. Enquanto as outras duas máquinas foram configuradas como clientes da máquina principal, como pode ser vista na Figura 25.

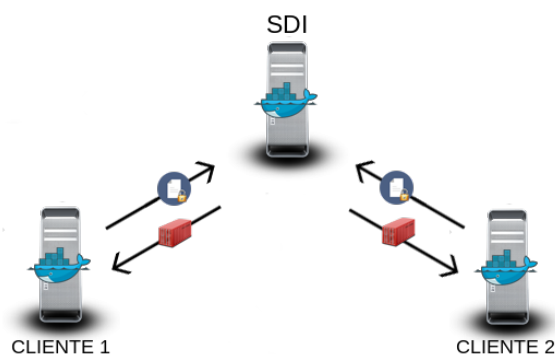


Figura 25 – Estrutura utilizada durante o desenvolvimento
Fonte: Autoria própria

Cada máquina utilizada possui um núcleo de processamento, 512MB de memória RAM e 20GB de armazenamento SSD. A *Digital Ocean* fornece um painel de controle para gerenciamento dos clouds com várias opções, como: aluguel de novas máquinas; edição de domínios da máquina; acesso ao console e destruição da máquina, que consiste em cancelar o aluguel do equipamento. A Figura 26 exibe este painel.

Name	IP Address	Created	Tags
sdi-client20 512 MB / 20 GB Disk / NYC3 - Ubuntu sdi-client-snap	165.227.76.106	1 hour ago	sdiclient
sdi-client19 512 MB / 20 GB Disk / NYC3 - Ubuntu sdi-client-snap	104.236.91.135	1 hour ago	sdiclient
sdi-client18 512 MB / 20 GB Disk / NYC3 - Ubuntu sdi-client-snap	165.227.76.105	1 hour ago	sdiclient
sdi-client17 512 MB / 20 GB Disk / NYC3 - Ubuntu sdi-client-snap	165.227.76.95	1 hour ago	sdiclient
sdi-client16 512 MB / 20 GB Disk / NYC3 - Ubuntu sdi-client-snap	165.227.76.102	1 hour ago	sdiclient

- Add a domain
- Access console
- Resize droplet
- View usage
- Enable backups
- Edit tags
- Destroy

Figura 26 – Painel de controle geral
Fonte: Digital Ocean

Além do painel de controle geral, a *Digital Ocean* oferece um painel de controle individual para cada máquina alugada, exibindo dados de: utilização do processador; utilização de memória; utilização de armazenamento e quantidade de tráfego da rede interna e pública. Cada máquina possui um endereço público que pode ser acessado de qualquer lugar da Internet e um endereço privado que pode ser acessado somente por máquinas que se encontrem na rede interna

do datacenter. A rede privada não possui limite de uso e possui elevada velocidade de transmissão, diferentemente da rede pública, que possui limites de utilização e sua velocidade é limitada através de vários fatores da Internet e a extrapolação do limite de uso gera custos adicionais. A Figura 27 exibe o painel de controle de uma máquina.



Figura 27 – Painel de controle de uma máquina

Fonte: *Digital Ocean*

O painel de controle das máquinas possui a opção para geração de imagens (*Snapshots*), sendo é possível criar uma máquina cliente totalmente configurada para ser utilizada pelo SDI, e em seguida criar uma imagem deste equipamento. Com a imagem criada o painel de controle oferece a opção de criar N máquinas com a mesma imagem. Esta opção se mostra interessante na criação do ambiente de execução, não sendo necessário realizar a configuração manual de todas as máquinas. A Figura 28 mostra uma etapa da criação de uma máquina utilizando uma imagem.

Create Droplets

Choose an image ?

Distributions One-click apps **Snapshots** Backups

Ubuntu sdi-client-snap
 Ubuntu app-construfacil-alpha-0...

Choose a size

\$5/mo \$0.007/hour	\$10/mo \$0.015/hour	\$20/mo \$0.030/hour	\$40/mo \$0.060/hour	\$80/mo \$0.119/hour
512 MB / 1 CPU 20 GB SSD disk 1000 GB transfer	1 GB / 1 CPU 30 GB SSD disk 2 TB transfer	2 GB / 2 CPUs 40 GB SSD disk 3 TB transfer	4 GB / 2 CPUs 60 GB SSD disk 4 TB transfer	8 GB / 4 CPUs 80 GB SSD disk 5 TB transfer

Select additional options ?

Private networking
 Backups
 IPv6
 User data
 Monitoring

Add your SSH keys ?

New SSH Key
 sdi-test2
 sdi-master-ssh-key
 Arch Virtual 2

Finalize and create

How many Droplets?

Deploy multiple Droplets with the same [configuration](#).

Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name only contain alphanumeric characters, dashes, and periods.

sdi-client1

sdi-client2

sdi-client3

[Add Tags](#)

Create

Figura 28 – Criação de uma máquina baseada em uma imagem

Fonte: *Digital Ocean*

A estrutura de operação oferecida pelos painéis de controle da *Digital Ocean* oferecem opções que facilitam a criação e gerenciamento de um ambiente de execução distribuída, atendendo todas as necessidades para a realização de experimentos utilizando o sistema deste trabalho.

5.2 EXECUÇÃO DE FÓRMULAS

O desenvolvimento do Sistema de Diagnóstico Instantâneo com suporte a execução de contêineres foi realizado buscando a criação de um ambiente de execução genérico, e não um ambiente para a execução de uma aplicação específica. Visando testar a funcionalidade do sistema e obter resultados gerais foi realizado um experimento através da execução de fórmulas pseudo-Boolean Optimization (PBO).

Os resolvidores de Boolean Satisfiability (SAT) e Integer Linear Programming (ILP) são utilizados para encontrar a solução e/ou otimização de desafios e problemas do mundo real (RIBAS et al., 2013). Um resolvidor utilizado para a execução de fórmulas PBO é o Clasp, oferecendo diversos modos de operação e configuração (POTASCO, 2017). A configuração e modos de operação do Clasp não serão apresentados neste trabalho.

O experimento do sistema irá realizar o processamento de 374 fórmulas, das quais 50 foram disponibilizadas juntamente com a publicação "PBFVMC: A new pseudo-Boolean formulation to virtual-machine consolidation"(RIBAS et al., 2013). Todas as fórmulas já se encontravam no formato PBO, formato de execução que pode ser executado pelo Clasp.

O tempo de execução de uma fórmula pseudo-Boolean pode ser variável, podendo ser decidida com sucesso em alguns segundos ou em até horas. Segundo informações obtidas referente as fórmulas do experimento, a grande maioria deveria ser decidida em um período menor de uma hora, desta forma, o tempo limite de execução foi definido como sendo uma hora.

5.2.1 Preparação do ambiente

A estrutura para realizar o processamento foi escolhida conforme a disponibilidade de recursos financeiros. Foram locadas 100 máquinas na *Digital Ocean*, com a opção de máquina mais simples. O somatório de recursos de todas as máquinas forneciam 100 núcleos de processamento, 50 GB de memória RAM e 2 TB de armazenamento SSD.

A imagem utilizada para as execuções das fórmulas não pode ser utilizada para ambas as execuções, pois é necessário alterar um comando do arquivo `inicia.sh` na imagem utilizada para a execução por pacotes de execução, além disto, a execução por pacotes necessita que seja criado o arquivo `command`, conforme demonstrado no Capítulo 4. A programação do `Dockerfile` pode ser utilizada na criação da ambas as imagens, as quais serão baseadas na distribuição Ubuntu. A única operação do `Dockerfile` foi realizar a instalação do aplicativo Clasp na imagem base, o qual é necessário para realizar as execuções. A Listagem 5.1 exibe todas as instruções contidas no `Dockerfile`.

```

# seleção da distribuição base
FROM ubuntu:latest

# atualiza os repositórios e instala o clasp
RUN apt-get update && apt-get install -y apt-utils clasp

# copia o arquivo inicia.sh para a raiz da imagem
COPY inicia.sh /

# altera as permissões do arquivo, autorizando sua execução
RUN chmod 755 /inicia.sh

# informa o programa iniciado quando o contêiner for inicializado
CMD /bin/bash inicia.sh

```

Listagem 5.1 – Dockerfile para a geração da imagem de execução das fórmulas

O arquivo `inicia.sh` não necessita realizar muitos procedimentos, somente irá funcionar como gatilho para iniciar a execução do Clasp e informar onde a saída da execução deve ser registrada. A Listagem 5.2 exibe as instruções comentadas do arquivo `inicia.sh`, utilizado nas imagens da execução por pacotes.

```

#!/bin/bash

# carrega o ticket para salvar a saída corretamente
TICKET="$(cat /ticket.txt)"

# executa o Clasp e grava seu resultado no arquivo de saída padrão
clasp /formula.pbs >> /compartilhada/$TICKET

# finaliza a execução do contêiner
exit 0

```

Listagem 5.2 – Programação do arquivo `inicia.sh` para execução das fórmulas

O arquivo `inicia.sh` para a execução por contêineres fechados será semelhante. A linha que carrega o valor do ticket do arquivo `ticket.txt` foi removida e o valor do ticket foi passado para a execução por parâmetro.

Na sequência, as imagens foram criadas, as quais foram geradas com o nome `sdi:formulas` e `sdi:formulastar`. Em seguida foram criados os arquivos para a realização das execuções. A geração dos 374 contêineres fechados necessitou de 7 minutos e 20 segundos para criação, utilizando mais de 61GB de armazenamento, que foram distribuídos e processados pelas máquinas clientes.

A criação dos pacotes de execução requer a criação do arquivo `command`, exibido e comentado na Listagem 5.3. Este arquivo é o gatilho de disparo da execução, foi responsável por criar o contêiner baseado na imagem criada anteriormente e inicializar sua execução. A criação dos pacotes de execução necessitou de 3 minutos e 11 segundos (menos da metade do tempo necessário para a criação dos contêineres fechados) e utilizou 646MB de armazenamento (aproximadamente 99% menos espaço).

```
#!/bin/bash
PREFIX=$(dirname $0)

# cria o container utilizando a imagem do docker registry
docker create -v /data:/compartilhada --name $1 -ti \
10.135.22.170:5000/formulastar /inicia.sh

# copia os arquivos para dentro do container criado
docker cp $PREFIX/formula.pbs $1:/formula.pbs
echo $1 > $PREFIX/ticket.txt
docker cp $PREFIX/ticket.txt $1:/ticket.txt

# executa o container
docker start $1
```

Listagem 5.3 – Programação do arquivo `command` para a criação dos pacotes de execução

A Listagem 5.3 exibe um endereço de IP na definição da imagem do contêiner, este endereço representa o endereço do *Docker Registry* que é executado por um contêiner na máquina do SDI.

5.2.2 Execução por contêineres fechados

Visando testar a integridade e estabilidade das execuções, foram realizadas três execuções de todas as fórmulas. Através das três execuções foi possível verificar se os resultados se mantiveram constantes e o tempo de execução de cada fórmula não sofreu grandes diferenças, pois como todas as máquinas clientes são semelhantes, a variação entre as execuções deveria ser mínima, mesmo as máquinas sendo *clouds* virtualizados em servidores, o desempenho oferecido pela empresa hospedeira deveria ser constante.

O funcionamento esperado neste experimento deveria retornar somente dois estados de execução: Tempo excedido (onde a fórmula não pode ser executada em uma hora) ou executado com sucesso. A execução de uma fórmula pode retornar que a mesma é SAT ou UNSAT, mas o resultados das fórmulas não fazem parte deste trabalho. Todos os contêineres foram inseridos três vezes no sistema e seus tickets foram arquivados para conferência futura.

A utilização da rede durante a simulação foi elevada, a Figura 29 exibe que ocorreram picos de transferência de quase 51 MB/s na máquina do SDI, esse tráfego consiste no envio dos contêineres para as máquinas clientes. Pode ser notado que ocorreu uma pausa de aproximadamente uma hora no envio dos contêineres, este intervalo se refere ao período onde todas as máquinas executaram as fórmulas que não podiam ser resolvidas no período máximo. A realização deste experimento utilizando máquinas conectadas via Internet seria inviável.

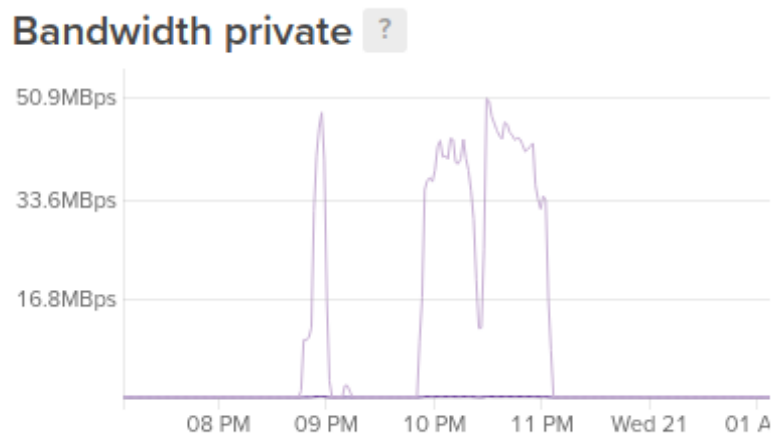


Figura 29 – Transferência de rede durante execução do experimento
Fonte: Painel de controle da Digital Ocean

A execução tripla das fórmulas foi realizada em 3 horas, 1 minuto e 1 segundo. Não foram encontradas diferenças de resultados entre as três execuções, as fórmulas que não foram executadas em uma hora retornaram TIMEOUT em todas as suas execuções. Do total de 374 fórmulas, 335 foram executadas com sucesso, enquanto 39 não puderam ser executadas no período limite. A Listagem 5.4 exibe a verificação das execuções.

```

root@sdi-master:~/# ./resultados.sh
Fórmulas com o mesmo resultado: 374
Fórmulas com o resultado diferente: 0
-----
Executadas com sucesso: 335
Timeout: 39

```

Listagem 5.4 – Verificação dos resultados

A distribuição dos 1122 contêineres entre as máquinas foi realizada como mostrado no gráfico da Figura 30.

