Universidade Federal Fluminense

RAQUEL GERHARDT GOMES BOECHAT

Problema de Distribuição de Capacidades, Réplicas e Requisições

VOLTA REDONDA

2017

RAQUEL GERHARDT GOMES BOECHAT

Problema de Distribuição de Capacidades, Réplicas e Requisições

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Otimização Combinatória.

Orientador:

Tiago Araújo Neves

Coorientador:

Luis Alberto Duncan Rangel

Universidade Federal Fluminense

VOLTA REDONDA

2017

Problema de Distribuição de Capacidades, Réplicas e Requisições

Raquel Gerhardt Gomes Boechat

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Aprovada por:

Prof. Tiago Araujo Neves, D. Sc. / VCE-UFF

Prof. Luis Alberto Duncan Rangel, D. Sc. / VEP-UFF

Prof. Gustavo Benitez Alvarez, D. Sc. / VCE-UFF

Prof. Lino Guimarães Marujo, D.Sc. / DEI-UFRJ

Prof. Celio Vinicius N. de Albuquerque, Ph. D. / IC-UFF

Prof. Weslley Luiz da Silva Assis, D. Sc/ UFF

Volta Redonda, 24 de Outubro de 2017.

Para meu esposo, meus pais e minha irmã.

Agradecimentos

A Deus, pela dávida da vida, pela saúde, por me capacitar a alcançar cada pequeno degrau para me fazer chegar até aqui, pelas bençãos de cada dia e pelo consolo em cada momento, pela ajuda que mais ninguém pode dar.

À minha família, por me incentivar na busca constante de meus objetivos, me apoiando desde o início de minha carreira acadêmica a nunca desistir dos meus sonhos e manter a garra e determinação para buscar o alcance de minhas metas.

Ao meu esposo Felipe Boechat Tavares dos Reis por toda paciência nos momentos de angústia, pela confiança e companheirismo que me traz desde o início de nosso relacionamento, pelo incentivo diário e por sempre acreditar no meu potencial.

Aos meus pais, Renato Eduardo Alves Gomes e Clarice Stutz Gerhardt Gomes, por todo amor, toda a vida de dedicação e compromisso que sempre tiveram comigo, vocês foram essenciais para construção da profissional que me tornei, obrigada por cada palavra que disseram a mim, por cada conselho que me deram, por nunca me desampararem, por me incentivarem e me acompanharem nas minhas decisões e escolhas. Obrigada por serem esses professores magnifícos, por serem esses lutadores e por confiarem que um dia eu alcançaria.

À minha irmã Renata Gerhardt Gomes Roza, por desde pequena me acompanhar em cada passo, por estar comigo nas mais difíceis decisões, por me ajudar a qualquer hora e não medir esforços para tornar possível a realização dos meus sonhos.

A todos os demais que contribuíram para a realização deste trabalho.

Ao Prof. Tiago Araújo Neves do VCE/EEIMVR/UFF pela gentileza, dedicação e disponibilidade em sua orientação, por compartilhar comigo todo seu conhecimento.

Resumo

As Redes de Distribuição de Conteúdos incluem grande número de servidores dedicados com o objetivo de oferecer aos provedores de conteúdos de Internet uma arquitetura que move o conteúdo para servidores próximos ao usuário, reduzindo o atraso e o tráfego. Nesta estrutura existem diversos problemas estudados pela literatura, entre eles o Problema de Alocação da Capacidade de Armazenamento (PACA) e o Problema de Posicionamento de Réplicas e Distribuição de Requisições (PPRDR). Este trabalho analisa de forma associada estes problemas e propõe a criação de um novo, nomeado de Problema de Distribuição de Capacidades, Réplicas e Requisições (PDCRR), que permite resolver tanto a alocação dinâmica de espaço nos servidores quanto a distribuição de réplicas, conteúdos e requisições de maneira conjunta. Neste trabalho é criado um novo modelo para resolução do PDCRR juntamente com uma nova formulação matemática, que associa variáveis e restrições presentes em formulações de resolução do PACA e do PPRDR. A virtualização também é utilizada para facilitar a resolução conjunta destes dois problemas. As formulações matemáticas foram analisadas e os testes computacionais executados identicaram que a resolução do PDCRR traz uma redução de custos operacionais e a posibilidade de desabilitação de alguns servidores não utilizados pela rede.

Abstract

The Content Distribution Networks (CDN) include a large number of dedicated servers creating an architecture that moves the content of servers that are closer to the user, reducing delays and traffic. In this structure several problems are studied, including the Problem of Allocation of Storage Capacity (PACA) and the Replica Placement and Request Distribution Problem (RPRDP). This work analyzes these problems in an integrated way and proposes the creation of a new problem named Problem of Distribution of Capabilities, Replicas and Requisitions (PDCRR), which enables both the dynamic allocation of space on the servers and distribution of replicas and requests. As main contributions of this work are the analisys of the mathematical formulations for PACA and RPRDP, the creation of a new problem and a new formulation which associates variables and restrictions presents in matehmatical formulations for this problems. Virtualization is also used to facilitate the resolution of these two problems. Mathematical formulations and were analized and Computational Results shows that the resolution of RPRDP provides operational costs reduction and the possibility of disabling unused servers over the network.

Palavras-chave

1. Redes de Distribuição de Conteúdos
2. Otimização Combinatória
3. Problema de Posicionamento de Réplicas e Distribuição de Requisições
4. Problema de Alocação de Capacidade de Armazenamento
5. Problema de Distribuição de Capacidades, Réplicas e Requisições.

Glossário

CDN	:	*Content Distribution Network*
DNS	:	*Domain Naming Service*
PACA	:	Problema de Alocação da Capacidade de Armazenamento
PPR	:	Problema de Posicionamento de Réplicas
PDR	:	Problema de Distribuição de Requisições
PPS	:	Problema de Posicionamento de Servidores
PDCRR	:	Problema de Distribuição de Capacidades,Réplicas e Requisições
PPRDR	:	Problema de Posicionamento de Réplicas e Distribuição de Requisições
QoS	:	*Quality of Service*
RDC	:	Rede de Distribuição de Conteúdos
RTT	:	*Round Trip Time*
UFF	:	Universidade Federal Fluminense
URL	:	*Uniform Resource Locator*

Sumário

Capítulo 1

Introdução

A demanda por serviços cada vez mais rápidos e confiáveis pela Internet e o aumento do tráfego de informações tem exigido das provedoras de conteúdo alternativas cada vez mais inovadoras para que os serviços sejam entregues da maneira mais eficiente possível ao cliente. Diante deste cenário é possível observar a utilização cada vez mais frequente das Redes de Distribuição de Conteúdos ou CDNs (Content Distribution Network). As CDNs visam o aprimoramento contínuo da performance dos serviços. Para isso as CDNs replicam os conteúdos em servidores estratégicamente posicionados, mais próximos aos usuários, o que propicia uma entrega aos clientes com maior agilidade e menor custo [29].

Para atender os requisitos de desempenho de seus usuários, uma CDN precisa ser confiável, ou seja, é necessário que garanta a entrega de conteúdo completo aos clientes. Uma das possíveis formas para atingir esta confiabilidade é posicionar diversos servidores em regiões geográficas distintas a fim de minimizar a perda de clientes por indisponibilidade da rede. Além disso, também é necessário replicar de maneira eficiente os conteúdos dentro da rede, já que que estes sistemas tendem a encaminhar os clientes para os servidores que possuem o conteúdo e estão em melhores condições de atendê-los. Assim, o desempenho de uma CDN se mostra diretamente ligado com o posicionamento dos conteúdos e com as estratégias de replicação e armazenamento. Uma CDN identifica quais conteúdos serão replicados e a partir de quais servidores bem como onde posicionar as réplicas e em que momento retirá-las ou mantê-las.

Dentro do contexto das CDNs, existem vários problemas de otimização que já foram abordados de múltiplas formas [32, 28, 31, 12, 26]. Este trabalho tem por objetivo geral propor um novo problema de otimização dentro do contexto das CDNs, chamado de Problema de Distribuição de Capacidades, Réplicas e Requisições (PDCRR). Este novo problema contempla a otimização simultânea das capacidades de disco dos servidores, do

posicionamento de réplicas e da distribuição de requisições, resolvendo de maneira conjunta dois problemas de otimização que estão presente no contexto das CDNs, o Problema de Alocação da Capacidade de Armazenamento (PACA) e o Problema de Posicionamento de Réplicas e Distribuição de Requisições (PPRDR). Com o crescimento do volume de dados e a popularidade dos conteúdos, as estruturas de *CloudCDN* [20] (ou CDN em nuvem) lançam novas perspectivas que utilizam a virtualização de rede para facilitar suas operações. A virtualização em nuvem também pode ser utilizada para facilitar a resolução conjunta do PACA e do PPRDR, visto que possibilita a criação de uma infraestrutura sem o alto custo de manutenção e operação geográfica.

Este trabalho possui como objetivos específicos uma análise de dois problemas citados (PACA e PPRDR), a proposição de uma nova formulação matemática de programação linear inteira para o PDCRR e resolução via CPLEX [11]. Os resultados computacionais mostram que é possível resolver estes dois problemas (PPRDR [28] e PACA [32]) simultaneamente e que esta resolução conjunta pode acarretar uma redução dos custos operacionais para as CDNs.

Dentre as contribuições deste trabalho estão:

a. Análise conjunta do PACA e do PPRDR.

b. Análise de formulações matemáticas para o PACA e PPRDR.

c. Proposição de um novo problema de otimização, o PDCRR.

d. Uma formulação Matemática para Resolução do PDCRR, incluindo a distribuição dinâmica de capacidade de armazenamento

O restante deste trabalho está organizado da seguinte maneira: no Capítulo 2 é feita uma breve exposição sobre o funcionamento de uma CDN. O Capítulo 3 faz uma apresentação do problema proposto, descrevendo o PACA e o PPRDR. O Capítulo 4 faz uma explanação dos trabalhos encontrados na literatura que estudaram o contexto deste problema tratado no presente trabalho.O Capítulo 5 apresenta as formulações matemáticas para o PACA e o PPRDR, expõe uma análise conjunta do PACA e PPRDR e propõe a criação de um novo problema: o PDCRR. Também é apresentada uma formulação matemática para resolução deste. No Capítulo 6 são apresentadas as instâncias de teste utilizadas para resolução das formulações apresentadas no Capítulo 5. Já no Capítulo 7 são apresentados os testes computacionais da formulação matemática para resolução do

PDCRR e expõe os resultados obtidos, bem como uma análise dos mesmos, e no Capítulo 8 encontram-se as conclusões do trabalho e algumas propostas para trabalhos futuros.

Capítulo 2

Redes de Distribuição de Conteúdos

Uma Rede de Distribuição de Conteúdos ou Content Delivery Network (CDN) é uma rede de computadores interligados através da Internet, que cooperam de modo transparente para fornecer conteúdo a usuários finais.

Esta rede inclui um grande número de servidores dedicados com o objetivo de oferecer aos provedores de conteúdos uma arquitetura que move o conteúdo para próximo do usuário, permitindo que os provedores atendam seus clientes sem ter que se preocupar em construir uma sólida infra-estrutura de rede para lidar com um tráfego associado [23]. Uma RDC cria cópias, ou réplicas, dos conteúdos inicialmente localizados no servidor original e as distribui entre os servidores espalhados pela rede conforme a demanda de requisições dos clientes.

A Figura 2.1 ilustra como é a estrutura de um sistema cliente-servidor tradicional. Nela são mostrados três clientes (C1,C2 e C3) requisitando dois conteúdos diferentes para um único servidor central (SC). As requisições são indicadas pela letra Q seguida de um número, assim, a sequência *Q1* representa uma requisição para o conteúdo *1*, e a sequência *Q2* representa requisições para o conteúdo *2* [28].

A Figura 2.2 ilustra um possível uso de uma arquitetura de CDN para o cenário proposto na Figura 2.1. Na Figura 2.2 são mostrados os mesmos clientes, com as respectivas requisições da Figura 2.1; porém, o papel do servidor central é desempenhado por uma CDN de dois servidores (*S1* e *S2*). Estes servidores abrigam réplicas dos conteúdos, representadas pela letra R seguidas por um número que representa qual conteúdo está replicado. Assim, a sequência *R1* representa uma réplica do conteúdo *1* [28]. As réplicas dos conteúdos são cópias

Desta forma, o ônus da distribuição dos conteúdos é transferido do provedor de con-

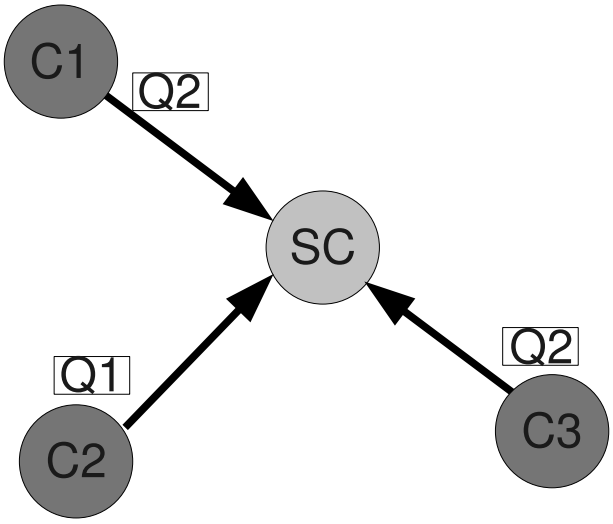

Figura 2.1: Sistema Cliente-Servidor
Fonte: [29]

teúdos para a CDN, fazendo com que o provedor gerencie seu *website* sem ter que se preocupar com espaço de armazenamento, tráfego de rede nem múltiplos acessos simultâneos. Os conteúdos são replicados onde são requisitados e os clientes são atendidos pelas réplicas, e não pelo provedor. Neste sistema, as requisições são enviadas para o provedor, mas o atendimento é feito pela CDN. Assim, as requisições enviadas para o provedor são reencaminhadas para a CDN de modo transparente, ou seja, o sistema da CDN age sem que o usuário e nem o provedor de conteúdos tenham que fazer esforços [23, 1]. Mais informações sobre detalhes operacionais de redes CDN podem ser encontrados em [23].

Visando reduzir a carga dos servidores, uma CDN posiciona os conteúdos mais próximos aos clientes, reduzindo o atraso e ainda o tráfego. Com isso, podem existir várias réplicas de um mesmo conteúdo em servidores diferentes [18]. Além disso, ao estudar padrões de fluxo e informações na rede, um provedor de serviços de Internet deve otimizar a localização dos servidores para um dado perfil de tráfego. Como uma CDN também é um sistema que opera pela Internet, a localização física de seus servidores também deve ser analisada.

Dentro de uma estrutura de CDN existem vários problemas de otimização, dentre os quais se destacam o Problema de Localização de Servidores (PLS), que trata da melhor localização para o posicionamento dos servidores; o Problema de Replicação (PR) que trata dos conteúdos a serem replicados; o Problema do Posicionamento de Réplicas (PPR) que trata da localização das réplicas nos servidores a fim de reduzir custo de transmissão; o Problema de Distribuição de Requisições (PDR) que trata da distribuição das requi-

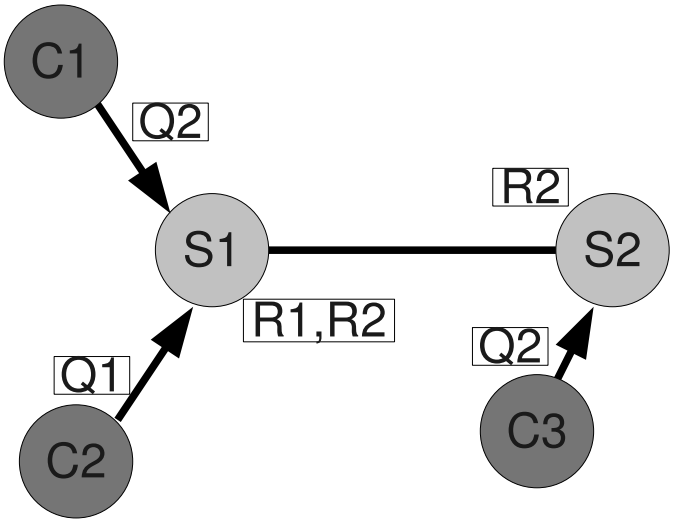

Figura 2.2: Sistema Cliente-Servidor com CDN
Fonte: [29]

sições para um conjunto de servidores. Combinando os dois últimos problemas, tem-se o Problema de Posicionamento de Réplicas e Distribuição de Requisições, ou seja, um problema que trata da localização das réplicas dos conteúdos e definição dos servidores que atenderão às requisições dos usuários finais [28]. O PPRDR pode ser classificado também como um problema dinâmico [28], onde os conteúdos, as condições da rede e as requisições podem ser alterados com o tempo. Além disso, mudanças nos conteúdos devem considerar a atualização das réplicas espalhadas na CDN.

Existe também o Problema de Alocação da Capacidade de Armazenamento (PACA), que consiste em determinar de forma ótima qual proporção do espaço total de armazenamento disponível deve ser alocada nos servidores [32]. Além disso, se bem dimensionada, esta distribuição pode auxiliar na melhoria de performance da CDN pois, ao alocar mais espaço para um servidor com mais requisições, este passa a poder armazenar mais conteúdos, o que aumenta a chance de uma requisição poder ser atendida por este servidor, evitando reencaminhamentos e, por consequência, reduzindo o atraso e a carga nos enlaces da rede [32].

Ao tratar do conceito de alocação de capacidade de armazenamento, um problema atualmente discutido é a possibilidade da alocação dinâmica dos servidores. Neste sentido, torna-se interessante o estudo da virtualização ou *CloudCDNs* [20]. Essa distribuição em nuvem surge para amenizar o problema dos altos custos de manutenção e administração trazidos pelos *hardwares* existentes numa rede. Para isto, este tipo de sistema utiliza softwares para simular a componentes físicos (peças) a fim de criar um sistema virtual

de computadores [7]. Assim, provedores de conteúdos ou CDNs que se utilizam destes serviços podem executar mais de um sistema virtual em um mesmo servidor real. Com base em uma infraestrutura de computação em nuvem que fornece alocação de recursos conforme a demanda, as *CloudCDNs* são capazes de fornecer uma distribuição de conteúdo eficiente em termos de custos [20].

Os provedores em nuvem oferecem recursos de armazenamento e entrega de conteúdos na *Internet* e colocam esses serviços em formas de máquinas virtuais. As empresas que possuem serviço de CDNs podem alugar máquinas virtuais de fornecedores de armazenamento em nuvem e implantar serviços neste tipo de arquitetura de *CloudCDN* [20].

Quando aplicada a servidores, esta estrutura pode ser conceitualizada como um mascaramento de recursos de servidores onde o servidor central utiliza um software para dividir um servidor físicos em múltiplos ambientes, ou servidores privados individuais [9]. Neste contexto são tratadas três abordagens diferentes, das quais é possível citar o modelo de máquina virtual, onde cada servidor executando uma espécie de imitação virtual dos *hardwares* e cada cliente roda sem modificações. Outra abordagem é a de máquina paravirtual, onde a máquina virtual modifica o código do sistema operacional do cliente. Além dessas, pode-se citar o modelo de virtualização do Sistema Operacional (SO) em que o cliente roda um SO e devem usar o mesmo SO que o provedor.

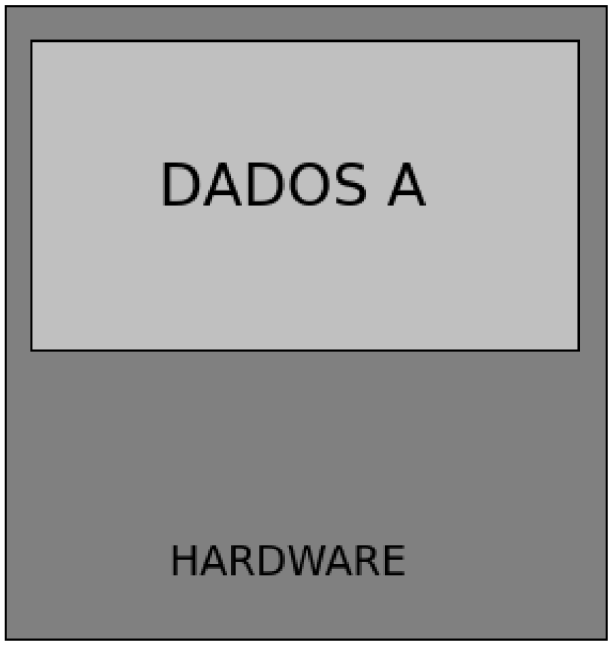

Figura 2.3: Servidor Físico
Fonte: [15]

A Figura 2.3 representa um sistema com servidores dedicados, onde um determinado servidor possui uma capacidade limitada e totalmente dedicado ao uso por um provedor de conteúdos. Desta maneira o *hardware* somente é capaz de hospedar os conteúdos provenientes do possuidor do servidor ou da rede ao qual este servidor faz parte.

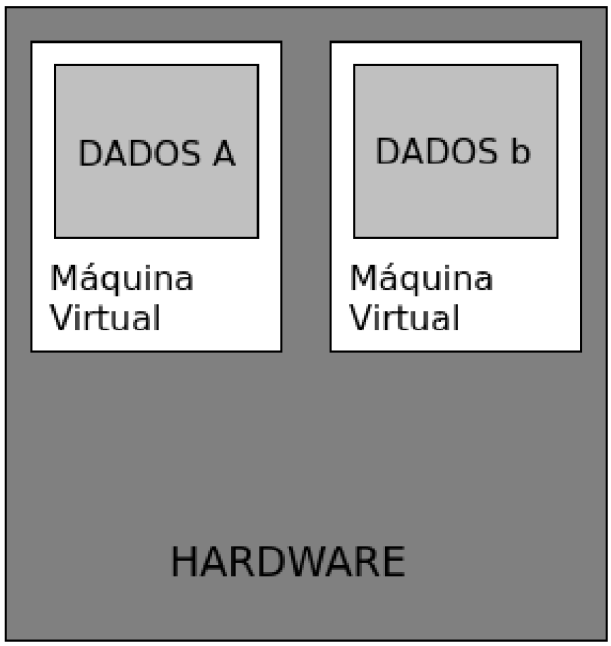

Figura 2.4: Servidor Virtualizado
Fonte: [15]

Já a Figura 2.4 mostra um sistema com utilização de servidores virtualizados. Neste caso, o servidor é dividido em máquinas virtuais, capazes de compartilhar conteúdos de provedores ou redes diferentes. Nesta imagem é possível observar que o *hardware* foi dividido em duas máquinas virtuais e cada um dos clientes ou provedores utiliza máquinas virtuais diferentes, com dados distintos. Nesta abordagem, o espaço em disco disponível para cada uma destas máquinas virtuais poderia ser atualizado conforme a necessidade, desde que a capacidade máxima do servidor seja respeitada. A virtualização de servidores torna o uso dos recursos mais eficientes e melhora a distribuição da capacidade do servidor. Além disso, facilita a recuperações de desastres ou inconsistências dentro da rede e centraliza a administração dos servidores.

Além da virtualização em servidores, outro tipo importante que vale a pena ser analisado é a virtualização de redes [8], que trata da combinação dos recursos disponíveis na rede, dividindo a largura de banda disponível nos canais existentes [8]. Este tipo de virtualização melhora a produtividade, eficiência e administração da rede. Uma característica interessante de ser evidenciada é que a capacidade de armazenamento, neste caso, pode

ser realocada entre os servidores. Vale ressaltar que este tipo de aplicação é mais eficaz em redes que experimentam picos de demanda repentinos e difíceis de se prever. Tais características, como poderá ser visto no próximo capítulo, estão presentes no PACA, no PPRDR e também no problema tratado neste trabalho, o que torna o uso desta tecnologia uma boa opção para a solução real do problema.

Apesar da tecnologia de virtualização se mostrar uma forma eficiente de atacar o problema dos custos, esta abordagem apresenta problemas de segurança. *Hackers* e pesquisadores de segurança mostraram que essas capacidades de virtualização podem ser exploradas para criar novas formas mais robustas de *malware* que são difíceis de detectar e podem escapar às tecnologias de segurança [30, 21]. Esses problemas de segurança se devem principalmente devido à grande diversidade de usuários dividindo as mesmas aplicações e espaço físico de um mesmo *hardware* para funcionar suas máquinas virtuais. Por vezes estes usuários não estão conscientes da localização de seus dados e serviços na rede. Além disso, com o aumento da carga em uma rede, também aumenta a complexidade de gerenciamento e facilitando a existência de *gaps* para entrada desses *malwares*. Visto que a arquitetura da nuvem é muito dinâmica, que existe mudança na carga de trabalho por criar e remover as máquinas virtuais ao longo do tempo e há uma enorme flexibilidade da rede, isto pode potencializar a vulnerabilidade de segurança. É importante mencionar que, apesar de a segurança de sistemas de redes virtuais ser uma preocupação real este tópico não faz parte do escopo deste trabalho. Mais informações sobre este tópico podem ser encontradas em [21].