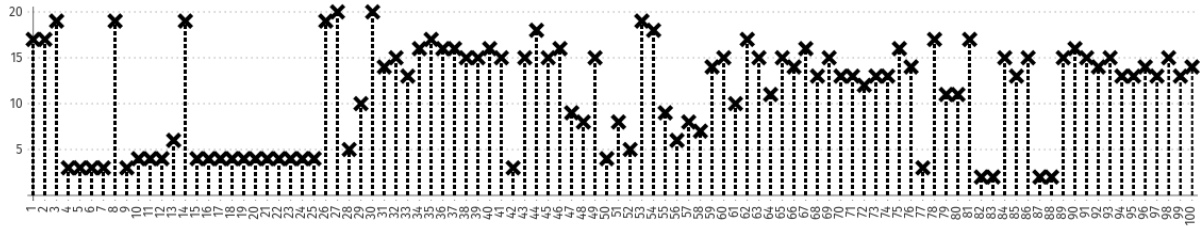


Figura 30 – Distribuição de contêineres entre as máquinas clientes

Fonte: Autoria própria

A Figura 30 exibiu que algumas máquinas executaram mais contêineres que outras, essa informação pode não significar que umas máquinas ficaram ociosas, enquanto outras estavam realizando processamento. Para verificar se as máquinas estavam sendo utilizadas foram analisadas algumas informações das execuções. A primeira informação necessária é o tempo total de utilização da máquina, que pode ser adquirida através da Equação 5.1. Onde TF_N representa o tempo final de execução do último contêiner (N) recebido e TI_1 representa o tempo inicial de execução do primeiro contêiner recebido. O termo $[s]$ indica que a unidade física da resposta é dada em segundos, conforme definido pelo padrão do Sistema Internacional.

$$T_s = TF_N - TI_1[s] \quad (5.1)$$

O tempo total da simulação e o tempo em que a máquina estava realizando a execução de um contêiner podem informar o aproveitamento do equipamento, através do tempo em que a máquina se encontrou no estado de ociosidade. A Equação 5.2 pode ser utilizada para calcular o tempo de uso de uma máquina. Onde TF_k representa o tempo final da execução do contêiner k ; TI_k representa o tempo inicial da execução do contêiner k e x representa o número total de contêineres executados pela máquina.

$$T_E = \sum_{k=1}^x TF_k - TI_k[s] \quad (5.2)$$

As informações obtidas com as equações de tempo de simulação e tempo de execução podem ser utilizadas para realizar o cálculo do tempo em que a máquina ficou ociosa, ou seja, não estava realizando a execução de nenhum contêiner. A Equação 5.3 pode ser utilizada para realizar a aquisição do tempo de ociosidade.

$$T_o = T_s - T_E[s] \quad (5.3)$$

As informações obtidas através das equações fornecem a possibilidade de calcular a porcentagem de aproveitamento das máquinas, onde ela estava realizando a execução de pelo menos um contêiner. A Tabela 2 exhibe um resumo das informações das execuções que se encontram no Apêndice A.

Tabela 2 – Estatísticas da execução por contêineres fechados

Tempo total da simulação:	3 horas, 1 minuto e 1 segundo
Tempo médio utilização das máquinas:	2 horas e 9 minutos
Tempo médio de ociosidade:	31 minutos
Porcentagem de utilização:	76.35%
Tráfego de rede gerado:	69.4 GB
Tempo médio de espera do contêiner:	1 hora, 28 minutos e 16 segundos

As máquinas obtiveram uma baixa utilização, durante as 3 horas, 1 minuto e 1 segundo de experimento. As máquinas foram utilizadas por cerca de 2 horas e 9 minutos, dos quais, 31 minutos foram de ociosidade, onde a máquina estava parada, aguardando algum contêiner para ser executado, resultando em uma porcentagem de 76.35% de utilização.

O tempo médio de espera dos contêineres foi elevado, pois quando o sistema era inicializado já existiam vários contêineres na fila, os quais eram divididos entre as 100 máquinas clientes. Além de existirem vários contêineres na fila, a distribuição dos contêineres agravou a situação, pois cada contêiner possuía um tamanho de aproximadamente 200MB, tornando a distribuição lenta. O primeiro contêiner da fila foi inicializado em 8 segundos, enquanto o último contêiner ficou cerca de 2 horas e 11 minutos na fila de espera, gerando uma média de 1 hora e 28 minutos de espera.

A execução por contêineres fechados requereu a criação de alguns scripts adicionais para realizar o envio dos contêineres, pois não havia espaço de armazenamento disponível para realizar a inserção de todos os contêineres na fila do SDI, desta forma os contêineres foram inseridos em lotes. A Figura 31 exibe a utilização do disco durante o experimento.



Figura 31 – Utilização do disco durante o experimento

Fonte: Painel de controle da *Digital Ocean*

5.2.3 Execução por pacotes de execução

A execução por pacotes de execução consiste no recebimento de um arquivo compactado contendo arquivos e comandos necessários para a criação e execução de um contêiner. Todas as imagens que o pacote necessite serão transferidas do *Docker Hub* ou do repositório de imagens do próprio SDI, criado através do *Docker Registry*. A execução de um contêiner através de pacotes de execução faz uso das propriedades do Docker e seu sistema de camadas.

A Figura 32 exibe dois picos de transmissão de dados pela rede. O primeiro, e menor, pico foi o registro de quatro minutos de transferência, onde todas as 100 máquinas estavam acessando o SDI e baixando a imagem necessária para as execuções, as máquinas já possuíam a imagem do Ubuntu, logo não foi necessário reenviar toda a imagem da fórmula, mas somente a camada que possuía a instalação do Clasp acima da imagem base do Ubuntu. O tráfego de dados chegou ao valor de 31.53 MBps. Este acesso precisa ser realizado somente na execução do primeiro pacote.

O segundo pico foi o registro de 8 minutos de transferência, onde todas as 100 máquinas estavam acessando o SDI para baixar a imagem necessária para as execuções, este segundo tráfego de dados foi maior em relação o primeiro, pois a imagem do Ubuntu foi excluída antes do início das execuções, logo foi necessário reenviar toda a imagem das fórmulas, juntamente com a imagem base do Ubuntu. Este segundo pico chegou ao valor de 38.33 MBps.

O tráfego posterior foi gerado durante a execução de todos os pacotes de execução. O pico mais elevado de transferência foi especificamente de 1.71 MBps.

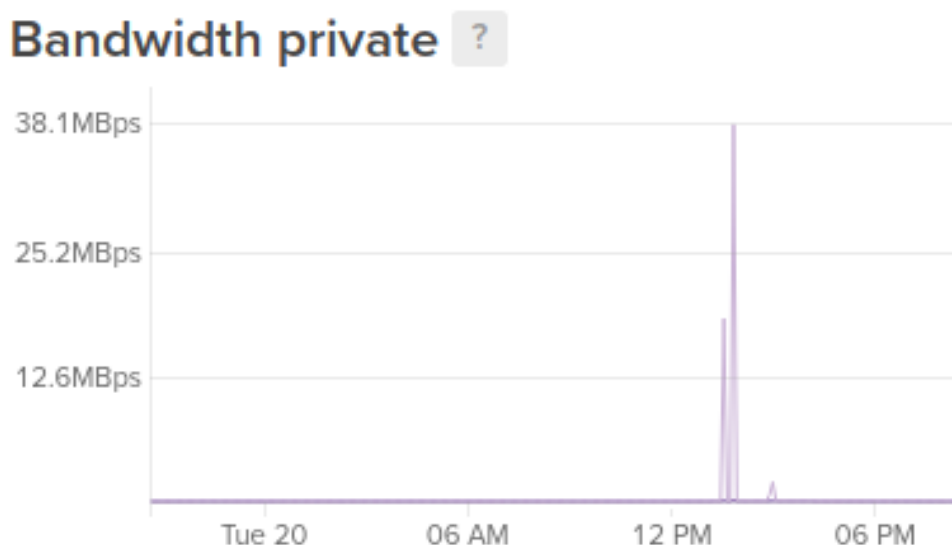


Figura 32 – Tráfego de rede gerado durante o experimentos com os pacotes de execução
Fonte: Painel de controle da *Digital Ocean*

A execução tripla das fórmulas foi realizada em 2 horas, 17 minutos e 30 segundos. Não foram encontradas diferenças de resultados entre as três execuções, as fórmulas que não

foram executadas em uma hora retornaram TIMEOUT em todas as suas execuções. Do total de 374 fórmulas, 335 foram executadas com sucesso, enquanto 39 não puderam ser executadas no período limite. A Listagem 5.5 exibe a verificação das execuções.

```

root@sdi-master:~/# ./resultados.sh
Fórmulas com o mesmo resultado: 374
Fórmulas com o resultado diferente: 0
-----
Executadas com sucesso: 335
Timeout: 39

```

Listagem 5.5 – Verificação dos resultados

A distribuição dos 1122 pacotes foi realizada entre as máquinas como mostrado no gráfico da Figura 33.

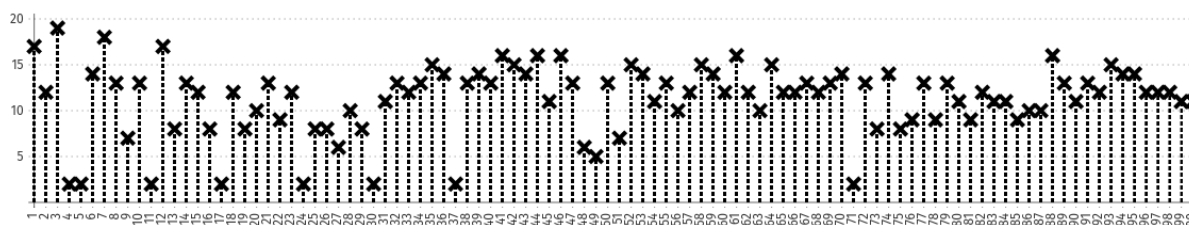


Figura 33 – Distribuição de pacotes de execução entre as máquinas clientes

Fonte: Autoria própria

As equações descritas na execução por contêineres fechados também podem ser utilizadas para realizar a aquisição de dados desta execução. As informações obtidas através das equações forneceram os dados do Apêndice B que se encontram resumidos na Tabela 3.

Tabela 3 – Estatísticas da execução por pacotes de execução

Tempo total da simulação:	2 horas, 17 minutos e 30 segundos
Tempo médio utilização das máquinas:	1 hora, 31 minutos e 39 segundos
Tempo médio de ociosidade:	6 minutos e 17 segundos
Porcentagem de utilização média:	93.57%
Tráfego de rede gerado:	19.2 GB
Tempo médio de espera do contêiner:	1 hora, 5 minutos e 18 segundos

As máquinas obtiveram uma taxa de utilização mais próxima do 100% desejado, durante as 2 horas e 17 minutos do experimento. As máquinas foram utilizadas por cerca de 1 hora e 31 minutos, dos quais, a média de ociosidade ficou em 6 minutos e 17 segundos, resultando em uma porcentagem de 93.57% de utilização.

Os pacotes de execução possuem um tamanho reduzido em comparação com o contêiner fechado, contendo somente o arquivo de comando e a fórmula que será executada, esta característica fez com que os contêineres fossem distribuídos com maior agilidade. O primeiro

contêiner da fila foi inicializado em 8 segundos, enquanto o último contêiner ficou 1 hora, 27 minutos e 31 segundos na fila de espera.

O tráfego de rede calculado se refere somente ao envio da imagem e dos pacotes de execução, ignorando todo o tráfego de rede gerado para realizar a comunicação entre as máquinas. A execução foi realizada sem dificuldades e sem a necessidade da criação de nenhum script auxiliar. A Figura 34 exibe a utilização do disco durante o experimento.