Capítulo 3

Apresentação do Problema

Certos conteúdos são mais populares em determinadas regiões geográficas do que outras. Além disso, nem sempre é possível alocar servidores de capacidades suficientes para atender os clientes nas proximidades destas regiões. Logo, por muitas vezes faz-se necessário dispender um custo maior para entregar esses conteúdos e tendo o risco de não prestar um serviço de qualidade. Neste contexto, uma das possíveis maneiras de medir o desempenho de uma CDN é analisar o custo despendido para o atendimento das requisições juntamente com o custo gerado pela necessidade de replicação dos conteúdos. O tempo de entrega dos conteúdos aos clientes também é um fator relevante no desempenho das CDNs e portanto também deve ser considerado. Um dos possíveis modos de analisar o tempo de entrega é introduzir uma penalidade por atraso na entrega de um conteúdo. Deste modo, é possível definir um indicador de qualidade para o serviço de uma CDN, somando os seguintes custos:

- custo de atendimento das requisições
- custo de penalidades por atraso na entrega
- custo de replicação de conteúdo

Caso o custo de entrega leve em consideração parâmetros de eficiênica da rede (atraso entre o cliente e servidor de atendimento, variação de atraso, etc), é possível também medir se o cliente está ou não sendo atendido dentro de uma qualidade de serviço adequada. Assim, caso o conteúdo não seja entrege na qualidade desejada, ou com atraso, isto reflete no aumento dos valores deste indicador, que deve sempre ser minimizado.

De maneira prática, pode-se destacar a preocucação das empresas de Tecnologia da Informações durante as Olimpíadas de Londres em 2012. Segundo informações do jornal

O Globo [5] poderia haver perda de receita e de pedidos de clientes devido a transações não sendo executadas em sistemas de missão crítica, além disso, entre 30% e 60% de largura de banda seria consumida por tráfego nos Jogos Olímpicos. O diretor geral da BlueCoat [6], uma empresa que fornece *hardware*, *software* e serviços projetados para segurança cibernética e gerenciamento de rede, destacou que o problema era de difícil solução, visto que a maioria das soluções tecnológicas não contemplavam a identificação de picos de tráfego na rede nem o entendimento da causa raiz dos problemas [5].

Para minimizar os custos de replicação de conteúdos e atendimento de requisições, uma das possíveis estratégias é a resolução do Problema de Posicionamento de Servidores (PPS), cujo objetivo é determinar as posições geográficas dos servidores. Neste problema, os servidores são alocados conforme frequência de acesso e quantidade de requisições, buscando posições mais próximas aos clientes. Porém este problema não trata da otimização das capacidades de armazenamento dos servidores.

Ao utilizar-se uma abordagem para o PPS, para determinar a localização ótima de servidores *proxy*, minimiza-se o atraso de acesso aos conteúdos. Assim, um conteúdo tem maior chance de ser enviado de um servidor próximo ao cliente, sem a necessidade de que este mesmo conteúdo tenha que percorrer todo o caminho desde o provedor, acelerando o tempo de resposta e reduzindo o custo de comunicação. Contudo, ao distribuir conteúdos através dos servidores espalhados pela rede, nem sempre é possível garantir a total utilização do espaço disponível, o que causa subutilização. Por restrições de memória, é impossível que todos os servidores armazenem todos os conteúdos do sistema, logo, deve-se decidir cuidadosamente o que cada servidor deve armazenar para alcançar a máxima eficiência do sistema, assim como decidir quais conteúdos devem ser removidos para aumentar espaço de armazenamento disponível [26]. Para isto, alguns fatores como frequência de acesso e carga de comunicação devem ser considerados.

As abordagens relacionadas ao PPS, como mencionado anteriormente, não consideram a otimização das capacidades de cada servidor. Ao invés disso, elas fazem a alocação de servidores em pontos próximos à demanda. Note que esta alocação pode ser desnecessária, uma vez que, em certos casos, apenas um ajuste na capacidade dos servidores, poderia ser suficiente para resolver a questão. Neste sentido, capacidade em disco de servidores ociosos, poderia ser transferida para servidores que estão próximos de seus limites, tornando assim, este servidor mais adequado para atender a demanda dos clientes próximos a ele, sem a necessidade da instalação de um novo servidor nas proximidades. Por este motivo surge a necessidade da alocação dinâmica do espaço em disco, na qual os servi-

dores podem ser adaptados de maneira mais flexível conforme a necessidade de entrega dos conteúdos. Além disso, esta alocação dinâmica pode ser implementada através da tecnologia de virtualização, como foi comentado anteriormente.

Durante a operação de uma CDN, caso sejam usados servidores com capacidade de armazenamento maior que o necessário, pode haver subutilização dos recursos disponíveis. Para evitar este tipo de ineficiência, uma opção é redistribuir a capacidade total entre os servidores conforme o volume de requisições, aumentando ou reduzindo a capacidade para melhor atender os clientes [32].

Considerando como solucionado o Problema do Posicionamento dos Servidores, ainda existe a necessidade de identificar o melhor posicionamento das réplicas nestes servidores. Com vistas ao atendimento total da demanda, o PPR, ou Problema de Posicionamento de Réplicas, surge com o objetivo de encontrar os melhores servidores para armazenar as réplicas dentro de uma CDN. No PPR as requisições de um mesmo conteúdo serão tratadas por um mesmo servidor.

De forma prática, o atendimento das requisições de maneira aglomerada pode não ser tão vantajoso para o cliente, visto que este pode sofrer com problemas de atraso e perda de qualidade de serviço. Com o objetivo de reduzir a carga de rede sem violar as restrições de QoS das requisições, surge uma outra variante do PPR que tem por objetivo a definição do posicionamento ótimo das réplicas dentro de uma CDN e a distribuição das requisições entre os servidores [28], tratando essas requisições de maneira individual, onde uma mesma requisição pode ser atendida por servidores diferentes, desde que possuam uma cópia do conteúdo. Esta variante é um dos problemas mais gerais no contexto das CDNs, e de grande relevância, conhecido como PPRDR, ou Problema de Posicionamento de Réplicas e Distribuição de Requisições.

O PPRDR é um dos problemas mais gerais estudados na literatura, pois além de contemplar a existência de limites de banda e espaço nos servidores e a multiplicidade de conteúdos, considera que mudanças na rede, nos conteúdos e na demanda podem ocorrer ao longo do tempo. Este problema busca continuamente atender aos critérios de qualidade, utilizando de seus limites máximos para atender sempre melhor seus clientes, quando possível, mas possui a presença de requisitos mínimos de banda e atraso nas requisições [29].

Porém, este problema ainda não trata a questão da alocação da capacidade dos servidores e, considera que a localização dos servidores já é previamente conhecida.

Desta forma, um ponto importante a ser analisado durante a construção de uma CDN é a determinação da capacidade de seus servidores, a fim de tornar servidores aptos para atendimento das requisições, sendo uma característica para garantir o atendimento com qualidade às requisições. Servidores superdimensionados podem acarretar custos às redes, porém se subdimensionados, podem deixar de atender às requisições próximas, levando o usuário a uma espera, às vezes excessiva, por atendimento no servidor designado ou por encaminhamento para outro servidor mais distante.

O Problema da Alocação de Capacidade de Armazenamento surge como alternativa para tratar este tipo de problema, quando busca determinar a proporção do espaço de armazenamento total disponível na rede que deve ser alocada em cada servidor da CDN. Porém, este problema trata apenas de uma otimização prévia do espaço total em disco, sendo realizada uma otimização para todo o horizonte de planejamento. Sendo assim, não considera possíveis alterações que possam ocorrer com os servidores durante todo o horizonte de atendimento de uma CDN. Os resultados do PACA aumentam consideravelmente as chances de uma requisição ser atendida por um servidor próximo, que possua uma cópia do conteúdo, reduzindo assim o atraso e ainda a carga nos enlaces da rede. Mas, conforme mencionado em [32], devido à crescente demanda por serviços multimídia, o PACA pode se tornar um problema crítico, pois alguns tipos de conteúdos possuem um tamanho considerável, podendo facilmente esgotar a capacidade dos servidores rapidamente.

Num contexto de globalização em que os conteúdos são compartilhados continuamente na rede e tendo em vista que a demanda pode mudar drasticamente em um curto prazo de tempo, torna-se cada vez mais complicado determinar o tamanho dos servidores alocados em suas respectivas regiões. Logo, o uso de servidores compartilhados e a virtualização torna-se cada vez mais comum e, junto a isso, novos problemas de otimização de redes devem ser analisados.

Com base nas lacunas identificadas em cada um dos problemas de otimização analisado e diante do cenário crescente da demanda por conteúdos cada vez mais diversificados, surge a necessidade de analisar de maneira conjunta esses problemas. O PPRDR, que é um dos problemas mais gerais dentro do contexto das CDNs, ainda não trata da questão do posicionamento dos servidores, e ainda considera de forma fixa a capacidade destes, reduzindo a flexibilidade de rede, podendo trazer custos de atraso ou perda de qualidade de serviço no atendimento das requisições. O PACA trata das questões de otimização da capacidade dos servidores, porém de forma não integrada ao PPRDR, sendo assim, cria um cenário ótimo, mas que atenda a rede de maneira geral, uma otimização é realizada

mas serve para todo o horizonte de planejamento.

Observando estes dois últimos problemas, é possível descrever um novo problema dentro do contexto das CDNs, chamado de Problema de Distribuição de Capacidades, Réplicas e Requisições, ou PDCRR. O PDCRR trata de maneira integrada a questão do posicionamento das réplicas nos servidores e a distribuição das requisições, mas paralelamente realizada a alocação dinâmica da capacidade de armazenamento dos servidores, as quais podem ser otimizadas para atender as variações de demanda nos diferentes períodos de tempo.

Neste contexto, as características do PDCRR, como abordado neste trabalho, são:

- Tratamento individual de requisições
- Requisições podem surgir ao longo do horizonte
- Capacidade de armazenamento pode ser alterada de acordo com a demanda
- Armazenamento dos servidores flexível ao longo do tempo
- Problema *offline*

No presente trabalho, considera-se que as informações são conhecidas a priori, ou seja, as mudanças que ocorrem na rede, nos conteúdos e nas demandas são conhecidas assim como também é conhecido quando estas mudanças ocorrerão o que faz com que o problema tratado no presente trabalho seja um problema *offline*. O estudo de problemas offline é importante, pois, apesar de muitas vezes não poderem ser usadas na prática, suas soluções podem ser utilizadas na obtenção de estratégias e de parâmetros de qualidade para os problemas *online*, onde as mudanças não são conhecidas a *priori*.

Para uma melhor comprensão do PACA, PPRDR e PDCRR, serão apresentadas formulações matemáticas para estes problemas no Capítulo 5. Testes computacionais envolvendo estas formulações foram executados para comprovar sua eficácia. Contudo, para melhor contextualizar o problema com a literatura existente, o próximo capítulo apresenta um apanhado de trabalhos relacionados.

Capítulo 4

Trabalhos Relacionados

Diversos trabalhos tem sido estudados dentro do contexto de otimização das Redes de Distribuição de Conteúdos, porém nenhum deles trata de maneira generalizada todas as características de uma CDN integrando o conceito de alocação dinâmica, como tratado neste trabalho.

Em [24] é realizado um estudo do Problema de Posicionamento de Servidores (PPS) utilizando-se de programação dinâmica, na qual determina a localização ótima dos servidores, minizando o atraso ao localizar e acessar conteúdos em CDNs. Porém, este problema não é conclusivo quanto ao dimensionamento dos servidores na rede. O PPS apenas trata da questão do posicionamento físico dos servidores, sem se preocupar com a questão da alocação de capacidade.

Várias metodologias foram estudadas para tratar o Problema de Alocação da Capacidade de armazenamento. Entre os trabalhos que tratan deste assunto, podemos citar [22, 32].