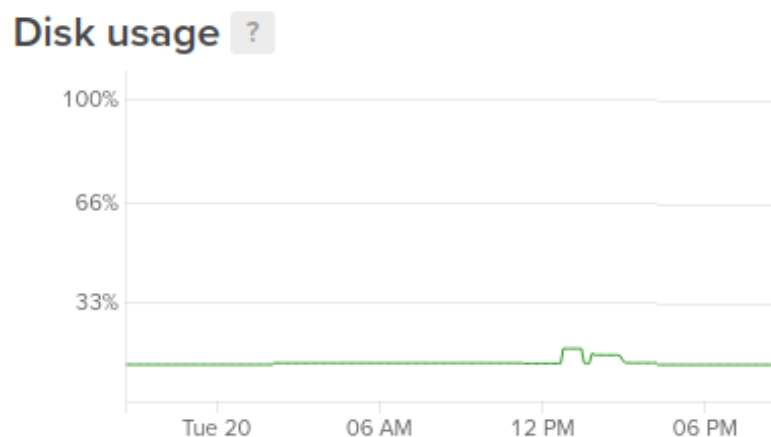


Figura 34 – Utilização do disco durante o experimento

Fonte: Painel de controle da *Digital Ocean*

5.2.4 Discussão

As primeiras execuções das fórmulas exibiram erros fatais, indicando que existia algum problema no sistema, nos contêineres e/ou nas máquinas clientes. Analisando os registros de erros do sistema das máquinas foi identificado que a execução das fórmulas não estava sendo realizada pois o processo do Clasp era encerrado automaticamente por excesso de utilização de memória RAM. A máquina cliente possuía somente 512 MB de memória RAM e era compartilhada entre o sistema operacional, o Docker e o Clasp. Esta quantidade de memória impedia a execução das fórmulas na estrutura montada.

A solução do problema de memória consistiu na criação de um arquivo de swap. Dos 20 GB de armazenamento disponíveis, 7 GB foram reservados para utilização como swap de memória. Desta forma cada máquina teria simbolicamente 7.5 GB de memória RAM. Os comandos para a criação do arquivo de Swap e sua ativação se encontram na Listagem 5.6, os comandos foram enviados para as máquinas através do próprio SDI.

```

root@sdi-master:~/sdi-docker# echo falldate -l 7G /swap >> cmds/general
root@sdi-master:~/sdi-docker# echo chmod 600 /swap >> cmds/general
root@sdi-master:~/sdi-docker# echo mkswap /swap >> cmds/general
root@sdi-master:~/sdi-docker# echo swapon /swap >> cmds/general

```

Listagem 5.6 – Criação do Swap nas máquinas clientes

Com o arquivo de Swap as execuções foram realizadas com sucesso. Em alguns momentos a execução de uma fórmula chegou a utilizar mais de 3 GB do swap, mostrando que caso esta solução não funcionasse, seria necessário utilizar máquinas clientes com maior quantidade de recursos. A utilização do swap diminuiu o desempenho das execuções, porém, como a medição do tempo das execuções não era o foco do experimento, esta queda de desempenho foi ignorada.

A diferença entre as duas formas de execução já se mostraram perceptíveis desde o momento da criação dos arquivos de execução, onde a diferença de utilização de espaço na criação dos contêineres e pacotes foi de 99%. Neste ponto o sistema de camadas do Docker mostra sua vantagem de uso, pois cada contêiner fechado possui a imagem completa do sistema; os aplicativos do sistema; o Clasp e a fórmula que foi executada, enquanto um pacote de execução contém somente um pequeno arquivo de comando informando qual imagem deve ser utilizada e a fórmula que foi executada. A Tabela 4 exibe um resumo de informações de ambas as execuções.

Tabela 4 – Informações da execução por contêineres fechados e por pacotes de execução

	Contêineres Fechados	Pacotes de execução
Tempo total da simulação:	3 horas, 1 minuto e 1 segundo	2 horas, 17 minutos e 30 segundos
Tempo médio utilização das máquinas:	2 horas e 9 minutos	1 hora e 31 minutos e 39 segundos
Tempo médio de ociosidade:	31 minutos	6 minutos e 17 segundos
Porcentagem de utilização:	76.35%	93.57%
Tráfego de rede gerado:	69.4 GB	19.2 GB
Tempo médio de espera do contêiner:	1 hora, 28 minutos e 16 segundos	1 hora, 5 minutos e 18 segundos

A diferença do tempo médio de ociosidade e porcentagem de utilização exibem uma grande diferença de uso dos equipamentos, onde 23.65% dos recursos das máquinas foram desperdiçados na execução por contêineres fechados. Estes recursos poderiam estar realizando o processamento de outros dados, porém, ficaram ociosos, enquanto que na execução por pacotes de execução ocorreu uma utilização de 93.57% dos equipamentos, um valor mais próximo dos 100% desejados e maior do que comparado com a execução por contêineres fechados.

A diferença na utilização do armazenamento pode ser vista na Figura 35. Durante a execução por contêineres fechados o armazenamento do SDI permaneceu em 93% de utilização, onde os contêineres foram inseridos gradativamente em lotes. Enquanto que a inserção de todos os pacotes de execução ao mesmo tempo não chegaram a utilizar 16% do armazenamento.



Figura 35 – Diferença de utilização de armazenamento as execuções

Fonte: Adaptado do painel de controle da *Digital Ocean*

O sistema mostrou um nível de confiabilidade e estabilidade das execuções. Mesmo com uma variação média de 38.99% no tempo de execução das fórmulas. A visualização do Apêndice C exibe que os tempos de execução foram estáveis, como por exemplo, na fórmula número 14, ocorreu uma variação de 30% entre as seis execuções, verificando os dados de execução da fórmula 14 no Apêndice C, verifica-se que os tempos de execução foram respectivamente: 0.002 segundos; 0.002 segundos; 0.002 segundos; 0.001 segundos; 0.001 segundos e 0.001 segundos. As seis execuções foram realizadas praticamente no mesmo período de tempo, mesmo com o valor de sua variação sendo elevado. A variação elevada também se deve ao fato de que alguns tempos de execuções foram realizados em milissegundos, onde pequenas diferenças de tempo resultam em uma elevada variação.

Mesmo com os algoritmos de distribuição buscando o envio mais igualitário possível, algumas máquinas executaram menos contêineres do que outras. Estas máquinas executaram menos contêineres, pois receberam as fórmulas que não foram executadas no tempo limite de uma hora, como os experimentos foram realizados no período de duas e três horas, essas máquinas não puderam executar mais do que dois ou três contêineres.

A menor utilização das máquinas e o maior tempo de execução do experimento por contêineres fechados, se deve ao fato que o distribuidor de contêineres não era capaz de enviar os contêineres com a velocidade necessária para manter as 100 máquinas em execução. O processo de verificação das execuções liberava as máquinas mais rapidamente do que o distribuidor, o que

gerou a diferença nas execuções. A Figura 36 exibe a diferença de tráfego entre os dois tipos de execução, durante o experimento com os contêineres fechados a rede foi muito mais utilizada.



Figura 36 – Diferença de utilização da rede durante as execuções

Fonte: Adaptado do painel de controle da *Digital Ocean*

O tempo total das execuções poderia ser otimizado caso a fórmula 24 fosse executada no início dos experimentos, pois como pode ser visualizado no Apêndice C, esta fórmula não é decidida no período de uma hora. Logo, quando o experimento se encaminha para sua conclusão, o SDI inicializa uma execução que irá demorar uma hora, aumentando o tempo total do experimento, pois somente esta máquina estava em execução.

Os dados levantados em conjunto com os apêndices A, B, C e D exibiram que a utilização da virtualização por contêineres do Docker é uma grande ferramenta. Essa nova versão do Sistema de Diagnóstico Instantâneo se mostrou uma ferramenta estável, que pode ser utilizada para as mais diversas finalidades.

6 CONCLUSÃO

O presente trabalho buscou a criação de um ambiente de execução distribuída, o qual seria controlado pelo Sistema de Diagnóstico Instantâneo com suporte a execução de contêineres e sua utilização pudesse ser realizada de forma genérica, possibilitando a execução de dados das mais diversas finalidades.

Desenvolver este projeto foi um desafio em diversos pontos, primeiramente o sistema base desde projeto era desconhecido do autor, além do sistema, a linguagem *Bash* também foi uma novidade. A falta de documentação deixou o processo ainda mais desafiador, mas foi um problema que depois de resolvido, facilitou o desenrolar de todo o desenvolvimento.

O sistema final se mostrou satisfatório ao seu planejado, mesmo sendo utilizado para testes específicos, sua utilização atende as mais diversas necessidades do setor computacional. Fornecendo uma ferramenta para realizar execução de problemas, desde processar dados, realizar simulações e qualquer tarefa que possa necessitar de uma elevada quantidade de processamento e/ou memória.

Empresas, instituições privadas e públicas podem economizar recursos financeiros através deste projeto, pois ele exclui a necessidade de investimento de recursos na aquisição de grandes centros de processamento, sendo possível realizar a locação de um número específico de máquinas em sistemas de *Cloud Computing* e pagar somente pelo seu tempo de utilização.

A ferramenta possui funcionalidades que provavelmente atendam todas as especificações e necessidades de seus utilizadores, porém, durante o desenvolvimento do projeto algumas ideias surgiram, possibilitando que sejam realizados trabalhos futuros, como: aumentar alguns níveis de verificações e isolamentos do sistema; possibilitar que seja informado um diretório de contêineres de uma única vez; limitar a quantidade de execuções de contêineres conforme a porcentagem de utilização da máquina cliente; atualizar o sistema para suportar a execução de múltiplas instâncias de envio e controle de execuções, buscando manter 100% das máquinas clientes realizando uma execução e possibilitar que o SDI possa realizar o acionamento e desligamento automático de máquinas clientes conforme necessidade.

O sistema final fornece várias funcionalidades e diferentes opções de utilização, esta variedade possibilita que a grande maioria das necessidades dos utilizadores possam ser atendidas. O utilizador é livre para organizar e realizar a execução de seus contêineres.

REFERÊNCIAS

- BELLARE, M.; KOHNO, T.; NAMPREMPRE, C. **RFC4344**. [S.l.], 2006. [Acessado em 17 de agosto de 2016]. Disponível em: <<http://www.ietf.org/rfc/rfc4344.txt>>. 29
- CARR, Nicholas G. **The big switch: Rewiring the world, from Edison to Google**. [S.l.]: WW Norton & Company, 2008. 14
- CHANDRA, T.; TOUEG, S. **Unreliable failure detectors for reliable distributed systems**. [S.l.]: Journal of the ACM. Vol 43, No. 2, pp. 225–67, 1996. 16
- COULOURIS, George et al. **Sistemas distribuídos: Conceitos e projeto**. 5. ed. [S.l.]: Editora Bookman, 2013. 13, 16, 17, 18
- DHAKATE, Suchit; GODBOLE, Anand. Distributed cloud monitoring using docker as next generation container virtualization technology. **IEEE INDICON 2015**, 2015. 13, 26
- DOCKER. **Docker Registry**. [S.l.], 2016. [Acessado em 15 de novembro de 2016]. Disponível em: <<https://docs.docker.com/registry/>>. 26
- DOCKER. **DOCKER: What is Docker**. [S.l.], 2016. [Acessado em 15 de agosto de 2016]. Disponível em: <<https://www.docker.com/what-docker>>. 15, 26
- DOCKER. **Understanding Docker Data Storage and Persistence**. [S.l.], 2016. [Acessado em 17 de outubro de 2016]. Disponível em: <<https://www.docker.com/sites/default/files/Understanding-Docker-Data-Storage-WP-rev3.pdf>>. 25
- GUPTA, K.; SAINI, J. K. Novel approach for distributed file system with multiple layers of fault tolerance. In: **Computing, Communication Automation (ICCCA), 2015 International Conference on**. [S.l.: s.n.], 2015. p. 616–619. 19
- HEIDEMANN, John S.; POPEK, Gerald J. File-system development with stackable layers. **ACM Transactions on Computer Systems, Vol 12, p.58-89**, 1994. 19
- LAUREANO, Marcos. **Máquinas virtuais e emuladores: conceitos, técnicas e aplicações**. [S.l.]: Novatec Editora, 2006. 21, 22, 23, 24, 25
- LXC. **Lxc - linux containers**. [S.l.], 2016. [Acessado em 26 de novembro de 2016]. Disponível em: <<https://linuxcontainers.org/lxc/introduction/>>. 26
- MORENO, Edward David; PEREIRA, Fábio Dacêncio; CHIARAMONTE, Rodolfo Barros. **Criptografia em software e hardware**. [S.l.]: São Paulo: Novatec, 2005. 29, 30
- OCEAN, Digital. **Digital Ocean**. [S.l.], 2017. [Acessado em 16 de junho de 2017]. Disponível em: <<https://www.digitalocean.com/>>. 14
- OPENVZ. **Openvz linux containers**. [S.l.], 2016. [Acessado em 26 de novembro de 2016]. Disponível em: <<http://openvz.org>>. 25
- POTASCO. **Clasp**. 2017. <<https://potassco.org/clasp/>>. [Acessado em 12 de junho de 2017]. 67

- PREETH, E. N. et al. Evaluation of docker containers based on hardware utilization. **2015 International Conference on Control, Communication and Computing India (ICCC)**, 2015. 26
- QUIGLEY, David et al. Unionfs: User- and community-oriented development of a unification file system. **Proceedings of the Linux Symposium, Vol 2**, p.349-362, 2006. 19, 20
- REESER, Jon; JONKOWSKI, Thomas; KEMPER, Greg M. **Maintaining HMI and SCADA Systems Through Computer Virtualization**. [S.l.]: IEEE Journals and Magazines, 2015. p 2558–2564. 13
- RIBAS, Bruno C. et al. Sdi - sistema de diagnóstico instantâneo. **WSL**, 2009. 13, 14, 29
- RIBAS, B. C. et al. Pbfvmc: A new pseudo-boolean formulation to virtual-machine consolidation. In: **2013 Brazilian Conference on Intelligent Systems**. [S.l.: s.n.], 2013. p. 201–206. 67
- RIBAS, Bruno C. et al. On modelling virtual machine consolidation to pseudo-boolean constraints. **IBERAMIA** p.361-370, 2012. 13
- RIZKI, Rizki; RAKHMATSYAH, Andrian; NUGROHO, M. Arief. Performance analysis of container-based hadoop cluster : Openvz and lxc. **International Conference on Information and Communication Technologies (ICoICT)**, 2016. 26
- RUOSO, Vinicius Kwiecien. Uma estratégia de testes logarítmica para o algoritmo hi-adsd. 2013. 38
- SALLOU, Olivier; MONJEAUD, Cyril. Go-docker: A batch scheduling system with docker containers. **IEEE International Conference on Cluster Computing**, 2015. 15
- SEED. **Secretaria de Estado da Educação**. 2017. <<http://www.sdi.seed.pr.gov.br/>>. [Acessado em 07 de julho de 2017]. 29
- SERVICES, Amazon Web. **O que é o Docker?** [S.l.], 2017. [Acessado em 16 de junho de 2017]. Disponível em: <<https://aws.amazon.com/pt/docker/>>. 25
- SHIREY, Russell G. et al. Analysis of implementations to secure git for use as an encrypted distributed version control system. **48th Hawaii International Conference on System Sciences**, 2015. 13
- STUBBS, J.; MOREIRA, W.; DOOLEY, R. Distributed systems of microservices using docker and serfnode. In: **Science Gateways (IWSG), 2015 7th International Workshop on**. [S.l.: s.n.], 2015. p. 34–39. 25
- TANENBAUM, Andrew S. **Sistemas operacionais modernos**. 3. ed. [S.l.]: Pearson, 2010. 19
- TANENBAUM, Andrew s.; STEEN, Marten van. **Sistemas distribuídos: princípios e paradigmas**. 2. ed. [S.l.]: São Paulo, SP: Pearson Prentice Hall, 2007. 16
- VASCONCELOS, P. Magalhaes; FREITAS, G. Azevedo de Araujo. Performance analysis of hadoop mapreduce on an opennebula cloud with kvm and openvz virtualizations. **Internet Technology and Secured Transactions (ICITST)** p.471-476, 2014. 25
- VERAS, Manoel. **Cloud Computing: nova arquitetura da TI**. [S.l.]: Brasport, 2012. 14

YLONEN, T. **RFC4251**. [S.l.], 2006. [Acessado em 17 de agosto de 2016]. Disponível em: <<http://www.ietf.org/rfc/rfc4251.txt>>. 29

ZAMPIERI, André. **Posicionamento de Réplicas em Sistemas Distribuídos**. Dissertação (Mestrado) — Universidade do Rio Grande do Sul - UFRGS, 2001. 18

Apêndices

APÊNDICE A – EXECUÇÃO POR CONTÊINERES FECHADOS

Este Apêndice contém as informações do experimento realizado com os contêineres fechados. A coluna cliente representa a máquina cliente utilizada; a coluna Tempo Total representa o tempo, em segundos, entre o início da primeira execução e a finalização da última execução; a coluna Tempo executando representa quanto tempo em segundos a máquina estava realizando a execução de contêineres; a coluna Tempo Ocioso representa a diferença entre o tempo total e o tempo executando; a coluna contêineres informa quantos contêineres foram executados pela máquina cliente e a última coluna exibe a porcentagem de utilização do equipamento.

Cliente	T. Total (s)	T. Executando (s)	T. Ocioso (s)	Contêineres	Uso (%)
10.135.19.56	10228	7080	3148	17	69
10.135.28.4	7834	4964	2870	17	63
10.135.33.66	8087	5056	3031	19	62
10.135.40.164	8391	8258	133	3	98
10.135.47.222	8123	7990	133	3	98
10.135.47.50	8189	8060	129	3	98
10.135.48.19	7782	7668	114	3	98
10.135.53.163	7558	4670	2888	19	61
10.135.56.230	7847	7735	112	3	98
10.135.60.135	7798	7539	259	4	96
10.135.63.183	7791	7512	279	4	96
10.135.64.79	7718	7464	254	4	96
10.135.65.216	7569	6864	705	6	90
10.135.65.252	7552	4647	2905	19	61
10.135.65.54	7704	7419	285	4	96
10.135.65.76	7690	7409	281	4	96
10.135.66.138	7571	7254	317	4	95
10.135.66.140	7557	7270	287	4	96
10.135.66.149	7559	7263	296	4	96
10.135.66.157	7575	7398	177	4	97
10.135.66.163	7562	7413	149	4	98
10.135.66.164	7559	7376	183	4	97
10.135.66.171	7539	7347	192	4	97
10.135.66.182	7554	7391	163	4	97
10.135.66.184	9775	9626	149	4	98
10.135.66.214	9267	6398	2869	19	69
10.135.66.222	9634	6543	3091	20	67
10.135.66.230	7538	7097	441	5	94
10.135.66.252	9634	7848	1786	10	81
10.135.66.47	7558	4616	2942	20	61
10.135.67.101	7541	5188	2353	14	68
10.135.67.106	7537	4839	2698	15	64
10.135.67.114	7554	5721	1833	13	75
10.135.67.132	7537	4740	2797	16	62

Cliente	T. Total (s)	T. Executando (s)	T. Ocioso (s)	Contêineres	Uso (%)
10.135.67.134	7532	4718	2814	17	62
10.135.67.136	7537	4846	2691	16	64
10.135.67.154	7550	4872	2678	16	64
10.135.67.163	10580	8002	2578	15	75
10.135.67.166	10643	8150	2493	15	76
10.135.67.168	7544	4865	2679	16	64
10.135.67.187	7526	4844	2682	15	64
10.135.67.188	7548	7213	335	3	95
10.135.67.190	7525	4743	2782	15	63
10.135.67.217	7533	4806	2727	18	63
10.135.67.222	7533	4856	2677	15	64
10.135.67.226	7537	4899	2638	16	64
10.135.67.230	7542	5771	1771	9	76
10.135.67.34	9641	8520	1121	8	88
10.135.67.44	7539	5475	2064	15	72
10.135.67.45	7529	7337	192	4	97
10.135.67.5	9635	8740	895	8	90
10.135.67.54	7543	7331	212	5	97
10.135.67.70	7538	4663	2875	19	61
10.135.67.77	7555	4646	2909	18	61
10.135.68.103	7539	5521	2018	9	73
10.135.68.121	9378	7954	1424	6	84
10.135.68.127	9378	7940	1438	8	84
10.135.68.130	9376	7960	1416	7	84
10.135.68.136	7541	4623	2918	14	61
10.135.68.149	7529	4527	3002	15	60
10.135.68.15	7790	6420	1370	10	82
10.135.68.162	7535	4641	2894	17	61
10.135.68.215	7515	4482	3033	15	59
10.135.68.22	7531	5824	1707	11	77
10.135.68.223	7535	4758	2777	15	63
10.135.68.228	7550	4721	2829	14	62
10.135.68.229	7537	4661	2876	16	61
10.135.68.237	7538	4893	2645	13	64
10.135.68.24	7544	4748	2796	15	62
10.135.68.250	7544	4812	2732	13	63
10.135.68.26	7533	4675	2858	13	62
10.135.68.29	7543	4904	2639	12	65
10.135.68.34	7549	5184	2365	13	68
10.135.68.36	7540	5004	2536	13	66
10.135.68.45	7542	4834	2708	16	64
10.135.68.49	7548	4756	2792	14	63
10.135.68.61	7540	7121	419	3	94
10.135.68.66	7531	4751	2780	17	63
10.135.68.93	7560	5774	1786	11	76
10.135.68.99	7550	5465	2085	11	72
10.135.69.115	6979	4356	2623	17	62
10.135.69.118	7019	7017	2	2	99

Cliente	T. Total (s)	T. Executando (s)	T. Ocioso (s)	Contêineres	Uso (%)
10.135.69.119	6998	6997	1	2	99
10.135.69.123	6999	4532	2467	15	64
10.135.69.130	6983	4433	2550	13	63
10.135.69.132	6988	4392	2596	15	62
10.135.69.154	7018	7007	11	2	99
10.135.69.155	7000	6998	2	2	99
10.135.69.16	7528	4780	2748	15	63
10.135.69.164	6960	4344	2616	16	62
10.135.69.176	7002	4432	2570	15	63
10.135.69.19	7538	4703	2835	14	62
10.135.69.20	7536	4483	3053	15	59
10.135.69.23	7539	4745	2794	13	62
10.135.69.35	7530	4959	2571	13	65
10.135.69.42	6990	4846	2144	14	69
10.135.69.55	6995	4616	2379	13	65
10.135.69.63	7003	4705	2298	15	67
10.135.69.74	6969	4770	2199	13	68
10.135.69.88	6993	4505	2488	14	64
TOTAL	777385	593533	183852	1122	76.35

APÊNDICE B – EXECUÇÃO POR PACOTES DE EXECUÇÃO

Este Apêndice contém as informações do experimento realizado com os pacotes de execução. A coluna cliente representa a máquina cliente utilizada; a coluna Tempo Total representa o tempo, em segundos, entre o início da primeira execução e a finalização da última execução; a coluna Tempo executando representa quanto tempo em segundos a máquina estava realizando a execução de contêineres; a coluna Tempo Ocioso representa a diferença entre o tempo total e o tempo executando; a coluna contêineres informa quantos contêineres foram executados pela máquina cliente e a última coluna exibe a porcentagem de utilização do equipamento.

Cliente	T. Total (s)	T. Executando (s)	T. Ocioso (s)	Contêineres	Uso (%)
10.135.19.56	5393	4813	580	17	89
10.135.28.4	5380	4904	476	12	91
10.135.33.66	5433	4888	545	19	89
10.135.40.164	7216	7214	2	2	99
10.135.47.222	7262	7259	3	2	99
10.135.47.50	5355	4766	589	14	89
10.135.48.19	5367	4829	538	18	89
10.135.53.163	5359	4926	433	13	91
10.135.56.230	8124	8071	53	7	99
10.135.60.135	5352	4867	485	13	90
10.135.63.183	7248	7247	1	2	99
10.135.64.79	5371	4920	451	17	91
10.135.65.216	8122	8053	69	8	99
10.135.65.252	5334	4931	403	13	92
10.135.65.54	5411	4916	495	12	90
10.135.65.76	8124	8042	82	8	98
10.135.66.138	7244	7243	1	2	99
10.135.66.140	5414	4955	459	12	91
10.135.66.149	5472	5419	53	8	99
10.135.66.157	5714	5671	43	10	99
10.135.66.163	8232	7735	497	13	93
10.135.66.164	8233	8111	122	9	98
10.135.66.171	5793	5746	47	12	99
10.135.66.182	7224	7222	2	2	99
10.135.66.184	5400	5340	60	8	98
10.135.66.214	8043	7985	58	8	99
10.135.66.222	8044	8008	36	6	99
10.135.66.230	8226	7943	283	10	96
10.135.66.252	8046	8002	44	8	99
10.135.66.47	7219	7217	2	2	99
10.135.67.101	5480	5153	327	11	94
10.135.67.106	5397	4982	415	13	92
10.135.67.114	5445	4985	460	12	91

Cliente	T. Total (s)	T. Executando (s)	T. Ocioso (s)	Contêineres	Uso (%)
10.135.67.132	5392	4910	482	13	91
10.135.67.134	5402	4838	564	15	89
10.135.67.136	5372	4793	579	14	89
10.135.67.154	7222	7218	4	2	99
10.135.67.163	5346	4851	495	13	90
10.135.67.166	5353	4821	532	14	90
10.135.67.168	5333	4854	479	13	91
10.135.67.187	5367	4742	625	16	88
10.135.67.188	5347	4816	531	15	90
10.135.67.190	5384	4859	525	14	90
10.135.67.217	5333	4842	491	16	90
10.135.67.222	5344	4782	562	11	89
10.135.67.226	5309	4809	500	16	90
10.135.67.230	5359	4894	465	13	91
10.135.67.34	5528	5159	369	6	93
10.135.67.44	5342	5131	211	5	96
10.135.67.45	5436	4872	564	13	89
10.135.67.5	5348	5267	81	7	98
10.135.67.54	5461	4973	488	15	91
10.135.67.70	5498	4971	527	14	90
10.135.67.77	5545	5157	388	11	93
10.135.68.103	5337	4855	482	13	90
10.135.68.121	5290	5040	250	10	95
10.135.68.127	5362	4836	526	12	90
10.135.68.130	5345	4865	480	15	91
10.135.68.136	5301	4763	538	14	89
10.135.68.149	6230	5842	388	12	93
10.135.68.15	5294	4851	443	16	91
10.135.68.162	6100	5616	484	12	92
10.135.68.215	6043	5677	366	10	93
10.135.68.22	5367	4856	511	15	90
10.135.68.223	5777	5167	610	12	89
10.135.68.228	5712	5302	410	12	92
10.135.68.229	5694	5175	519	13	90
10.135.68.237	5346	4831	515	12	90
10.135.68.24	5305	4843	462	13	91
10.135.68.250	5356	5017	339	14	93
10.135.68.26	7222	7218	4	2	99
10.135.68.29	5353	4839	514	13	90
10.135.68.34	5946	5866	80	8	98
10.135.68.36	5304	4946	358	14	93
10.135.68.45	5604	5496	108	8	98
10.135.68.49	5917	5843	74	9	98
10.135.68.61	7629	7162	467	13	93
10.135.68.66	7713	7458	255	9	96
10.135.68.93	7133	6611	522	13	92
10.135.68.99	5324	4999	325	11	93
10.135.69.115	5266	4729	537	9	89