Nikolaos Laoutaris, Vassilios Zissimopoulos e Ioannis Stavrakakis em 2004 [22] apresentaram o PACA como possível solução para o problema de otimização de capacidade de armazenamento dentro do contexto das CDNs para topologias hierárquicas. A partir desta formulação, os autores formularam uma modelagem matemática sob forma de Problema de Programação Linear (PPL) para resolução do PACA. Em [22] é utilizada uma aplicação do PACA para topologias hierárquicas ou em árvore, isto é, uma série de galhos interconectadas onde as informações são buscadas em um servidor central e, caso não haja acesso direto são realizados redirecionamentos a fim de buscar o servidor que o possua. Neste trabalho foram analisados alguns modelos que restringem a taxa de requisições atendidas por unidade de tempo, e permitem que requisições sejam redirecionadas

por servidores de mesmo nível hierárquico, fato este que potenciaiza a redução do custo da solução quando este custo entre servidores de mesmo nível herárquico é inferior. Neste problema, primeiramente é decido onde instalar o servidor, após isto é definido o quanto de capacidade de armazenamento alocar em cada um deles e então, quais conteúdos serão armazenados em cada servidor. Desta maneira, passa a ter o uso de memória de forma mais eficiente e os algoritmos propostos podem ser úteis para regular a utilização de armazenamento em cada reserva de memória. Logo, o objetivo do modelo é desenvolver algoritmos que aproximam a performance ótima em vários cenários, porém que o faça de maneira mais eficiente possível. Para isso, são utilizadas técnicas de balanceamento de carga, ou seja, que evitam a sobrecarga de alguns nós a subutilização de outros. Ainda em [22] foi prosposta a implementação do conceito de balanceamento de cargas entre os servidores de CDN, através da inclusão de mais uma restrição no PPL inicialmente proposto, limitando um número máximo de requisições que um servidor pode atender. Além disso, outro conceito relevante estudado em [22] foi uma modificação proposta no modelo a fim de que servidores de mesmo nível hierárquico poderiam encaminhar requisições uns para os outros, técnica conhecida como *request peering*, aumentando a complexidade da formulação [32]. Através de resolução por um método heurístico, foi constatado redução de até 50% no custo total da solução quando os custos de comunicação entre os servidores de um mesmo nível hierárquico é suficientemente baixo.

Baseado na formulação proposta em [22], Uderman em [32] propõe modificações no modelo para contemplar as características da topologia de rede sem hierarquia definida, onde todos os servidores estão aptos a interagir entre si e qualquer servidor pode ser designado para atender uma requisição, podendo ocorrer o atendimento simultâneo entre servidores. Uderman [32] ao tratar do Problema da Alocação da Capacidade de Armazenamento, em seu trabalho, destaca que este tipo de tratamento do espaço alocado nos servidores aumentam as chances de uma CDN ser capaz de realizar a entrega do conteúdo solicitado a partir de um servidor próximo ao cliente, visto que, conforme a demanda, esse servidor pode vir a ter maior espaço alocado. Além disso, reduz problemas de atrasos e diminui a carga de enlaces nas redes. Desta maneira, Uderman e outros [32] inferem que em redes sem topologia hierárquica definida, onde teoricamente qualquer servidor é capaz de atender qualquer requição, poderá ocorrer uma redução ainda maior do custo total, visto que a quantidade de servidores que poderão atender requisições é ampliada quando necessário.

Com a utilização da alocação dinâmica de capacidades, surgem também os estudos sobre virtualização e otimização das capacidades. Em [27] ocorre a implementação de

servidores de uma CDN em máquinas virtuais que podem ter seus recursos alocados dinamicamente.

O trabalho [14] afirma que a capacidade total disponível para o serviço da CDN pode ser realocada entre os servidores conforme as taxas de requisição por diferentes conteúdos oscila em uma CDN. Assim, evita a subutilização dos servidores e torna possível o atendimento da demanda cada vez mais crescente por conteúdos.

Os autores de [25] tratam de análises em topologias hierárquicas, estudando a ideia do cache corporativo, em que alguns nós devem ter os conteúdos carregados na memória. Desta maneira, um conteúdo pode ser enviado de um servidor vizinho sem percorrer o caminho até o servidor central, acelerando o tempo de resposta e reduzindo o custo de comunicação, além de rezudir a carga de conteúdos nos servidores. Por questões de restrições de carga, é impossível que todos os servidores armazenem todos os conteúdos do sistema, logo, tem-se por objetivo decidir em qual servidor armazenar um conteúdo para alcançar a máxima eficiência do sistema.

Com o aumento de volume e popularidade dos conteúdos as estruturas de CDNs por si só podem se tornar inadequadas. Neste contexto, uma arquitetura de *CloudCDN* [20] pode apresentar novas perspectivas. Uma *CloudCDN* é construída numa infraestrutura de fornecimento de capacidade em nuvem. Conforme [20], comparando com uma estrutura tradicional de CDN, este tipo de arquitetura em nuvem pode prover dois benefícios: personalização da estrutura sem alto custo de aquisição ou operação de servidores geograficamente dispersos; pequenas companhias tem possibilidade de alugar um serviço de CDN em nuvem de um fornecedor de armazenamento em nuvem, utilizando-se de um tipo de pagamento por demanda, reduzindo custos.

Em seu trabalho, Chen *et al.* [16] discursa sobre a construção de CDNs sobre estruturas em nuvem e Wu *et al.* [34] trata de padrões de demanda e acesso de usuários em aplicações tradicionais de video por demanda (VoD), além de desenvolver serviços em plataformas em nuvem. As CDNs em nuvem são capazes de de prover redução no custo operacional e eficiência na distribuição dos conteúdos [20].

Além disso, em [33], os autores tratam um tipo de CDN acadêmica, conhecida como CoDeeN, onde os clientes configuram seus navegadores a usar servidor CoDeeN, que atua como um *proxy forward*. Assim, erros de cache são desordenados e redirecionados para outro proxy CoDeeN que atua de modo inverso, concentrando pedidos de um provedor particular. Deste modo, menos solicitações são encaminhadas ao provedor origem. O CoDeeN não opera em servidores dedicados e com recursos confiáveis, nem emprega um

centro de operações de rede para coletar e distribuir informações de status. Cada instância monitora a saúde do sistema e fornece os dados de redirecionamento de requisições locais.

Um dos problema mais completos em se tratando do estudo de otimização de CDNs é apresentado por Neves [28] e conhecido como Prooblema de Posicionamento de Réplicas e Distribuição de Requisições (PPRDR). Neves apresenta formulações exatas utilizadas como métodos para resolução das versões *offline* do PPRDR, supondo que as mudanças que irão acontecer são previamente conhecidas. Para entendê-lo melhor, Neves inicia seu trabalho tratando de formulações mais simples, acrescentando progressivamente restrições, sofisticando o problema, até obter uma formulação mais próxima ao real. A formulação de Neves foi aplicada com sucesso na resolução de problemas de até 50 servidores e aproximadamente três mil requisições, contudo, estes números são pequenos quando comparados aos de CDNs reais de grande porte [35, 17, 13]. O grande limitador para o uso desta formulação é o volume de memória utilizado pelo conjunto de restrições, que cresce exponencialmente com o aumento de servidores e requisições.

Uma tentativa de reduzir o consumo de memória nas formulações de Neves foi feito por Gerhardt [18, 19], onde foram detectadas dois conjuntos de restrições redundantes na formulação matemática, denotada por *Formulação Dinâmica* (FD), proposta por Neves. A retirada das restrições redundantes ocasionou uma redução de 6% no volume de equações do problema, mas ainda sim não foi possível resolver instâncias com mais de 50 servidores.

Como pode ser observado ao longo deste capítulo, tanto o PACA quanto o PPRDR foram abordados através formulações matemáticas. Estas formulações serão discutidas em maiores detalhes no próximo capítulo, e também serão utilizadas como base para uma nova formulação matemática para o PDCRR, tratado neste trabalho.

Capítulo 5

Formulações Matemáticas para o Problema Proposto

A modelagem para o PDCRR deve conter características das modelagens propostas para o PACA e o PPRDR. Neste capítulo são analisadas as formulações matemáticas propostas por Uderman [32] e Neves [28] separadamente. Na seção 5.3 é apresentado a proposta deste trabalho, uma nova formulação para solucionar de maneira mais geral e atualizada múltiplos problemas identificados em uma CDN.

5.1 Problema de Alocação de Capacidade de Armazenamento

Com o aumento do uso de conteúdos na Internet e a disponibilidade de conteúdos cada vez mais distintos para atender a diversidade de demanda dos clientes e o crescimento crescente desta demanda, torna-se necessário que a capacidade de armazenamento dos servidores seja realizada de forma adequada tornando relevante a restrição de espaço nos servidores.

Para tentar resolver esses problemas, Uderman [32] estudou uma série de problemas matemáticos que tratam separadamente estes conceitos e montou uma formulação matemática para solução do PACA, adequada para CDNs sem uma topologia hierárquica definida e capaz de suportar conteúdos de tamanho não unitário, reduzindo o número de variáveis necessárias para descrever o modelo e reduzindo a complexidade.

Uderman definiu o problema da seguinte forma: Seja ψ o conjunto de conteúdos, onde cada conteúdo k tem tamanho L_k ; E a capacidade total de armazenamento de uma

CDN; J o conjunto de clientes, onde cada cliente j possui uma taxa de requisição λ_j e uma distribuição de demanda p_j; um conjunto V de servidores onde a distância entre o cada cliente j e o servidor v é dada por $d_{j,v} : J$ x V. O PACA tem por objetivo minimizar a distância entre o cliente e o conteúdo requisitado, dado altas taxas de demanda pelo conteúdo e alto índice de requisições dos clientes, definindo a alocação ótima do espaço total de armazenamento disponível E entre os servidores do conjunto V.

Utilizando uma abordagem estática, na qual existe um único período de tempo, a alocação ótima da capacidade total disponível em uma CDN pode ser obtida pela formulação proposta por Uderman em 2012 [32] para resolução do PACA, dada por:

Variáveis:

- $\delta_v(k) = \begin{cases} 1, & \text{se o conteúdo } k \text{ está armazenado no nó } v \\ 0, & \text{caso contrário} \end{cases}$
- $X_{j,v}(k) =$ o cliente j recebe o conteúdo k a partir do servidor v

Constantes:

- E a capacidade total de armazenamento;
- L_k o tamanho do conteúdo k;
- J o conjunto de clientes;
- U uma constante qualquer, maior que o valor absoluto de J;
- V o conjunto de servidores;
- ψ o conjunto de conteúdos;
- λ_j é o valor esperado que modela a taxa de requisição do cliente, que é uma função de probabilidade discreta j;
- $p_j(k) \in [0, 1]$ a fração de de requisições do cliente j pelo conteúdo k.
- $p_j = \{p_j(k) : \sum_{k \in C} p_j(k) = 1$, de acordo com uma distribuição de probabilidade discreta$\}$.
- $d_{j,v}$ a distância entre o cliente j e o servidor v;

- d_j a distância entre o cliente j e o servidor origem.

As seguintes equações podem descrever o PACA:

$$Max \quad \sum_{j \in J} \lambda_j \sum_{k \in \psi} p_j(k) \sum_{v \in V} (d_j - d_{j,v}) X_{j,v}(k) \tag{5.1}$$

$$S.a.$$

$$\sum_{j \in J} X_{j,v}(k) \leq U \cdot \delta_v(k), \ \forall v \in V, \forall k \in \psi, U \geq \|J\|, \tag{5.2}$$

$$\sum_{v \in V} \sum_{k \in \psi} \delta_v(k) \cdot L_k \leq E. \tag{5.3}$$

O objetivo da problema (5.1) é minimizar a distância entre o cliente e o conteúdo a ser solicitado, visto que a distância entre o cliente o o servidor origem será sempre menor ou igual à distância entre o cliente k e o servidor v que atende à requisição, o terceiro termo da função objetivo será sempre menor ou igual a zero. Esta minimização ocorrerá principalmente quando a taxa de requisição desse cliente e a demanda são altas. As restrições (5.2) representam que um conteúdo k só pode ser entregue para o cliente caso esteja presente no servidor v e as restrições (5.3), limitam o capacidade alocada em todos os servidores à capacidade total de armazenamento disponível.

Este modelo não contempla a capacidade de rede dos servidores, é um modelo estático e, por isso, o uso direto deste na solução de problemas mais realistas não se torna o mais adequado.