Cliente	T. Total (s)	T. Executando (s)	T. Ocioso (s)	Contêineres	Uso (%)
10.135.69.118	5335	4818	517	12	90
10.135.69.119	5319	4714	605	11	88
10.135.69.123	5316	4671	645	11	87
10.135.69.130	5319	4737	582	9	89
10.135.69.132	5303	4737	566	10	89
10.135.69.154	5265	4724	541	10	89
10.135.69.155	5332	4736	596	16	88
10.135.69.16	5278	4932	346	13	93
10.135.69.164	5294	4727	567	11	89
10.135.69.176	5328	4707	621	13	88
10.135.69.19	5274	5004	270	12	94
10.135.69.20	5295	4867	428	15	91
10.135.69.23	5282	4702	580	14	89
10.135.69.35	5367	4774	593	14	88
10.135.69.42	5291	4785	506	12	90
10.135.69.55	5357	4864	493	12	90
10.135.69.63	5291	4897	394	12	92
10.135.69.74	5285	5045	240	11	95
10.135.69.88	5271	4755	516	11	90
TOTAL	587570	549821	37749	1122	93.57

APÊNDICE C – RESULTADOS DAS FÓRMULAS

Este Apêndice contém as informações das fórmulas utilizadas nos experimentos do trabalho. A coluna ID representa a identificação numérica da fórmula utilizada; as colunas E_x representam o tempo, em segundos, da execução x da fórmula; a coluna variação média informa a variação ocorrida entre os tempos das seis execuções.

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
1	0.836	0.703	1.128	1.11	0.707	0.798	20.72
2	0.012	0.016	0.027	0.021	0.012	0.009	51.67
3	0.313	0.142	0.174	0.129	0.158	0.093	55.53
4	0.03	0.053	0.027	0.027	0.016	0.02	35.33
5	0.122	0.167	0.136	0.081	0.132	0.113	19.51
6	0.14	0.099	0.075	0.06	0.091	0.065	44.29
7	0.852	1.103	0.906	0.986	0.78	0.919	13.57
8	0.008	0.007	0.017	0.002	0.004	0.007	52.50
9	0.019	0.012	0.01	0.011	0.009	0.007	48.42
10	0.526	0.552	0.555	0.541	0.8	0.848	25.32
11	0.29	0.403	0.317	0.516	0.563	0.216	49.17
12	0.034	0.047	0.04	0.049	0.035	0.034	20.59
13	0.153	0.05	0.063	0.031	0.057	0.059	66.01
14	0.002	0.002	0.002	0.001	0.001	0.001	30.00
15	1.047	2.421	1.318	0.917	1.153	0.936	38.05
16	0.026	0.101	0.022	0.026	0.038	0.038	79.23
17	2239.225	1734.222	2100.501	2405.147	2492.228	1910.832	12.42
18	0.33	0.381	0.457	0.255	0.217	0.224	28.61
19	5.292	5.944	7.352	11.392	6.169	5.506	37.43
20	4.422	4.448	4.611	5.825	5.818	5.622	19.06
21	0.03	0.096	0.068	0.03	0.045	0.031	80.00
22	127.632	147.139	135.771	251.672	214.611	132.008	38.08
23	231.538	223.69	214.468	282.75	210.107	217.444	9.64
24	—	—	—	—	—	—	—
25	0.168	0.215	0.158	0.214	0.134	0.194	19.40
26	0.12	0.493	0.232	0.164	0.11	0.15	94.83
27	11.579	3.944	10.246	3.706	3.774	2.819	57.70
28	32.044	22.515	24.544	15.95	24.261	27.528	28.35
29	0.009	0.015	0.015	0.007	0.009	0.007	35.56
30	94.04	96.982	112.901	130.144	64.485	65.88	24.59
31	109.58	123.125	101.275	73.088	106.759	76.414	17.22
32	0.015	0.013	0.022	0.017	0.029	0.014	34.67
33	1.825	1.474	1.436	3.028	2.105	1.508	27.84
34	0.649	0.729	0.53	0.752	0.814	0.544	17.63
35	0.056	0.087	0.082	0.068	0.042	0.036	36.79
36	732.034	387.773	466.186	615.494	549.458	520.944	30.61

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
37	698.492	1003.246	809.823	1063.715	930.625	880.824	34.24
38	15.976	22.051	27.972	30.298	34.144	28.872	79.44
39	0.074	0.064	0.064	0.063	0.068	0.075	10.27
40	0.007	0.006	0.007	0.01	0.005	0.004	25.71
41	0.008	0.007	0.01	0.007	0.009	0.013	25.00
42	0.001	0.004	0.001	0.002	0.003	0.001	120.00
43	0.006	0.007	0.006	0.003	0.003	0.006	23.33
44	0.015	0.02	0.013	0.008	0.011	0.009	32.00
45	1.214	1.312	1.448	1.539	1.268	1.262	12.50
46	0.05	0.042	0.028	0.033	0.023	0.018	42.40
47	0.197	0.171	0.229	0.19	0.432	0.309	41.83
48	2.467	2.478	3.416	3.623	5.708	2.561	44.19
49	4.592	4.493	4.471	6.586	5.29	7.455	25.15
50	0.036	0.053	0.038	0.019	0.021	0.026	33.89
51	0.113	0.111	0.098	0.112	0.114	0.105	4.78
52	0.088	0.087	0.082	0.078	0.057	0.094	12.27
53	2.961	16.646	4.699	3.084	3.252	4.119	114.79
54	3.37	2.576	5.789	2.046	3.974	3.073	32.27
55	0.007	0.013	0.009	0.003	0.006	0.011	48.57
56	0.008	0.009	0.008	0.009	0.017	0.008	27.50
57	1.563	1.507	1.173	1.476	1.436	1.351	11.16
58	0.37	0.352	0.503	0.718	0.533	0.514	43.57
59	0.062	0.048	0.039	0.035	0.037	0.058	30.00
60	18.891	16.035	14.553	18.588	16.72	18.814	10.32
61	0.066	0.058	0.047	0.065	0.053	0.081	16.97
62	1.569	1.318	2.615	1.584	1.532	1.489	18.22
63	0.038	0.021	0.022	0.024	0.03	0.041	30.53
64	0.427	0.255	0.307	0.481	0.245	0.22	34.43
65	4.929	4.219	5.567	4.47	4.857	4.421	9.69
66	11.66	15.639	12.758	12.815	16.812	14.195	23.87
67	0.045	0.054	0.098	0.082	0.038	0.032	52.89
68	313.696	334.647	371.225	507.04	445.517	343.097	27.61
69	0.17	0.175	0.147	0.137	0.181	0.234	16.00
70	0.16	0.151	0.075	0.114	0.125	0.133	25.25
71	5.502	5.508	5.613	5.204	4.724	4.869	6.64
72	16.794	14.202	16.516	21.544	16.987	18.548	11.39
73	0.01	0.01	0.014	0.011	0.006	0.004	30.00
74	25.172	27.245	33.271	24.083	23.253	26.592	11.60
75	73.989	85.068	88.881	93.181	48.517	118.15	31.03
76	0.033	0.024	0.016	0.013	0.018	0.014	48.48
77	1.172	1.134	0.95	1.504	1.48	1.572	22.18
78	0.396	0.398	0.529	0.776	0.731	0.81	63.84
79	0.068	0.052	0.079	0.04	0.044	0.049	28.82
80	12.258	10.339	9.537	17.277	15.459	13.328	22.73
81	11.385	9.348	7.267	10.32	9.728	9.348	19.17
82	0.08	0.058	0.121	0.071	0.055	0.085	25.50

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
83	0.011	0.009	0.014	0.011	0.01	0.011	10.91
84	0.082	0.12	0.073	0.05	0.06	0.049	32.68
85	0.018	0.022	0.028	0.013	0.015	0.011	32.22
86	0.057	0.07	0.077	0.079	0.103	0.036	42.81
87	0.047	0.081	0.044	0.071	0.084	0.042	43.83
88	0.003	0.003	0.003	0.002	0.002	0.005	26.67
89	0.015	0.006	0.006	0.007	0.017	0.013	40.00
90	0.04	0.021	0.055	0.025	0.012	0.012	52.50
91	0.02	0.046	0.04	0.037	0.021	0.029	73.00
92	0.013	0.009	0.013	0.005	0.008	0.004	40.00
93	0.018	0.016	0.016	0.006	0.008	0.007	41.11
94	0.002	0.003	0.002	0.001	0.001	0.001	40.00
95	0.005	0.003	0.004	0.006	0.002	0.015	68.00
96	0.012	0.018	0.013	0.016	0.01	0.008	28.33
97	0.011	0.008	0.009	0.01	0.014	0.013	20.00
98	0.092	0.09	0.084	0.09	0.087	0.05	12.83
99	0.012	0.025	0.009	0.009	0.018	0.014	45.00
100	0.127	0.07	0.07	0.07	0.066	0.064	46.46
101	0.067	0.037	0.033	0.051	0.109	0.043	43.58
102	0.003	0.003	0.005	0.01	0.006	0.005	93.33
103	0.015	0.014	0.011	0.004	0.004	0.006	48.00
104	0.016	0.046	0.037	0.013	0.025	0.013	82.50
105	0.043	0.032	0.028	0.029	0.03	0.017	36.74
106	1.041	1.07	1.011	1.145	1.148	1.681	17.48
107	0.053	0.039	0.035	0.022	0.02	0.019	49.06
108	0.303	0.306	0.263	0.521	0.221	0.217	28.32
109	8.147	6.982	8.112	10.889	7.973	13.352	22.88
110	10.067	11.655	8.296	8.62	13.148	12.829	21.16
111	0.052	0.054	0.098	0.029	0.056	0.025	39.23
112	83.471	79.024	84.758	104.066	93.093	77.226	10.11
113	0.225	0.182	0.129	0.119	0.158	0.201	29.87
114	0.069	0.073	0.1	0.101	0.113	0.106	42.90
115	4.676	3.09	2.932	2.557	2.539	3.647	36.85
116	47.887	40.192	34.63	39.539	33.149	41.885	20.90
117	0.016	0.007	0.007	0.004	0.008	0.009	56.25
118	142.016	89.809	61.956	67.873	65.568	87.756	47.48
119	333.036	296.41	178.008	250.529	197.423	334.871	24.72
120	0.025	0.042	0.021	0.013	0.021	0.019	34.40
121	0.842	0.896	0.861	1.175	0.905	0.802	12.09
122	0.662	0.983	1.019	0.708	0.758	0.697	25.83
123	0.046	0.089	0.068	0.053	0.039	0.06	40.43
124	10.868	10.95	12.26	11.055	11.609	16.433	14.66
125	11.041	9.598	9.911	9.652	11.066	17.485	18.90
126	0.091	0.119	0.096	0.103	0.057	0.056	25.05
127	1.202	1.248	1.265	1.213	1.472	1.355	9.03
128	0.025	0.02	0.018	0.018	0.017	0.013	31.20

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
129	0.295	0.27	0.167	0.255	0.277	0.182	21.97
130	0.013	0.016	0.049	0.022	0.014	0.043	121.54
131	0.133	0.091	0.08	0.101	0.132	0.08	27.22
132	0.057	0.082	0.083	0.128	0.048	0.062	47.72
133	0.934	0.858	0.789	0.764	0.764	1.02	13.85
134	0.004	0.006	0.011	0.004	0.003	0.013	95.00
135	0.013	0.011	0.009	0.005	0.01	0.006	36.92
136	0.976	0.835	0.505	0.509	0.698	0.604	35.43
137	0.538	0.815	0.461	0.429	0.579	0.415	23.31
138	0.044	0.046	0.058	0.024	0.036	0.023	29.55
139	0.067	0.131	0.081	0.037	0.036	0.037	50.45
140	1.686	1.987	1.658	2.343	2.457	2.425	29.61
141	0.041	0.044	0.041	0.036	0.029	0.022	19.02
142	1191.074	1742.275	1231.517	1419.576	1075.901	1387.534	19.00
143	0.252	0.242	0.262	0.348	0.271	0.246	11.19
144	6.823	6.485	11.263	6.647	7.213	7.645	18.07
145	12.652	14.888	14.289	16.086	15.989	13.471	18.12
146	0.099	0.057	0.07	0.047	0.063	0.043	43.43
147	640.73	446.038	941.402	602.797	744.704	472.328	25.15
148	125.022	99.578	97.593	91.484	77.941	108.093	24.06
149	—	—	—	—	—	—	—
150	0.124	0.147	0.25	0.147	0.166	0.123	34.68
151	0.17	0.148	0.114	0.074	0.071	0.096	40.82
152	1.988	2.434	3.288	2.038	2.381	2.333	25.49
153	29.372	26.88	23.025	24.405	28.496	37.113	15.27
154	0.01	0.012	0.018	0.009	0.011	0.014	32.00
155	215.578	157.741	237.854	280.231	157.375	394.163	35.40
156	254.492	459.678	361.814	470.473	287.537	494.971	63.03
157	0.022	0.02	0.027	0.013	0.019	0.017	21.82
158	1.06	1.42	1.155	0.985	1.002	0.981	12.58
159	0.611	0.56	0.692	0.549	0.637	0.79	13.06
160	0.083	0.072	0.071	0.051	0.085	0.049	21.93
161	—	—	—	—	—	—	—
162	1321.527	933.632	1160.462	1012.207	1265.712	1334.425	14.03
163	28.597	32.043	35.187	30.615	38.839	27.345	16.47
164	0.084	0.08	0.131	0.066	0.134	0.091	30.00
165	0.001	0.001	0.001	0.001	0.001	0	20.00
166	1.1	1.112	1.412	0.952	1.33	1.995	29.04
167	0.072	0.045	0.025	0.023	0.013	0.017	65.83
168	0.181	0.18	0.265	0.186	0.196	0.183	11.82
169	1.902	1.867	1.944	2.635	1.909	1.961	9.21
170	29.627	11.615	7.771	14.018	14.613	8.355	61.95
171	0.064	0.072	0.05	0.03	0.06	0.028	30.00
172	0.107	0.156	0.116	0.163	0.129	0.099	26.92

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
173	0.184	0.12	0.288	0.061	0.064	0.08	55.98
174	1.541	1.444	1.605	1.941	2.269	1.688	18.64
175	2.661	2.848	2.716	3.82	3.385	4.11	26.86
176	0.007	0.007	0.006	0.007	0.006	0.008	8.57
177	0.028	0.019	0.012	0.012	0.014	0.016	47.86
178	0.786	0.721	0.635	0.874	0.856	1.325	23.23
179	0.514	0.392	0.319	0.558	0.617	0.432	21.25
180	0.029	0.06	0.088	0.055	0.037	0.03	86.21
181	20.246	13.695	28.791	15.458	13.581	18.995	27.46
182	0.071	0.138	0.073	0.04	0.03	0.039	48.73
183	0.018	0.009	0.018	0.007	0.01	0.023	36.67
184	0.024	0.044	0.024	0.014	0.012	0.012	45.00
185	0.001	0.002	0.001	0.001	0.001	0.001	20.00
186	0.011	0.008	0.003	0.012	0.008	0.009	30.91
187	0.19	0.089	0.163	0.067	0.086	0.062	50.84
188	0.111	0.112	0.144	0.082	0.094	0.081	19.82
189	0.298	0.242	0.222	0.134	0.275	0.19	28.66
190	0.017	0.017	0.017	0.029	0.012	0.019	22.35
191	0.056	0.068	0.054	0.051	0.037	0.028	23.57
192	0.155	0.215	0.145	0.154	0.164	0.143	11.87
193	0.173	0.186	0.213	0.132	0.153	0.174	13.29
194	7.494	25.936	9.947	5.34	4.627	8.538	71.95
195	3.154	2.222	2.661	1.508	2.054	2.007	33.72
196	21.459	5.219	7.363	3.948	4.8	5.421	75.07
197	7.325	7.554	14.023	8.283	7.743	7.97	24.43
198	8.922	7.187	6.892	7.93	6.551	8.882	16.07
199	2.121	2.66	2.287	1.986	2.505	2.395	14.13
200	10.945	16.617	12.264	11.986	12.252	12.158	19.28
201	4.191	3.53	3.575	5.29	5.04	4.457	16.66
202	3.603	3.842	4.81	4.908	5.64	2.395	33.28
203	6.851	8.43	9.1	8.044	6.79	7.809	17.63
204	18.017	10.675	9.341	11.085	8.547	10.542	44.29
205	0.302	0.179	0.57	0.162	0.17	0.135	54.97
206	15.368	16.236	13.934	14.365	15.056	12.819	8.02
207	0.67	1.182	0.609	0.445	1.056	0.49	40.72
208	5.503	5.823	7.352	6.178	4.962	5.523	12.38
209	5.795	6.254	5.534	4.664	6.672	6.241	10.95
210	1.916	2.518	2.527	2.504	2.636	3.221	39.94
211	7.222	7.259	5.604	9.999	16.385	12.711	52.85
212	2.637	2.487	4.944	3.354	3.104	2.217	30.80
213	25.866	22.188	16.085	16.224	24.403	35.202	26.21
214	6.208	6.577	4.268	7.451	7.107	8.363	21.28
215	20.338	17.532	17.251	10.84	16.225	25.385	24.14
216	25.803	28.418	23.475	29.577	40.915	48.748	36.25
217	50.573	31.78	31.323	53.954	25.909	37.513	31.30
218	13.074	8.522	7.033	8.628	10.69	11.394	29.22

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
219	49.661	52.152	54.461	31.093	40.735	42.003	17.09
220	25.291	35.392	18.067	15.234	13.187	17.975	37.01
221	11.398	12.683	14.694	11.377	14.598	17.824	24.97
222	26.997	24.645	23.358	29.199	31.684	21.695	13.47
223	23.472	30.932	39.764	46.675	27.715	38.411	56.35
224	0.362	0.29	0.276	0.305	0.379	0.816	37.90
225	25.955	27.321	56.543	52.81	33.432	26.444	51.45
226	33.732	25.926	36.725	61.901	32.587	44.334	30.07
227	1.001	4.829	1.812	1.722	1.032	1.478	117.24
228	14.209	32.474	20.697	22.31	16.26	19.47	56.54
229	20.623	18.644	18.237	23.604	23.473	16.799	13.60
230	11.402	11.202	9.643	9.897	11.016	8.959	11.04
231	51.237	46.88	44.609	47.622	78.885	99.922	35.50
232	44.946	47.808	49.178	45.955	65.636	37.023	16.34
233	31.222	26.782	30.702	21.202	30.215	34.706	12.47
234	10.262	13.3	14.587	10.046	13.955	13.956	29.17
235	0.009	0.015	0.014	0.007	0.006	0.011	40.00
236	0.002	0.001	0.001	0.001	0.002	0.005	60.00
237	0.004	0.009	0.009	0.006	0.009	0.01	115.00
238	13.673	8.161	6.691	8.059	12.613	13.756	28.16
239	1.898	2.013	2.014	1.667	2.003	3.238	20.09
240	6.284	5.196	6.434	7.667	8.385	6.384	15.35
241	30.852	9.855	9.063	11.458	12.377	9.042	66.42
242	13.125	26.268	10.744	11.355	10.505	9.937	35.20
243	2.812	4.769	2.982	2.247	4.386	4.638	43.33
244	14.597	23.921	50.75	10.523	14.736	14.62	68.11
245	4.437	4.318	4.978	5.23	5.548	3.458	15.97
246	4.883	4.145	4.202	3.206	4.765	10.95	38.01
247	5.743	5.662	8.161	7.818	14.568	11.759	67.61
248	18.725	9.632	9.483	11.122	13.86	11.618	40.49
249	0.209	0.151	0.197	0.219	0.263	0.14	19.43
250	14.742	11.949	9.362	16.447	12.81	10.419	21.89
251	0.492	0.483	0.846	0.683	0.484	0.466	23.90
252	9.005	4.561	10.985	7.406	12.744	6.604	31.46
253	9.121	8.124	8.442	9.265	8.345	5.684	13.23
254	3.008	3.26	4.534	4.321	5.198	9.331	77.15
255	12.331	14.464	10.842	9.86	11.878	9.13	15.81
256	3.529	3.749	4.147	5.204	6.968	7.274	54.96
257	28.819	19.457	15.593	41.945	43.991	17.106	43.44
258	5.956	4.967	6.672	5.454	6.417	6.075	9.36
259	11.483	13.111	13.34	12.578	14.566	13.948	17.64
260	40.331	23.45	23.522	32.109	15.968	20.508	42.70
261	20.395	20.923	23.746	16.141	22.961	25.105	15.11
262	7.322	11.62	7.545	5.59	6.391	7.215	19.92
263	34.946	30.811	30.803	35.807	25.206	30.62	13.28
264	11.774	12.015	11.011	14.911	11.035	11.123	9.40

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
265	16.9	33.8	11.337	11.062	10.996	6.546	52.73
266	27.167	26.081	44.24	17.22	17.652	31.222	30.68
267	28.116	33.526	47.769	27.188	45.141	49.31	45.68
268	0.352	0.336	0.36	0.555	0.335	0.699	33.58
269	26.34	29.318	26.546	38.374	23.806	42.674	25.88
270	25.24	19.014	17.284	24.332	51.286	51.994	53.80
271	1.024	1.02	2.143	3.32	1.585	1.046	78.16
272	15.774	11.234	17.327	30.297	24.667	36.326	63.47
273	13.049	34.268	19.353	22.626	16.594	15.047	65.36
274	7.963	13.149	7.775	11.614	11.037	21.613	64.67
275	45.325	43.417	36.258	25.172	35.221	43.076	19.19
276	36.035	43.382	40.861	37.172	29.082	48.522	18.18
277	22.755	39.724	37.676	30.44	26.015	49.497	61.15
278	10.325	11.904	10.088	11.363	13.877	13.421	18.41
279	0.012	0.007	0.014	0.009	0.008	0.009	28.33
280	0.002	0.002	0.002	0.008	0.001	0.001	80.00
281	0.005	0.003	0.011	0.002	0.007	0.007	60.00
282	2.257	5.378	3.199	2.531	2.07	2.331	40.74
283	0.514	0.423	0.418	0.825	0.446	1.581	63.54
284	1.325	1.648	1.403	1.472	1.829	1.469	18.05
285	0.507	0.439	0.416	0.475	0.749	1.54	57.83
286	1.167	2.136	1.158	3.216	2.077	6.973	166.98
287	0.796	1.123	0.925	1.415	1.182	1.144	45.45
288	3.526	2.788	2.044	3.073	2.384	2.417	27.93
289	0.109	0.073	0.061	0.05	0.054	0.084	40.92
290	0.37	0.171	0.124	0.329	0.294	0.266	36.00
291	2.297	3.971	1.971	3.514	4.597	2.581	50.51
292	1.982	1.869	1.745	1.456	2.236	1.759	13.65
293	0.973	1.667	2.967	2.075	0.713	1.799	100.23
294	0.811	0.925	1.185	1.372	0.844	2.078	57.93
295	0.003	0.007	0.004	0.002	0.003	0.005	53.33
296	0.003	0.001	0.001	0	0	0	86.67
297	0.006	0.001	0.001	0.001	0.001	0.001	83.33
298	0.021	0.02	0.018	0.008	0.009	0.011	37.14
299	0.038	0.012	0.024	0.022	0.023	0.024	44.74
300	0.001	0.001	0.002	0.001	0.001	0.002	40.00
301	0.004	0.004	0.005	0.006	0.024	0.003	120.00
302	2.765	1.824	1.742	3.11	2.319	1.843	26.60
303	1.811	0.523	2.49	0.887	0.612	0.429	60.43
304	1.652	2.545	2.328	4.148	2.353	1.654	57.72
305	0.511	0.622	0.622	0.707	1.456	1.163	78.86
306	1.49	1.223	1.176	2.367	2.138	0.8	37.53
307	7.268	1.569	0.681	0.843	1.134	3.49	78.76
308	1.397	2.17	1.676	6.427	5.59	5.356	203.78