5.2 Problema de Posicionamento de Réplicas e Distribuição de Requisições

A capacidade de armazenamento nos servidores, determinada pelo PACA, tem influência direta no PPRDR. O PPRDR consiste em determinar o posicionamento ótimo das réplicas dos conteúdos na rede e distribuir as requisições pelos servidores, a fim de reduzir o custo de funcionamento da rede, atendendo aos critérios de qualidade. Ao se fazer uma réplica, o conteúdo é copiado integralmente no outro servidor.

Para distribuir as requisições, o PPRDR considera que esta só será redirecionada para um servidor que possua uma cópia do conteúdo solicitado, averiguando as restrições de

qualidade que, se violadas, devem penalizar o modelo. Desta maneira, em certas situações pode ser necessário alterar o número de réplicas e alterar a distribuição. Já que o objetivo do PPRDR é melhorar a qualidade de serviço, em alguns casos pode ser necessário usar um servidor mais distante, porém menos carregado.

Assim, de forma simplificada, dado um conjunto R de requisições dos clientes a serem atendidas; um conjunto S de servidores da CDN e C conjunto de conteúdos replicados, deve-se determinar a melhor localização dessas réplicas nos servidores e a redistribuição das requisições pelo servidores origem de forma a minimizar as penalidades p_{it} por atraso nas entregas dos conteúdos k aos clientes e respeitando aos limites de banda máxima e espaço disponível do servidor e da requisição.

Uma formulação do PPRDR proposta por Neves em 2011 considera as demandas divisíveis, não havendo diferença entre os custos de replicação de um conteúdo. Além disso, trabalha com conteúdos múltiplos, considerando que cada produto é associado a um conteúdo. Esta formulação faz uso da técnica de *backlog*, que obriga o atendimento de alguma das partes da demanda nos períodos subsequentes caso a demanda não seja completamente atendida no período corrente. Considera múltiplas origens para um conteúdo e de atualização, baseando a abordagem em tempo de vida para os conteúdos [28]. Esta formulação foi ainda modificada por Gerhardt em trabalhos posteriores, como citado acima.

Nas análises realizadas por Gerhardt [18, 19], foram detectadas duas restrições redundantes na formulação matemática proposta por Neves. Uma delas indica que o atendimento de cada requisição deve ser completo. Porém foi identificado que outra restrição já contemplava este atendimento, pois indica que a soma das quantidades entregues em um dia, somada à quantidade que será postergada para o próximo dia é equivalente à demanda do dia mais a demanda postergada do dia anterior. Como toda a demanda é sempre suprida e a soma das demandas de cada requisição é igual ao tamanho do conteúdo, estas restrições garantem o atendimento em sua totalidade. Outras restrições, também redundantes, exigem que, no período de submissão de um conteúdo, exista uma réplica no servidor origem deste conteúdo. Porém, esta restrição também é redundante nesta formulação. Quando outras restrições impedem que réplicas de um conteúdo surjam fora da origem no período de submissão, as variáveis de replicação para um determinado conteúdo ficam com valores em aberto. Como as demais variáveis de replicação para o conteúdo possuem valores definidos pelas restrições, a minimização da função objetivo induz a criação de réplicas para atender às requisições, que leva à criação de uma réplica

do conteúdo na origem.

A Formulação de Neves modificada por Gerhardt [18, 19] pode ser descrita como segue:

- x_{ijt} =fração do conteúdo solicitado pela requisição i entregue pelo servidor j no período t

- $y_{kjt} = \begin{cases} 1, & \text{se e somente se o conteúdo } k \text{ está replicado no servidor } j \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$

- b_{it} = *backlog* da requisição i no período t

- $w_{kjlt} = \begin{cases} 1, & \text{se e somente se o conteúdo } k \text{ é copiado pelo servidor } j \text{ a partir} \\ & \text{do servidor } l \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$

Constantes:

- c_{ijt} custo de atendimento da requisição i no servidor j, no período t.
- q_{it} penalidade por usar *backlog* da requisição i no período t.
- h_{kjlt} o custo de replicar o conteúdo k no servidor j a partir do servidor l no período t.
- R conjunto de requisições a serem atendidas.
- S conjunto de servidores da CDN.
- C conjunto de conteúdos replicados.
- T conjunto de todos os períodos.
- L_k o tamanho do conteúdo k (em bytes).
- AS_j espaço em disco disponível no servidor j (em bytes).
- MB_j banda máxima do servidor j (em bytes/segundo).
- BR_i exigência de banda da requisição i (em bytes/segundo).
- $G(i)$ o conteúdo exigido pela requisição i.

A formulação matemática é dada por:

$$Min \quad \sum_{i\in R}\sum_{j\in S}\sum_{t\in T} c_{ijt}x_{ijt} + \sum_{i\in R}\sum_{t\in T} q_{it}b_{it} + \sum_{k\in C}\sum_{j\in S}\sum_{l\in S}\sum_{t\in T} h_{kjlt}w_{kjlt} \tag{5.4}$$

S.a.

$$\sum_{j\in S} L_{G(i)}x_{ijt} - b_{i(t-1)} + b_{it} = D_{it},\ \forall i \in R, \forall t \in [B_{G(i)}, E_{G(i)}], \tag{5.5}$$

$$\sum_{i\in R} L_{G(i)}x_{ijt} \leq \delta MB_j,\ \forall j \in S, \forall t \in T, \tag{5.6}$$

$$\sum_{j\in S} L_{G(i)}x_{ijt} \leq \delta BX_i,\ \forall i \in R, \forall t \in T, \tag{5.7}$$

$$y_{G(i)jt} \geq x_{ijt},\ \forall i \in R, \forall j \in S, \forall t \in T, \tag{5.8}$$

$$\sum_{j\in S} y_{kjt} \geq 1,\ \forall k \in C, \forall t \in [B_k, E_k], \tag{5.9}$$

$$y_{kjt} = 0,\ \forall k \in C, \forall j \in S, \forall t \notin [B_k, E_k], \tag{5.10}$$

$$y_{kjB_k} = 0,\ \forall k \in C, \forall j \in \{S | j \neq O_k\}, \tag{5.11}$$

$$y_{kj(t+1)} \leq \sum_{l\in S} w_{kjlt},\ \forall k \in C, \forall j \in S, \forall t \in T, \tag{5.12}$$

$$y_{kjt} \geq w_{kljt},\ \forall k \in C, \forall j, \forall l \in S, \forall t \in T, \tag{5.13}$$

$$\sum_{k\in C} L_k y_{kjt} \leq AS_j,\ \forall j \in S, \forall t \in T, \tag{5.14}$$

$$x_{ijt} \in [0, 1],\ \forall i \in R, \forall j \in S, \forall t \in T, \tag{5.15}$$

$$y_{kjt} \in \{0, 1\},\ \forall k \in C, \forall j \in S, \forall t \in T, \tag{5.16}$$

$$b_{it} \geq 0,\ \forall i \in R, \forall t \in T, \tag{5.17}$$

$$w_{kjlt} \in \{0, 1\},\ \forall k \in C, \forall j \in S, \forall l \in S, \forall t \in T. \tag{5.18}$$

A função objetivo (5.4) minimiza o custo da entrega dos conteúdos aos clientes, a quantidade de backlogs feitos ao longo do tempo e o custo de replicação. As restrições indicam que a soma das quantidades entregues no período atual somada à quantidade que será entregue no próximo período equivale à demanda atual mais a demanda atrasada do período anterior (5.5); exigem que o fluxo total entregue pelo servidor seja menor ou igual à sua capacidade máxima (5.6); impedem que cliente receba uma banda maior que a suportada (5.7); condicionam que as requisições só podem ser atendidas pelo servidor que possui réplica do conteúdo (5.8); controlam a existência de no mínimo uma réplica durante o tempo de vida do conteúdo, sendo que nenhuma deve existir fora desse tempo (5.9) e (5.10); possibilitam que durante o surgimento de um conteúdo somente o servidor de

origem possua uma réplica (5.11); garantem que a replicação não seja intantânea (5.12); exigem que a replicação deve ocorrer a partir de um servidor que possua o conteúdo replicado (5.13) e ainda, garantem a capacidade de disco dos servidores não seja violada (5.14) [28] [18].

A formulação FD proposta por Neves considera as capacidades de cada servidor como constantes, desta maneira, não contempla a análise do espaço em disco utilizada por cada servidor. A inclusão desta análise será proposta posteriormente.

5.3 Problema de Distribuição de Capacidades, Réplicas e Requisições

Em redes que os usuários são mais rigorosos com a qualidade de serviço, com exigências de banda e atraso, a capacidade dos servidores pode acabar sendo exaurida tornando assim importante o estudo do PACA. No entanto, as capacidades de armazenamento dos servidores influenciam na resolução do PPRDR, uma vez que estas capacidades podem ser consideradas parâmetros para o gerenciamento de réplicas e requisições.

Devido à interdependência entre o PACA e o PPRDR, ambos os problemas foram estudados por Uderman [32], a fim de que os resultados do primeiro fossem diretamente utilizados como dados de entrada para o segundo. Nesta abordagem, o PACA é utilizado para encontrar uma distribuição otimizada do espaço de armazenamento para cada servidor de CDN. Em seu trabalho, Uderman utiliza duas abordagens. Na primeira o PACA é resolvido utilizando taxas médias das requisições, porém os resultados obtidos pela resolução do PACA são utilizados apenas uma vez, no início da operação da CDN. Na segunda, o PACA é resolvido para cada período do horizonte de planejamento, considerando como dados de entrada o estado atual da CDN. Entretanto, esta segunda abordagem continua tratando os dois problemas (PACA e PPRDR) de maneira isolada, o que faz com que as mudanças ocorridas ao longo do tempo não sejam observadas com o devido cuidado.

Modificando a formulação do PPRDR proposta por Neves [28], e mais tarde revisada por Gerhardt [19], através da inclusão de algumas restrições, é possível fazer uma formulação que também consegue lidar com o conceito de otimização do espaço de armazenamento. Esta formulação, pode ser utilizada para resolver um novo problema de otimização de CDNs, nomeado como Problema de Distribuição de Capacidades, Réplicas e Requisições (PDCRR), onde o objetivo principal é a alocação ótima da capacidade de armazenamento disponível, distribuição das réplicas de conteúdos e das requisições de

clientes nos servidores, a fim de otimizar a utilização dos recursos e reduzir os custos operacionais, mantendo os padrões de qualidade de serviço (QoS). Deste modo, o PDCRR pode ser visto como um problema mais geral, que engloba o PPRDR e o PACA.

Desta forma, utilizando as equações de Uderman e de Neves modificada, pode-se obter uma nova formulação matemática para o PDCRR.

A formulação de Uderman [32] é uma formulação mais simples e que trata de menos conceitos que a formulação para o PPRDR construída por Neves [28]. Portanto, a fim de evitar maiores dificuldades para inclusão de todos esses conceitos na formulação para o PACA, torna-se mais coerente realizar modificações na formulação de Gerhardt [19] para contemplar os aspectos faltantes e criar uma nova formulação que trate de forma mais geral todos estes conceitos.

Na formulação de Neves as capacidades de cada servidor são constantes. Porém, para o PDCRR, a capacidade de cada servidor passa a ser variável. A formulação proposta neste trabalho utiliza variáveis r_{jt} que representam o espaço em disco alocado no servidor j, durante o período t, ao invés das constantes AS_{jt} da formulação de Neves. Uma versão particular desta formulação utiliza variáveis r_j, para o caso onde a alocação de espaço nos servidores deve ser homogênea ao longo do tempo. No caso homogêneo, a capacidade de cada servidor deve assumir o mesmo valor em todo o horizonte de planejamento.

Portanto, para incluir os conceitos do PACA na formulação de Neves, faz-se necessário as seguintes alterações:

- Retirada das restrições (5.14);
- Inclusão das restrições (5.19) para o problema onde a alocação deve ser homogênea no tempo ou (5.20) para o caso onde a alocação pode mudar com o tempo;

$$\sum_{k \in C} L_k y_{kjt} \leq r_j, \ \forall j \in S, \forall t \in T, \tag{5.19}$$

$$\sum_{k \in C} L_k y_{kjt} \leq r_{jt}, \ \forall j \in S, \forall t \in T, \tag{5.20}$$

- Inclusão das restrições (5.21) para o problema onde a alocação deve ser homogênea no tempo ou (5.22) para o caso onde a alocação pode mudar com o tempo

$$\sum_{j \in S} r_j \leq E, \tag{5.21}$$

$$\sum_{j \in S} r_{jt} \leq E, \ \forall t \in T, \tag{5.22}$$

Sendo E é o espaço total disponível para distribuição entre os servidores.