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
309	0.072	0.085	0.052	0.065	0.096	0.131	34.17
310	0.173	0.379	0.186	0.45	0.634	0.571	156.65
311	1.742	7.455	6.821	3.465	7.859	3.159	230.18
312	2.46	1.518	1.518	3.402	1.336	3.167	37.86
313	0.696	0.632	0.699	0.823	1.435	2.267	71.95
314	0.861	0.86	1.813	0.608	0.919	1.575	45.95
315	0.003	0.006	0.003	0.003	0.006	0.005	53.33
316	0.001	0	0.001	0.004	0.001	0	100.00
317	0.002	0.001	0.003	0.002	0.002	0.001	30.00
318	0.087	0.097	0.097	0.203	0.131	0.295	89.20
319	0.344	0.156	0.115	0.157	0.168	0.128	57.91
320	0.646	0.261	1.115	0.511	0.545	0.343	43.13
321	0.042	0.01	0.015	0.034	0.021	0.022	51.43
322	0.041	0.058	0.035	0.071	0.042	0.099	54.63
323	0.185	0.178	0.155	0.126	0.189	0.237	16.43
324	0.306	0.34	0.366	0.177	0.142	0.234	30.00
325	—	—	—	—	—	—	—
326	2.142	2.229	2.96	4.381	8.661	3.444	102.38
327	—	—	—	—	—	—	—
328	0.042	0.052	0.034	0.088	0.066	0.078	59.05
329	—	—	—	—	—	—	—
330	0.126	0.201	0.154	0.214	0.166	0.295	63.49
331	—	—	—	—	—	—	—
332	—	—	—	—	—	—	—
333	—	—	—	—	—	—	—
334	0.362	0.385	1.377	1.198	0.721	1.075	162.76
335	—	—	—	—	—	—	—
336	—	—	—	—	—	—	—
337	—	—	—	—	—	—	—
338	—	—	—	—	—	—	—
339	—	—	—	—	—	—	—
340	20.792	20.398	20.208	22.15	24.47	20.974	5.96
341	—	—	—	—	—	—	—
342	13.708	18.539	13.648	16.022	19.143	14.534	19.65
343	—	—	—	—	—	—	—
344	17.564	21.568	18.759	18.167	23.681	18.721	14.89
345	—	—	—	—	—	—	—
346	—	—	—	—	—	—	—
347	—	—	—	—	—	—	—
348	—	—	—	—	—	—	—
349	—	—	—	—	—	—	—
350	12.145	11.968	11.189	13.465	11.729	11.686	5.48
351	—	—	—	—	—	—	—
352	—	—	—	—	—	—	—
353	12.177	17.938	13.03	18.029	20.679	14.398	38.09
354	—	—	—	—	—	—	—
355	—	—	—	—	—	—	—

ID	Contêineres Fechados			Pacotes de execução			Variação Média (%)
	E1 (s)	E2 (s)	E3 (s)	E4 (s)	E5 (s)	E6 (s)	
356	—	—	—	—	—	—	—
357	—	—	—	—	—	—	—
358	41.196	56.353	42.054	40.686	62.887	42.838	19.35
359	—	—	—	—	—	—	—
360	—	—	—	—	—	—	—
361	—	—	—	—	—	—	—
362	—	—	—	—	—	—	—
363	0.012	0.016	0.012	0.006	0.01	0.023	38.33
364	1027.457	963.682	964.953	1251.079	1001.734	1240.835	11.47
365	58.756	51.308	79.418	66.531	62.211	87.293	23.10
366	—	—	—	—	—	—	—
367	—	—	—	—	—	—	—
368	—	—	—	—	—	—	—
369	—	—	—	—	—	—	—
370	—	—	—	—	—	—	—
371	—	—	—	—	—	—	—
372	—	—	—	—	—	—	—
373	—	—	—	—	—	—	—
374	67.815	75.534	69.548	71.188	84.811	71.895	10.00

APÊNDICE D – TEMPO DE ESPERA DAS FÓRMULAS

Este Apêndice contém as informações referente o tempo de espera dos contêineres das fórmulas. A coluna ID representa a identificação numérica da fórmula utilizada; as colunas T_x representam o tempo em segundos de espera na fila x da fórmula; a coluna média informa a média ocorrida entre os tempos das execuções.

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
1	7882	7887	7891	7886	5248	5249	5251	5249
2	7869	7873	7877	7873	5245	5246	5248	5246
3	7854	7859	7863	7858	5239	5241	5243	5241
4	7839	7844	7849	7844	5235	5236	5237	5236
5	7823	7828	7833	7828	5228	5230	5232	5230
6	7810	7815	7819	7814	5223	5226	5227	5225
7	7798	7801	7806	7801	5219	5220	5222	5220
8	7784	7788	7792	7788	5215	5217	5218	5216
9	7772	7777	7781	7776	5209	5212	5214	5211
10	7759	7763	7768	7763	5204	5205	5207	5205
11	7746	7750	7755	7750	5201	5202	5204	5202
12	7735	7738	7742	7738	5197	5199	5200	5198
13	7721	7727	7731	7726	5194	5195	5196	5195
14	7705	7711	7715	7710	5192	5192	5193	5192
15	7690	7693	7701	7694	5189	5190	5191	5190
16	7675	7679	7684	7679	5186	5187	5188	5187
17	7659	7664	7670	7664	5184	5185	5185	5184
18	7643	7650	7655	7649	5181	5182	5183	5182
19	7631	7634	7639	7634	5179	5180	5180	5179
20	7620	7624	7627	7623	5178	5178	5178	5178
21	7608	7612	7616	7612	5175	5176	5177	5176
22	7595	7599	7603	7599	5172	5173	5174	5173
23	7578	7583	7589	7583	5170	5171	5172	5171
24	7548	7552	7573	7557	5167	5168	5169	5168
25	7539	7542	7545	7542	5165	5166	5166	5165
26	7521	7525	7532	7526	5163	5164	5165	5164
27	7503	7510	7514	7509	5160	5161	5162	5161
28	7491	7496	7499	7495	5158	5159	5160	5159
29	7479	7482	7487	7482	5157	5156	5157	5156
30	7463	7470	7474	7469	5154	5155	5156	5155
31	7451	7456	7460	7455	5151	5152	5153	5152
32	7441	7444	7448	7444	5149	5150	5151	5150
33	7432	7435	7438	7435	5148	5149	5148	5148
34	7420	7423	7429	7424	5145	5146	5147	5146
35	7408	7412	7417	7412	5143	5144	5145	5144

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
36	7395	7400	7403	7399	5141	5142	5142	5141
37	7384	7388	7391	7387	5139	5140	5141	5140
38	7374	7378	7381	7377	5135	5136	5138	5136
39	7366	7369	7372	7369	5134	5134	5135	5134
40	7356	7359	7361	7358	5131	5132	5133	5132
41	7347	7350	7352	7349	5129	5130	5131	5130
42	7334	7338	7342	7338	5128	5129	5129	5128
43	7321	7324	7331	7325	5125	5127	5127	5126
44	7311	7314	7318	7314	5123	5123	5124	5123
45	7301	7304	7308	7304	5121	5121	5122	5121
46	7290	7294	7297	7293	5119	5120	5120	5119
47	7278	7282	7286	7282	5116	5117	5118	5117
48	7268	7271	7275	7271	5115	5116	5116	5115
49	7260	7262	7265	7262	5113	5114	5115	5114
50	7249	7252	7257	7252	5111	5112	5113	5112
51	7239	7242	7245	7242	5109	5109	5110	5109
52	7229	7232	7236	7232	5107	5108	5109	5108
53	7210	7213	7222	7215	5105	5106	5107	5106
54	7198	7201	7205	7201	5085	5088	5090	5087
55	7187	7190	7194	7190	5078	5080	5083	5080
56	7177	7180	7183	7180	5070	5073	5075	5072
57	7167	7170	7174	7170	5062	5064	5068	5064
58	7158	7160	7162	7160	5056	5058	5061	5058
59	7147	7151	7153	7150	5047	5050	5052	5049
60	7137	7140	7143	7140	5039	5042	5045	5042
61	7129	7132	7135	7132	5032	5034	5037	5034
62	7117	7120	7123	7120	5025	5028	5030	5027
63	7105	7109	7112	7108	5020	5022	5023	5021
64	7094	7098	7101	7097	5012	5015	5017	5014
65	7083	7088	7091	7087	5005	5008	5010	5007
66	7073	7077	7080	7076	4998	5000	5003	5000
67	7061	7064	7069	7064	4990	4993	4995	4992
68	7047	7051	7055	7051	4983	4986	4988	4985
69	7037	7040	7042	7039	4977	4980	4981	4979
70	7027	7031	7034	7030	4970	4973	4975	4972
71	7017	7021	7024	7020	4963	4966	4968	4965
72	7008	7011	7014	7011	4956	4959	4961	4958
73	6997	7001	7005	7001	4950	4952	4954	4952
74	6981	6987	6992	6986	4939	4945	4947	4943
75	6971	6974	6978	6974	4932	4935	4936	4934
76	6961	6965	6968	6964	4925	4927	4930	4927
77	6950	6953	6959	6954	4918	4920	4922	4920
78	6940	6944	6946	6943	4910	4912	4915	4912
79	6928	6933	6936	6932	4903	4905	4907	4905
80	6919	6921	6925	6921	4897	4899	4901	4899
81	6910	6913	6916	6913	4891	4893	4894	4892
82	6900	6903	6907	6903	4884	4886	4889	4886
83	6891	6894	6897	6894	4876	4879	4882	4879

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
84	6880	6883	6888	6883	4868	4871	4873	4870
85	6866	6870	6873	6869	4861	4863	4866	4863
86	6854	6857	6861	6857	4853	4857	4859	4856
87	6843	6847	6851	6847	4847	4849	4851	4849
88	6834	6837	6840	6837	4841	4843	4845	4843
89	6823	6828	6831	6827	4834	4836	4838	4836
90	6811	6816	6819	6815	4828	4830	4832	4830
91	6801	6804	6807	6804	4822	4824	4826	4824
92	6789	6793	6797	6793	4816	4818	4819	4817
93	6778	6782	6785	6781	4809	4811	4813	4811
94	6768	6771	6774	6771	4802	4804	4807	4804
95	6758	6761	6764	6761	4796	4798	4800	4798
96	6748	6751	6755	6751	4789	4791	4793	4791
97	6739	6742	6745	6742	4784	4785	4787	4785
98	6730	6732	6736	6732	4777	4780	4782	4779
99	6721	6723	6728	6724	4770	4773	4775	4772
100	6710	6713	6718	6713	4764	4766	4768	4766
101	6700	6703	6707	6703	4756	4758	4762	4758
102	6687	6690	6695	6690	4749	4751	4753	4751
103	6674	6680	6683	6679	4743	4745	4747	4745
104	6665	6667	6671	6667	4737	4739	4741	4739
105	6654	6658	6661	6657	4731	4733	4735	4733
106	6645	6648	6651	6648	4725	4727	4729	4727
107	6632	6638	6642	6637	4718	4720	4723	4720
108	6620	6623	6629	6624	4712	4714	4716	4714
109	6612	6614	6617	6614	4705	4707	4709	4707
110	6602	6606	6609	6605	4699	4700	4703	4700
111	6591	6594	6598	6594	4693	4695	4697	4695
112	6582	6586	6588	6585	4686	4688	4690	4688
113	6569	6573	6578	6573	4680	4682	4684	4682
114	6560	6563	6566	6563	4674	4676	4678	4676
115	6550	6553	6557	6553	4668	4670	4672	4670
116	6542	6546	6548	6545	4662	4664	4667	4664
117	6533	6538	6540	6537	4656	4658	4660	4658
118	6524	6528	6531	6527	4651	4653	4654	4652
119	6513	6518	6521	6517	4645	4647	4649	4647
120	6499	6503	6510	6504	4638	4640	4642	4640
121	6489	6491	6494	6491	4632	4633	4635	4633
122	6477	6480	6483	6480	4626	4628	4630	4628
123	6467	6470	6474	6470	4620	4622	4624	4622
124	6457	6461	6463	6460	4614	4616	4618	4616
125	6448	6452	6455	6451	4608	4610	4612	4610
126	6437	6440	6442	6439	4601	4603	4606	4603
127	6427	6430	6433	6430	4596	4598	4600	4598
128	6418	6421	6424	6421	4589	4592	4594	4591
129	6408	6411	6415	6411	4583	4585	4587	4585
130	6398	6401	6404	6401	4577	4579	4581	4579
131	6388	6392	6395	6391	4572	4573	4576	4573

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
132	6379	6381	6384	6381	4566	4568	4570	4568
133	6371	6373	6377	6373	4562	4564	4565	4563
134	6362	6364	6367	6364	4556	4558	4560	4558
135	6353	6357	6360	6356	4551	4553	4554	4552
136	6343	6347	6351	6347	4546	4548	4550	4548
137	6332	6336	6339	6335	4541	4542	4544	4542
138	6320	6323	6329	6324	4535	4537	4539	4537
139	6309	6312	6317	6312	4530	4532	4533	4531
140	6299	6302	6307	6302	4523	4526	4528	4525
141	6287	6290	6293	6290	4519	4521	4522	4520
142	6277	6280	6284	6280	4514	4516	4517	4515
143	6269	6272	6274	6271	4508	4510	4512	4510
144	6257	6260	6262	6259	4503	4504	4506	4504
145	6249	6251	6254	6251	4498	4500	4502	4500
146	6239	6243	6247	6243	4492	4494	4496	4494
147	6223	6229	6232	6228	4487	4489	4490	4488
148	6209	6213	6219	6213	4482	4484	4486	4484
149	6188	6192	6203	6194	4477	4478	4480	4478
150	6179	6182	6185	6182	4470	4472	4474	4472
151	6171	6174	6176	6173	4466	4467	4469	4467
152	6162	6165	6167	6164	4461	4463	4464	4462
153	6154	6156	6159	6156	4456	4458	4459	4457
154	6143	6148	6151	6147	4451	4452	4454	4452
155	6137	6139	6141	6139	4445	4447	4449	4447
156	6129	6131	6134	6131	4440	4442	4443	4441
157	6120	6122	6126	6122	4436	4437	4438	4437
158	6111	6114	6117	6114	4431	4433	4434	4432
159	6102	6104	6108	6104	4425	4427	4429	4427
160	6091	6095	6098	6094	4420	4422	4424	4422
161	6079	6081	6087	6082	4415	4417	4418	4416
162	6072	6074	6077	6074	4410	4412	4413	4411
163	6061	6063	6068	6064	4406	4408	4409	4407
164	6051	6053	6057	6053	4400	4402	4404	4402
165	6042	6044	6049	6045	4395	4397	4398	4396
166	6032	6035	6038	6035	4391	4392	4394	4392
167	6019	6023	6029	6023	4386	4387	4389	4387
168	6010	6013	6017	6013	4380	4382	4384	4382
169	6002	6005	6007	6004	4376	4378	4379	4377
170	5991	5997	6000	5996	4370	4373	4375	4372
171	5979	5982	5988	5983	4364	4366	4368	4366
172	5963	5968	5973	5968	4360	4361	4363	4361
173	5951	5955	5958	5954	4355	4356	4358	4356
174	5944	5946	5948	5946	4350	4351	4353	4351
175	5935	5938	5941	5938	4344	4346	4348	4346
176	5929	5930	5932	5930	4339	4341	4342	4340
177	5919	5923	5926	5922	4334	4336	4337	4335
178	5910	5913	5916	5913	4329	4331	4333	4331
179	5901	5905	5907	5904	4324	4325	4327	4325

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
180	5892	5896	5898	5895	4319	4321	4323	4321
181	5884	5887	5889	5886	4315	4316	4318	4316
182	5876	5879	5881	5878	4310	4311	4313	4311
183	5867	5870	5872	5869	4305	4306	4308	4306
184	5858	5861	5865	5861	4300	4301	4303	4301
185	5848	5851	5855	5851	4294	4296	4298	4296
186	5839	5842	5846	5842	4289	4291	4291	4290
187	5830	5833	5836	5833	4283	4285	4287	4285
188	5821	5823	5827	5823	4279	4280	4282	4280
189	5812	5816	5819	5815	4274	4276	4277	4275
190	5806	5808	5810	5808	4270	4272	4273	4271
191	5797	5799	5803	5799	4266	4267	4268	4267
192	5789	5792	5794	5791	4261	4263	4264	4262
193	5781	5783	5787	5783	4256	4258	4259	4257
194	5764	5768	5776	5769	4253	4254	4255	4254
195	5753	5757	5760	5756	4248	4249	4251	4249
196	5738	5743	5749	5743	4244	4245	4246	4245
197	5728	5731	5734	5731	4240	4242	4242	4241
198	5717	5721	5723	5720	4235	4237	4238	4236
199	5708	5710	5713	5710	4230	4232	4234	4232
200	5476	5480	5485	5480	4225	4226	4228	4226
201	5465	5469	5472	5468	4221	4221	4223	4221
202	5454	5457	5461	5457	4216	4218	4220	4218
203	5440	5444	5450	5444	4211	4213	4215	4213
204	5422	5431	5436	5429	4206	4208	4209	4207
205	5402	5408	5412	5407	4202	4203	4205	4203
206	5387	5392	5398	5392	4196	4198	4200	4198
207	5371	5375	5381	5375	4193	4195	4196	4194
208	5353	5359	5362	5358	4188	4190	4192	4190
209	5342	5346	5350	5346	4184	4185	4187	4185
210	5329	5333	5337	5333	4179	4180	4182	4180
211	5315	5320	5325	5320	4173	4175	4177	4175
212	5300	5303	5308	5303	4170	4171	4172	4171
213	5288	5292	5296	5292	4164	4166	4169	4166
214	5275	5280	5284	5279	4159	4161	4162	4160
215	5260	5266	5271	5265	4155	4156	4157	4156
216	5245	5250	5255	5250	4150	4151	4153	4151
217	5224	5233	5240	5232	4144	4146	4148	4146
218	5210	5217	5221	5216	4136	4138	4142	4138
219	5189	5196	5202	5195	4131	4132	4134	4132
220	5169	5174	5185	5176	4127	4128	4129	4128
221	5158	5161	5165	5161	4123	4124	4126	4124
222	5144	5149	5153	5148	4119	4120	4122	4120
223	5130	5135	5140	5135	4114	4116	4117	4115
224	5119	5122	5127	5122	4110	4111	4113	4111
225	5107	5110	5113	5110	4106	4107	4109	4107
226	5093	5099	5102	5098	4102	4103	4105	4103
227	5080	5083	5090	5084	4098	4099	4100	4099
228	5065	5069	5075	5069	4093	4095	4096	4094

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
229	5052	5057	5061	5056	4089	4090	4093	4090
230	5040	5043	5049	5044	4084	4086	4087	4085
231	5025	5030	5034	5029	4078	4080	4082	4080
232	5006	5011	5020	5012	4075	4076	4077	4076
233	4994	4998	5001	4997	4070	4072	4073	4071
234	4982	4986	4990	4986	4066	4066	4068	4066
235	4971	4974	4979	4974	4061	4063	4064	4062
236	4962	4965	4968	4965	4057	4058	4060	4058
237	4948	4952	4957	4952	4053	4054	4056	4054
238	4935	4940	4943	4939	4049	4050	4052	4050
239	4926	4929	4932	4929	4045	4046	4047	4046
240	4914	4919	4922	4918	4039	4040	4043	4040
241	4897	4904	4911	4904	4035	4036	4038	4036
242	4878	4882	4892	4884	4031	4032	4033	4032
243	4867	4870	4874	4870	4027	4029	4030	4028
244	4843	4848	4853	4848	4023	4024	4026	4024
245	4832	4836	4839	4835	4019	4021	4022	4020
246	4817	4822	4828	4822	4016	4017	4017	4016
247	4808	4811	4814	4811	4011	4012	4014	4012
248	4796	4800	4803	4799	4006	4007	4009	4007
249	4787	4790	4793	4790	4003	4004	4005	4004
250	4774	4780	4783	4779	3999	4000	4002	4000
251	4760	4763	4769	4764	3995	3996	3998	3996
252	4750	4754	4756	4753	3992	3992	3994	3992
253	4738	4742	4747	4742	3988	3989	3991	3989
254	4729	4732	4735	4732	3983	3984	3986	3984
255	4715	4721	4725	4720	3981	3981	3983	3981
256	4702	4705	4710	4705	3976	3977	3979	3977
257	4690	4695	4699	4694	3971	3973	3975	3973
258	4678	4682	4686	4682	3969	3969	3970	3969
259	4668	4671	4674	4671	3965	3967	3968	3966
260	4653	4661	4665	4659	3962	3964	3965	3963
261	4640	4643	4649	4644	3957	3959	3961	3959
262	4628	4631	4637	4632	3954	3955	3957	3955
263	4616	4620	4624	4620	3949	3951	3952	3950
264	4603	4609	4612	4608	3947	3948	3949	3948
265	4586	4590	4599	4591	3942	3944	3945	3943
266	4570	4575	4579	4574	3938	3939	3941	3939
267	4553	4559	4564	4558	3935	3936	3937	3936
268	4544	4547	4550	4547	3929	3932	3933	3931
269	4528	4532	4539	4533	3925	3925	3927	3925
270	4513	4520	4522	4518	3921	3922	3923	3922
271	4504	4507	4510	4507	3916	3918	3920	3918
272	4494	4497	4500	4497	3912	3912	3914	3912
273	4480	4483	4491	4484	3908	3909	3910	3909
274	4468	4471	4477	4472	3903	3905	3906	3904
275	4451	4457	4462	4456	3899	3900	3902	3900
276	4436	4440	4445	4440	3892	3893	3897	3894

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
277	4418	4423	4430	4423	3887	3888	3890	3888
278	4402	4409	4414	4408	3884	3885	3887	3885
279	4393	4396	4398	4395	3877	3879	3881	3879
280	4369	4372	4390	4377	3874	3876	3876	3875
281	4355	4359	4363	4359	3871	3872	3873	3872
282	4340	4343	4351	4344	3868	3869	3869	3868
283	4332	4335	4337	4334	3865	3865	3866	3865
284	4322	4325	4329	4325	3861	3862	3863	3862
285	4314	4318	4320	4317	3856	3858	3860	3858
286	4304	4307	4311	4307	3850	3851	3854	3851
287	4295	4298	4301	4298	3847	3849	3849	3848
288	4282	4289	4292	4287	3840	3843	3845	3842
289	4272	4277	4279	4276	3834	3835	3836	3835
290	4262	4267	4270	4266	3830	3831	3833	3831
291	4250	4253	4259	4254	3824	3826	3828	3826
292	4239	4243	4247	4243	3820	3821	3822	3821
293	4223	4228	4233	4228	3817	3817	3819	3817
294	4214	4217	4220	4217	3813	3815	3816	3814
295	4203	4207	4211	4207	3808	3809	3811	3809
296	4192	4197	4200	4196	3804	3805	3807	3805
297	4184	4187	4189	4186	3799	3801	3803	3801
298	4173	4177	4181	4177	3796	3797	3798	3797
299	4164	4167	4170	4167	3792	3793	3794	3793
300	4153	4157	4160	4156	3787	3789	3790	3788
301	4144	4148	4150	4147	3783	3784	3786	3784
302	4131	4138	4141	4136	3778	3780	3782	3780
303	4115	4119	4123	4119	3774	3775	3776	3775
304	4099	4103	4110	4104	3769	3770	3771	3770
305	4090	4093	4096	4093	3765	3765	3767	3765
306	4079	4084	4087	4083	3761	3762	3764	3762
307	4066	4072	4077	4071	3757	3758	3760	3758
308	4056	4059	4063	4059	3752	3753	3755	3753
309	4043	4049	4054	4048	3748	3748	3749	3748
310	4031	4033	4038	4034	3742	3743	3747	3744
311	4015	4018	4024	4019	3737	3739	3741	3739
312	4005	4009	4012	4008	3733	3734	3736	3734
313	3994	3999	4002	3998	3726	3728	3731	3728
314	3978	3985	3989	3984	3721	3722	3724	3722
315	3963	3967	3971	3967	3716	3718	3719	3717
316	3948	3951	3954	3951	3711	3712	3714	3712
317	3938	3941	3944	3941	3704	3704	3707	3705
318	3929	3932	3935	3932	3699	3701	3702	3700
319	3916	3921	3923	3920	3691	3695	3696	3694
320	3898	3903	3907	3902	3684	3687	3688	3686
321	3888	3892	3895	3891	3679	3681	3683	3681
322	3877	3882	3885	3881	3671	3672	3675	3672
323	3865	3869	3873	3869	3664	3667	3669	3666
324	3848	3853	3860	3853	3659	3661	3662	3660
325	3836	3841	3844	3840	3653	3654	3655	3654

ID	Contêineres Fechados				Pacotes de Execução			
	T1 (s)	T2 (s)	T3 (s)	Média (s)	T1 (s)	T2 (s)	T3 (s)	Média (s)
326	3818	3822	3829	3823	3645	3647	3650	3647
327	3810	3813	3815	3812	3639	3642	3644	3641
328	3801	3804	3808	3804	3624	3630	3636	3630
329	3779	3795	3797	3790	1503	3611	3618	2910
330	3761	3770	3774	3768	1371	1410	1491	1424
331	3749	3753	3757	3753	245	1080	1322	882
332	3730	3739	3745	3738	213	231	233	225
333	3715	3719	3725	3719	194	198	206	199
334	3702	3706	3709	3705	185	187	188	186
335	3693	3696	3700	3696	169	179	184	177
336	3673	3683	3689	3681	160	164	166	163
337	3651	3658	3663	3657	150	153	159	154
338	1210	3624	3642	2825	137	140	143	140
339	453	1142	1151	915	128	133	134	131
340	442	447	450	446	110	115	123	116
341	423	428	433	428	86	87	97	90
342	417	419	421	419	85	85	86	853
343	409	412	414	411	81	83	84	82
344	400	403	407	403	79	80	81	80
345	391	394	398	394	77	78	78	77
346	380	383	388	383	75	76	77	76
347	362	371	376	369	69	71	74	713
348	352	357	360	356	68	69	69	68
349	341	344	349	344	66	67	68	67
350	330	333	338	333	64	65	66	65
351	308	313	324	315	60	62	63	61
352	300	303	306	303	59	59	60	593
353	292	295	298	295	56	57	58	57
354	272	280	286	279	52	53	54	53
355	263	266	269	266	50	51	51	50
356	254	258	260	257	49	49	50	493
357	247	249	251	249	47	48	48	47
358	234	240	243	239	45	46	47	46
359	226	228	231	228	43	44	44	43
360	217	219	222	219	42	42	43	423
361	198	210	213	207	40	41	41	40
362	181	187	194	187	35	38	39	373
363	172	175	178	175	33	34	34	33
364	160	164	169	164	31	32	32	31
365	149	153	156	152	29	29	30	293
366	131	137	142	136	26	27	28	27
367	102	112	122	112	22	24	25	23
368	89	92	95	92	21	21	22	213
369	72	78	82	773	16	17	19	173
370	59	63	68	633	14	15	15	14
371	46	50	55	503	13	14	14	13
372	32	37	41	36	11	12	12	11
373	19	22	27	22	9	10	10	9
374	8	11	14	11	8	8	9	833
Média				5295				3918