A formulação matemática proposta neste trabalho foi baseada nas formulações PPRDR modificado e realizada sob a forma de um Problema de Programação Linear (PPL), com as alterações necessárias para incluir os conceitos do PACA para resolver o problema de maneira conjunta.

O PPL apresentado como método de resolução do PDCRR é utlizado para versões *offline* do PDCRR. Vale ressaltar que o estudo de problemas *offline* é interessante pois serve como mecanismo para definição do comportamento da rede, o que permite analisar estratégias de alocação e distribuição mais eficiente para os problemas *online*.

Considerando S o conjunto de servidores da CDN, C o conjunto de conteúdos replicados, R o conjunto de requisições a serem atendidas e E o espaço total disponível para distribuição entre os servidores. Sejam as contantes c_{ijt} custo de atendimento da requisição i no servidor j, no período t, q_{it} penalidade por usar *backlog* da requisição i no período t, h_{kjlt} o custo de replicar o conteúdo k no servidor j a partir do servidor l no período t. Além disso, L_k o tamanho do conteúdo k (em bytes), MB_j banda máxima do servidor j (em bytes/segundo), BR_i exigência de banda da requisição i (em bytes/segundo), $G(i)$ o conteúdo exigido pela requisição i e M uma constante que representa a soma do tamanho de todos os conteúdos.

Define-se também os seguintes conjuntos de variáveis:

- x_{ijt} =fração do conteúdo solicitado pela requisição i entregue pelo servidor j no período t

- $y_{kjt} = \begin{cases} 1, & \text{se e somente se o conteúdo } k \text{ está replicado no servidor } j \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$

- b_{it} = *backlog* da requisição i no período t

- $w_{kjlt} = \begin{cases} 1, & \text{se e somente se o conteúdo } k \text{ é copiado pelo servidor } j \text{ a partir} \\ & \text{do servidor } l \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$

- r_{jt} = espaço alocado no servidor j durante o período t

$$Min \quad \sum_{i\in R}\sum_{j\in S}\sum_{t\in T} c_{ijt}x_{ijt} + \sum_{i\in R}\sum_{t\in T} q_{it}b_{it} + \sum_{k\in C}\sum_{j\in S}\sum_{l\in S}\sum_{t\in T} h_{kjlt}w_{kjlt} \tag{5.23}$$

S.a.

$$\sum_{j\in S} L_{G(i)}x_{ijt} - b_{i(t-1)} + b_{it} = D_{it},\ \forall i \in R, \forall t \in [B_{G(i)}, E_{G(i)}], \tag{5.24}$$

$$\sum_{i\in R} L_{G(i)}x_{ijt} \leq \delta MB_j,\ \forall j \in S, \forall t \in T, \tag{5.25}$$

$$\sum_{j\in S} L_{G(i)}x_{ijt} \leq \delta BX_i,\ \forall i \in R, \forall t \in T, \tag{5.26}$$

$$y_{G(i)jt} \geq x_{ijt},\ \forall i \in R, \forall j \in S, \forall t \in T, \tag{5.27}$$

$$\sum_{j\in S} y_{kjt} \geq 1,\ \forall k \in C, \forall t \in [B_k, E_k], \tag{5.28}$$

$$y_{kjt} = 0,\ \forall k \in C, \forall j \in S, \forall t \notin [B_k, E_k], \tag{5.29}$$

$$y_{kjB_k} = 0,\ \forall k \in C, \forall j \in \{S | j \neq O_k\}, \tag{5.30}$$

$$y_{kj(t+1)} \leq \sum_{l\in S} w_{kjlt},\ \forall k \in C, \forall j \in S, \forall t \in T, \tag{5.31}$$

$$y_{kjt} \geq w_{kljt},\ \forall k \in C, \forall j, \forall l \in S, \forall t \in T, \tag{5.32}$$

$$\sum_{k\in C} L_k y_{kjt} \leq r_{jt},\ \forall j \in S, \forall t \in T, \tag{5.33}$$

$$\sum_{j\in S} r_{jt} \leq E,\ \forall t \in T, \tag{5.34}$$

$$x_{ijt} \in [0, 1],\ \forall i \in R, \forall j \in S, \forall t \in T, \tag{5.35}$$

$$y_{kjt} \in \{0, 1\},\ \forall k \in C, \forall j \in S, \forall t \in T, \tag{5.36}$$

$$b_{it} \geq 0,\ \forall i \in R, \forall t \in T, \tag{5.37}$$

$$w_{kjlt} \in \{0, 1\},\ \forall k \in C, \forall j \in S, \forall l \in S, \forall t \in T. \tag{5.38}$$

$$0 \leq r_{jt} \leq M,\ \forall j \in S, \forall t \in T. \tag{5.39}$$

Assim, a formulação descrita acima tem como objetivo (5.23) minimizar o custo de

entrega dos conteúdos aos clientes bem como a penalidade por atraso de atendimento das requisições e ainda o custo de replicação dos conteúdos nos servidores. As restrições (5.24) garantem que a demanda será totalmente atendida, fazendo uma relação que indica que a quantidade entregue do período atual somada à penalidade do período posterior deverá ser equivalente à demanda do período atual somada às penalidades por atraso do dia anterior. As restrições (5.25) e (5.26) limitam a quantidade de conteúdos entregues à banda máxima permitida pelo servidor e as exigências de banda do cliente. As restrições (5.27) garantem que uma requisição só pode ser atendida por um servidor que possua uma cópia do conteúdo solicitado. As restrições (5.28) e (5.29) controlam as réplicas, garantindo que exista pelo menos uma cópia do conteúdo durante seu período de existência e que não exista nenhuma réplica fora deste período. As restrições (5.30) garantem que durante o surgimento do conteúdo, todos os outros servidores exceto o servidor origem não contenham nenhuma réplica do conteúdo. As restrições (5.31) exigem que seja criada uma nova réplica a cada replicação e as restrições (5.32) exigem que a replicação deve ocorrer a partir de um servidor que possua o conteúdo replicado. As restrições (5.33) indicam que a soma do espaço ocupado em um servidor não deve ultrapassar o espaço alocado neste servidor. As restrições (5.34) garantem que, em cada período, a soma dos espaços em discos alocados nos servidores deve ser menor ou igual que o espaço total disponível. As demais restrições são de integralidade e não negatividade.

5.4 Abordagens Heurísticas

Devido às dificuldades encontradas para a resolução de algumas instâncias, principalmente aquelas com maior quantidade de servidores, tornou-se necessário o estudo de abordagens heurísticas para resolver o problema proposto.

Com o aumento da complexidade dos problemas tratados em um determinado modelo, os métodos exatos podem se tornar inviáveis para a aplicação e obtenção da solução ótima de maneira mais eficiente. Neste contexto surge a necessidade de se tratar alguns problemas de maneira heurística ou meta-heurística, como é proposto a seguir.

Considerando que o atendimento ao cliente deve respeitar os critérios mínimos de qualidade de serviço e que um dos critérios fortemente considerado pelos usuários é o tempo de entrega do conteúdo solicitado, considera-se que o atraso no atendimento de uma requisição deve ser evitado e é fortemente penalizado.

Na formulação original, as requisições que não são atendidas dentro do período re-

quisitado, vão agregando um custo muito alto à função objetivo. Como o objetivo do problema é de minimização, este recurso de atraso no prazo de atendimento é evitado sempre que possível, mesmo que isso implique o uso de servidores que não são capazes de atender a requisição dentro dos outros parâmetros de qualidade.

Na heurística proposta, a estratégia utilizada é justamente a oposta. Servidores que não são capazes de atender uma requisição dentro dos parâmetros de qualidade não são considerados como opção para atender uma requisição, fazendo que nestes casos, seja preferível atrasar o atendimento da mesma.

Uma outra modificação proposta pela heurística é a inclusão de um tempo máximo de atendimento para cada requisição. Na formulação original este atendimento fica em aberto, podendo o atendimento a uma requisição ser prolongado por todo o horizonte de planejamento, caso isto seja vantajoso. A heurística impõe um limite máximo para isso. Este limite é calculado em função do tempo mínimo de atendimento. O tempo mínimo é calculado supondo que o cliente é atendido na capacidade máxima de sua rede local. O tempo máximo é dado por duas vezes o tempo mínimo.

Todas estas mudanças no modelo foram feitas através de mudanças nos limites das variáveis.

Desta forma, as variáveis x_{ijt}, que na formulação original possuem limite inferior igual a zero e limite superior igual a 1 para todas as variáveis, poderá assumir valor nulo já na construção do modelo, se identificado que o servidor j irá gerar um atraso no atendimento que seja maior que o permitido pelos critérios de QoS da requisição i. Durante a construção das variáveis x_{ijt}, foi incluído o seguinte algoritmo:

Foi estabelecido um limite mais forte para as variáveis de *backlog* b_{it}, as quais estão limitadas ao valor máximo da demanda dos clientes, assim, o limite máximo de períodos em que a requisição pode ser atrasada é o dobro da soma do tempo mínimo de entrega com o tempo de chegada do conteúdo. Além disso, foram realizadas várias tentativas de manipulação nas variáveis y_{kjt} e w_{kjlt}, porém todas as tentativas de criação de um mecacnismo para racionalizá-las resultaram em um modelo sem solução para alguns casos, resultando em uma heurística falha.

O Algoritmo 1 mostra de forma mais clara como foi realizada a inclusão das modificações destas variáveis na formulação exata.

O modelo construído a partir das especificações propostas nesta seção foi executado para as instâncias onde o modelo original do PDCRR não conseguiu bons resultados. Os

Algoritmo 1 Definindo Atendimento de Requisições e *Backlog*

1: **para** cada variável x_{ijt} **faça**
2: limite inferior = 0
3: **se** (servidor não atende QoS) **então**
4: limite superior $x_{ijt} = 1$
5: **senão**
6: limite superior $x_{ijt} = 0$
7: **fim se**
8: **fim para**
9: **para** cada variável b_{it} **faça**
10: determinar o tempo mínimo para atender i
11: encontra o tempo máximo t'' em função de t'
12: limite inferior $b_{it} = 0$
13: **se** ($t > t'$) **então**
14: limite superior $b_{it} = 0$
15: **senão**
16: limite superior $b_{it} = +\infty$
17: **fim se**
18: **fim para**

resultados obtidos pela heurística são apresentados no Capítulo 7.

Capítulo 6

Instâncias de Testes

Para testar o modelo proposto pela formulação (5.23), foram utilizadas as instâncias com números variados de servidores, conteúdos e requisições, disponíveis pelo Laboratório de Inteligência Computacional da UFF [10], utilizadas em vários trabalhos [28], [19]. Este conjunto de instâncias é o primeiro conjunto que considera simultaneamente várias características próximas à realidade, como, requisitos de QoS, capacidade de servidores diferentes e conteúdos dinâmicos.

Conforme citado em [28], estas instâncias são divididas em quatro classes, A, B, C e D. As instâncias da classe A são instâncias em escala reduzida, utilizadas para testes e por isso praticamente todos os valores utilizados para esta classe são escolhidos de maneira arbitrária. As Instâncias da classe B são instâncias construídas com base em valores encontrados na literatura e com base em equipamentos de mercado disponíveis quando estas instâncias foram criadas (final de 2008). As instâncias da classe C são instâncias similares às instâncias da classe B, contudo, nas instâncias da classe C os servidores possuem restrições mais rígidas na capacidade de armazenamento. As instâncias da classe D são instâncias com restrições mais severas em termos de capacidade de armazenamento e em termos de banda nos servidores. Para cada possível tamanho de 10, 20, 30 ou 50 servidores, utilizadas para implementação do modelo do PPRDR, foram criadas 5 instâncias de cada classe. Após a realização de vários testes computacionais, foi constatado por Neves que as instâncias das classes A e B são de fácil resolução. Para todas as instâncias, considerou-se que o tamanho dos períodos é de 60 segundos.

Em seu texto [28], Neves detalha como foi o processo de geração das intâncias. Dentre as informações a serem destacadas neste trabalho, vale ressaltar que foram considerados 15 períodos de tempos para as instâncias de classe A e 35 para as demais classes. Para a geração das informações de enlace da rede, atribuiu-se para cada aresta do grafo de

adjacências, um atraso aleatório, variando de 60 a 100 milissegundos e, então, executado o algoritmo de Dijkstra para encontrar a distância mínima entre cada par de servidores, de onde são definidas as distâncias no primeiro período de tempo. Como estudos de medidas de atraso pela Internet reportam atrasos que variam entre 15 e 300 milissegundos, utilizou-se valores dentro deste intervalo para a criação das instâncias.

Arbitrariamente definiu-se que a cada cinco períodos de tempos, um dos enlaces tem seu atraso reconfigurado para um valor entre 60 e 100 ms, e aplicado o algoritmo de Dijkstra novamente [28].

As instâncias pertencentes às classes A, B e C possuem redes simétricas onde o tempo para ir de um servidor ao outro é o mesmo que o retorno. As instâncias de classe D possuem canais assimétricos, ou seja, o tempo gasto para ir de um servidor j para um servidor l pode ser diferente do tempo de ir de l a j. Esta assimetria torna as isntâncias de classe D mais próximas da realidade, visto que as redes de computadores também possuem este tipo de características [28].

Para gerar o espaço em disco de cada servidor foi utilizada a distribuição uniforme de probabilidade com os valores entre 100 e 200 MB para as instâncias de classe A, 100 e 150 GB para a classe B, 3 e 4 GB para a classe C e 2,5 e 3,2 GB para a classe D[28].

Como as instâncias da classe A são instâncias de teste, o intervalo para os tamanhos dos discos foi escolhido arbitrariamente. Já para as instâncias da classe B, o intervalo foi definido de acordo com a literatura [28]. Entretanto, ao perceber que estes valores eram demasiadamente grandes para o número de conteúdos considerados, optou-se por reduzir estes valores para as classes C e D com o objetivo de criar instâncias mais difíceis. Os valores, tanto para a classe C quanto para a classe D foram determinados experimentalmente. Para gerar a banda de cada servidor também foi utilizada a distribuição uniforme de probabilidade com valores oscilando entre 1500 e 2000 MB por período para as instâncias da classe A, 4050 MB por período para as classes B e C, e 2300 e 2350 MB por período para a classe D. Os valores de banda para as classes B e C foram escolhidos com base em equipamentos servidores disponíveis no mercado quando as instâncias foram construídas. Contudo, constatou-se posteriormente que os valores de banda para os servidores utilizados nas Classes B e C geram instâncias em que o PDR é de fácil resolução, e por isso as instâncias da classe D tem o valor da banda dos servidores reduzido para aproximadamente 60% (determinado experimentalmente) dos valores usados nas classes B e C. Os valores usados na classe A foram determinados de maneira arbitrária [28].

Após a geração dos servidores, é feita a geração das informações dos conteúdos. Para

gerar estas informações, a primeira tarefa a ser feita é definir o número de conteúdos permanentes, ou seja, o número de conteúdos que existirão desde o primeiro período de tempo até o último período de tempo, o número de conteúdos voláteis, ou seja, que surgem fora do período inicial e podem ser removidos da CDN antes do período final, e também atribuir a cada conteúdo um identificador único. No trabalho de Neves estabeleceu-se que o número de conteúdos permanentes é de 3 conteúdos para as instâncias da classe A, 10 para as instâncias da casse B e C e 12 para as instâncias da classe D. Já para determinar o número de conteúdos voláteis são utilizados números aleatórios que variam entre 1 e 3 para a classe A e entre 1 e 5 para as classes B e C. Para as instâncias da classe D este número é escolhido de maneira aleatória entre 1 e 7 [28].

Para gerar as requisições considerou-se a necessidade de conhecimento das informações dos conteúdos devido ao fato de um número maior de requisições ser dado aos conteúdos mais populares e também para não gerar requisições para conteúdos que não se encontram disponíveis na CDN. Por conteúdos não disponíveis entenda-se conteúdos que ainda não foram criados ou conteúdos que já foram removidos. Informações sobre o número de servidores também são necessárias já que, quanto maior o número de servidores, maior o número de pontos de entrada para a CDN e consequentemente maior o número de requisições. Outro motivo para se ter as informações sobre o número de servidores no momento da geração das requisições é o fato de que cada requisição é atrelada a um servidor, chamado servidor de origem, que é o servidor ao qual o cliente que gerou a requisição está conectado. Às requisições também são atribuídos atrasos locais, que representam o tempo que os pacotes levam para transitar entre o cliente e o servidor ao qual o cliente está conectado. A cada requisição também é atribuído um servidor aleatoriamente de modo que todos os servidores possuem a mesma chance de escolha. Cada requisição também sabe o período em que ela é recebida pela CDN [28].

As informações sobre o tamanho de cada servidor não torna-se necessário para a resolução do PDCRR, visto que o tamanho alocado em cada servidor neste modelo é uma variável. As instâncias citadas neste capítulo foram utilizadas como base para a criação de um conjunto de novas instâncias para resolução do PDCRR. Este novo conjunto utiliza a soma das capacidades de cada servidor destas instâncias para definir o tamanho do espaço total em disco disponível para a execução do modelo.

A fim de analisar a otimização do espaço do servidores na rede e uma possível subutilização dos servidores, ou ainda a desabilitação destes, foi proposta a criação de um novo conjunto de instâncias.

Para a criação deste novo conjunto de instâncias foram utilizadas como base as informações de instâncias com 20 servidores. Durante a formação do conjunto de requisições, a quantidade foi reduzida em 40%. Sobre estas, foi estabelecido que apenas 15 dos 20 servidores poderiam ser usados como servidores origem das requisições. Assim, somente esses 15 servidores poderiam estar contidos nas informações de origem de requisição. Desta maneira, na reconstrução das requisições, caso determinada requisição estivesse sendo originada em um servidor diferente dos 15 selecionados, era restabelecido um servidor origem entre os 15 escolhidos. Durante o período seguinte do surgimento do conteúdo na rede, esses servidores são disponibilizados para que possam ter réplicas do conteúdo solicitado. Os resultados para este conjunto são analisados no Capítulo 7.

Capítulo 7

Análise da Formulação por Meio de Testes Computacionais

A fim de validar o modelo, foram realizados testes utilizando as instâncias com o PPRDR modificado e o PDCRR proposto neste trabalho. O objetivo dos testes é verificar se a alocação dinâmica de espaço pode acarretar redução dos custos sem que a qualidade de serviço seja comprometida.

A formulação do PPRDR foi reexecutada devido à diferença entre os hardwares utilizados e a nova versão 12.5.1 do CPLEX [11]. Para estes novos testes foi utilizado um hardware com as seguintes características: processador Intel® CoreTM i7-2600K com 8 núcleos, 3.40 GHz por núcleo e 16 GB de RAM e sistema operacional Ubuntu 12.04 LTS.

7.1 PDCRR Dinâmico

A Tabela 7.1 expõe os resultados obtidos nos testes computacionais. A coluna 1 é composta por um conjunto de números que representa a quantidade de servidores da instância analisada, seguida do identificador da instância, divididas nas classes A, B, C e D. A coluna 2 mostra o *status* de solução reportado pelo CPLEX para resolução do PDCRR, podendo ser classificado como resultado ótimo (*Ótimo*), ótimo dentro da tolerância estabelecida de 4 casas decimais (*Ótm. Tol.*) ou melhor solução encontrada dentro do tempo limite configurado neste trabalho como 10800 segundos (*Tempo*). A coluna 3 mostra o *status* de solução apresentado pelo CPLEX para resolução do PPRDR, as classificações correspondem às mesmas da coluna 2. A coluna 4 contém a diferença percentual (*gap*) entre as duas abordagens, obtido pela seguinte equação:

$$\left(\frac{F_1 - F_2}{F_1}\right) \cdot 100 \tag{7.1}$$

Onde F_1 é o resultado da função objetivo para o PPRDR e F_2 é o resultado da função objetivo para o PDCRR. O gap positivo mostra que a solução da função objetivo do PDCRR foi menor que a função objetivo para o PPRDR numa mesma instância, indicando redução de custo operacional.

Tabela 7.1: Comparação entre função objetivo PPRDR e PDCRR - Status Ótimo ou Ótimo na tolerância estabelecida

"Servidor.Instância"	Status PPRDR	Status PDCRR	*GAP* em relação ao PPRDR
10.1	Ótimo	Ótimo	0,00%
10.2	Ótimo	Ótimo	0,00%
10.3	Ótimo	Ótimo	0,00%
10.4	Ótimo	Ótimo	0,00%
10.5	Ótimo	Ótimo	0,00%
10.6	Ótimo	Ótimo	0,00%
10.7	Ótimo	Ótimo	0,00%
10.8	Ótimo	Ótimo	0,00%
10.9	Ótimo	Ótimo	0,00%
10.10	Ótimo	Ótimo	0,00%
10.11	Ótimo	Ótimo	4,55%
10.12	Ótimo	Ótimo	0,85%
10.13	Ótimo	Ótimo	3,57%
10.14	Ótimo	Ótm. Tol.	4,17%
10.15	Ótimo	Ótimo	4,31%
20.1	Ótimo	Ótimo	0,00%
20.2	Ótimo	Ótimo	0,00%
20.3	Ótimo	Ótimo	0,00%
20.4	Ótimo	Ótimo	0,00%
20.5	Ótimo	Ótimo	0,00%
20.6	Ótimo	Ótimo	0,00%
20.7	Ótimo	Ótimo	0,00%

"Servidor.Instância"	Status PPRDR	Status PDCRR	*GAP* em relação ao PPRDR
20.8	Ótimo	Ótimo	0,00%
20.9	Ótimo	Ótimo	0,00%
20.10	Ótimo	Ótimo	0,00%
20.12	Ótimo	Ótimo	2,02%
20.13	Ótm. Tol.	Ótimo	3,52%
20.15	Ótm. Tol.	Ótimo	1,68%
30.1	Ótimo	Ótimo	0,00%
30.2	Ótimo	Ótimo	0,00%
30.3	Ótimo	Ótimo	0,00%
30.4	Ótimo	Ótimo	0,00%
30.5	Ótimo	Ótimo	0,00%
30.6	Ótimo	Ótimo	0,00%
30.7	Ótimo	Ótimo	0,00%
30.8	Ótimo	Ótimo	0,00%
30.9	Ótimo	Ótimo	0,00%
30.10	Ótimo	Ótimo	0,00%
30.11	Ótm. Tol.	Ótm. Tol.	2,08%
30.12	Ótimo	Ótimo	0,85%
30.13	Ótimo	Ótimo	0,86%
30.14	Ótimo	Ótimo	1,01%
30.15	Ótimo	Ótimo	0,88%
50.1	Ótm. Tol.	Ótimo	0,00%
50.2	Ótm. Tol.	Ótimo	0,00%
50.3	Ótm. Tol.	Ótimo	0,00%
50.4	Ótm. Tol.	Ótimo	0,00%
50.5	Ótimo	Ótimo	0,00%
50.7	Ótimo	Ótimo	0,00%
50.12	Ótimo	Ótimo	0,25%

A Tabela 7.1 mostra os resultados para as instâncias que obtiveram *status* ótimo ou ótimo dentro da tolerância. Desta maneira, os *gaps* mostrados na quarta coluna indicam uma redução de até 4,55% de uma solução ótima sobre o PPRDR quando utilizado o PDCRR dinâmico. Estes resultados podem ser melhor visualizados através da Figura 7.1.

Os dados mostrados através da Tabela 7.1 e da Figura 7.1 representam que a melhor utilização dos servidores pode reduzir os custos de distribuição dos conteúdos na rede, pois a alocação do espaço é feita de acordo com o volume de requisições. A alocação dinâmica do espaço propicia a redução das distâncias entre os clientes e os servidores que os atendem, principalmente quando o volume das requisições é elevado, pois possibilita que os servidores utilizados como origem das requisições sejam melhor utilizados, evitando reencaminhamentos desnecessários, reduzindo os atrasos e garantindo a qualidade de serviço.

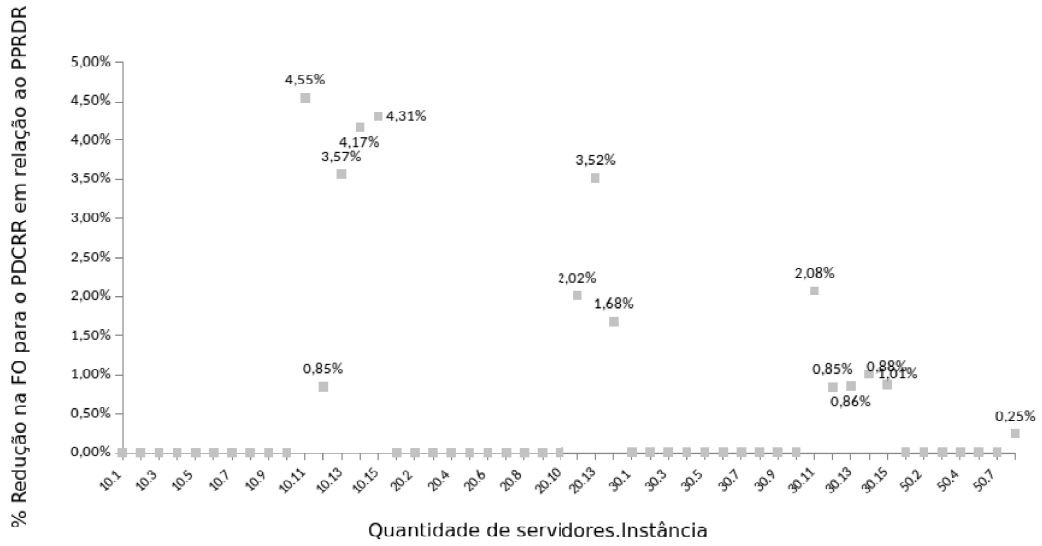

Figura 7.1: *Gaps* encontrados nas instâncias que obtiveram *status* **ótimo** ou **ótimo dentro da tolerância**

Alguns resultados apresentaram um *gap* de 0% na melhor solução encontrada. Estes resultados se encontram entre as instâncias pertencentes às classes A e B, as quais são as mais simples e principalmente usadas para testes. Nestes casos, ambas as abordagens encontram com facilidade as melhores soluções possíveis.

Tabela 7.2: Comparação entre função objetivo PPRDR e PDCRR - Status de Tempo Limite Atingido

"Servidor.Instância"	Status PPRDR	Status PDCRR	*GAP* em relação ao PPRDR
10.16	Ótm. Tol.	Tempo	2,08%
10.17	Ótm. Tol.	Tempo	1,99%
10.18	Ótm. Tol.	Tempo	2,20%
10.19	Ótm. Tol.	Tempo	2,47%

"Servidor.Instância"	Status PPRDR	Status PDCRR	*GAP* em relação ao PPRDR
10.20	Ótm. Tol.	Tempo	2,99%
20.11	Ótimo	Tempo	5,33%
20.14	Ótm. Tol.	Tempo	3,44%
20.17	Ótm. Tol.	Tempo	2,31%

A Tabela 7.2 mostra os resultados de oito instâncias que obtiveram o *status* de tempo limite atingido. Porém, apesar destas instâncias não terem encontrado a solução ótima para o PDCRR, observa-se que todas eles obtiveram *status* de ótima solução para o PPRDR e quando calculado o *gap* das soluções, todas elas mostram resultado positivo. Dentre os valores observados, nota-se uma redução de até 5,33% na função objetivo do PDCRR quando comparado com o PPRDR. Isto mostra que mesmo não atingindo o resultado ótimo, o melhor resultado encontrado no tempo limite como solução para o PDCRR já é melhor que a solução ótima para o PPRDR, de onde pode-se concluir que para estas instâncias, também houve certa redução do custo de distribuição dos conteúdos na rede.

Tabela 7.3: Comparação entre função objetivo PPRDR e PDCRR - Resultados Pouco Cnclusivos

"Servidor.Instância"	Status PDCRR	Status PPRDR	*GAP* em relação ao PPRDR
20.16	Tempo	Tempo	2,24%
20.18	Tempo	Tempo	2,16%
20.19	Tempo	Tempo	2,11%
20.20	Tempo	Tempo	2,03%
30.16	Tempo	Tempo	2,67%
30.17	Tempo	Tempo	1,65%
30.18	Tempo	Tempo	2,21%
30.19	Tempo	Tempo	2,48%
30.20	Tempo	Tempo	99,99%
50.6	Ótimo	Tempo	-33,17%
50.8	Ótimo	Tempo	-31,87%
50.9	Ótimo	Tempo	-30,10%
50.10	Ótimo	Tempo	-32,68%
50.11	Tempo	Tempo	-75,26%
50.13	Tempo	Tempo	-23,61%

"Servidor.Instância"	Status PPRDR	Status PDCRR	*GAP* em relação ao PPRDR
50.14	Tempo	Tempo	26,02%
50.15	Tempo	Tempo	-3472315,19%
50.16	Tempo	Tempo	-2,92%
50.17	Tempo	Tempo	6,33%
50.18	Tempo	Tempo	3,49%
50.19	Tempo	Tempo	7,99%
50.20	Tempo	Tempo	3,78%

A Tabela 7.3 mostra os demais resultados das instâncias não contempladas nas Tabelas 7.1 e 7.2. A comparação das soluções fornecidas nestes casos não pode ser considerada conclusiva, visto que nestes caso as soluções fornecidas são soluções intermediárias. Em alguns destes casos, não foi possível resolver nem mesmo a relaxação linear do problema dentro do limite de tempo estabelecido o que torna ainda mais difícil de concluir algo a partir destes resultados, uma vez que os *gaps* calculados nestes casos não fazem sentido.

Desta maneira, não foi possível provar para estas instâncias que a alocação dinâmica trouxe redução no custo de distribuição de conteúdos. Porém, vale lembrar que as soluções encontradas nestes casos não são ótimas e sim soluções intermediárias encontradas durante o processo de busca. Deste modo, caso seja fornecido mais tempo computacional, a redução de custos pode vir a ser encontrada.

Observando a instância 50.15 pela Tabela 7.3, nota-se que possui um *gap* de solução com valor absoluto relativamente alto. Ao analisar o log de solução dado pelo CPLEX, é possível perceber que o *bound* de solução encontrado é negativo, o que indica que não foi possível resolver o sistema linear para este caso, visto que no problema não havia nenhuma variável negativa. Além disso, os *gaps* dados pelo CPLEX ao invés de reduzirem com as iterações, vão aumentando seu percentual. Esta instância será analisada atravéz da heurística nas seções a seguir.

A fim de encontrar soluções de boa qualidade para as instâncias contempladas pela Tabela 7.3, foi proposta uma heurísticas baseada na formulação matemática. Esta heurísca consiste em fixar algumas das variáveis do problema reduzindo o espaço de busca. Os resultados desta heurística serão apresentados mais adiante neste capítulo.

7.2 Instâncias que não contemplam todos os servidores como origem

Quando não há a otimização do espaço em disco nos servidores, pode-se ter desperdício de recurso devido à subutilização de alguns servidores. A manutenção dos servidores acarreta custos às CDNs, por isto, ao utilizar o PDCRR, alguns servidores podem até mesmo não serem utilizados no atendimento das requisições e reduzir ainda mais os custos. A fim de provar este conceito, foram criadas algumas instâncias, com base nas instâncias de classe D, que são instâncias mais próximas à realidade e que apresentam certa escassez de recursos.

Primeiramente foi reduzido em 40% o número de requisições existentes nestas instâncias, porém, nada se pode afirmar com os resultados, visto que ainda utiliza todos os servidores dentro dos períodos de resolução.

A segunda proposta foi a mitigação de alguns servidores como origem durante a criação do modelo. Para isto foram utilizadas instâncias com 20 servidores e, durante a criação das instâncias, considerou-se que apenas 15 dos 20 servidores poderiam ser usados como origem. Para estes casos obteve-se em alguns períodos determinados servidores sem espaço alocado, por não serem necessários para a resolução do problema.

Desta maneira, a alocação dinâmica de espaço em disco nos servidores permitiu a otimização do espaço e evitou a subutilização dos servidores nos períodos.

Tabela 7.4: Distribuição das capacidades nos servidores por período - Servidores 0 a 10

	Identificação do Servidor										
Período	0	1	2	3	4	5	6	7	8	9	10
0	850	0	403	0	351	0	0	0	837	0	0
1	351	1622	403	0	444	393	441	403	377	1232	351
2	351	1999	0	392	0	837	441	403	0	1232	351
3	744	1981	0	0	0	1295	441	813	0	791	761
4	393	1969	0	363	0	1653	441	1223	377	791	761
5	744	2018	0	755	0	1735	366	1223	1209	1117	761
6	1154	1702	0	392	0	1700	366	1222	1209	1486	1124
7	410	1613	0	392	393	2140	366	1222	1969	1486	1124
8	410	2054	0	392	744	2002	0	1588	2410	1135	1124

	Identificação do Servidor										
Período	0	1	2	3	4	5	6	7	8	9	10
9	1154	1766	0	0	754	1954	0	1965	1592	1512	714
10	761	1766	0	0	1147	1954	0	1965	2002	1512	714
11	761	1766	0	0	1147	1954	0	1965	2003	1512	714
12	761	1766	0	0	403	2716	795	1965	1202	1552	351
13	761	1258	0	403	403	2716	351	2070	1202	786	351
14	1202	1261	0	403	795	1550	351	2104	792	786	351
15	1202	1261	0	403	392	1608	351	1254	1169	0	351
16	1202	1261	0	403	392	2018	351	844	1169	0	351
17	851	1612	0	0	392	1625	351	844	377	393	351
18	851	2021	0	0	392	1216	351	844	377	393	351
19	441	2424	0	0	392	776	351	441	0	393	0
20	441	2483	0	0	833	366	351	0	0	393	0
21	441	2483	0	0	833	366	351	0	0	393	0
22	0	2483	0	0	1274	366	351	0	0	393	0
23	0	2043	0	0	1274	366	351	0	440	393	0
24	0	2043	0	0	1274	366	351	0	440	393	0
25	0	2043	0	0	1274	366	351	0	440	393	0
26	0	2043	0	0	1274	366	351	0	440	393	0
27	0	2043	0	0	1274	366	351	0	440	393	0
28	0	2043	0	0	1274	0	351	0	440	393	0
29	0	2043	0	0	1274	0	351	0	440	393	0
30	0	2043	0	0	882	0	351	0	440	393	0

Tabela 7.5: Distribuição das capacidades nos servidores por período - Servidores 11 a 19

	Identificação do Servidor								
Período	11	12	13	14	15	16	17	18	19
0	410	817	363	403	0	0	441	440	0
1	0	0	0	0	0	351	351	0	366
2	0	0	0	351	0	1525	717	0	0
3	0	0	0	351	351	754	1078	0	0
4	0	0	351	760	351	1117	1078	0	0

	Identificação do Servidor								
Período	11	12	13	14	15	16	17	18	19
5	369	0	728	1123	709	714	712	717	0
6	0	0	728	1123	709	714	1075	717	0
7	0	0	377	1123	709	714	1515	776	0
8	0	0	377	1137	709	714	1201	366	0
9	0	0	377	1137	709	1117	1578	366	0
10	0	0	377	1137	1086	714	1138	0	0
11	0	0	377	1137	1086	714	761	377	0
12	0	0	714	1137	1449	351	761	377	0
13	0	0	1091	1137	1072	351	761	821	0
14	0	0	1091	1137	1481	351	1081	1190	0
15	0	0	714	1086	772	351	1081	813	0
16	0	0	714	735	772	351	1081	813	0
17	0	0	363	735	772	351	1133	1605	0
18	0	0	363	735	363	0	1133	1195	0
19	0	0	0	358	773	0	1170	1195	0
20	0	0	0	358	773	0	1170	754	0
21	0	0	0	358	773	0	801	772	0
22	0	0	0	358	773	0	801	772	0
23	0	0	0	358	773	0	801	772	0
24	0	0	0	358	773	0	801	772	0
25	0	0	0	358	773	0	801	772	0
26	0	0	0	0	773	0	801	403	0
27	0	0	0	0	773	0	440	403	0
28	0	0	0	0	773	0	440	0	0
29	0	0	0	0	773	0	440	0	0
30	0	0	0	0	773	0	440	0	0

Por meio das tabelas (7.4) e (7.5) é possível observar uma série de servidores com espaço zero alocado. A Figura 7.2 mostra de maneira gráfica um comparativo entre o percentual de servidores ocupados por período e também o espaço total alocado nos servidores.

Os dados indicam que a utilização do PDCRR pode determinar que estes servidores

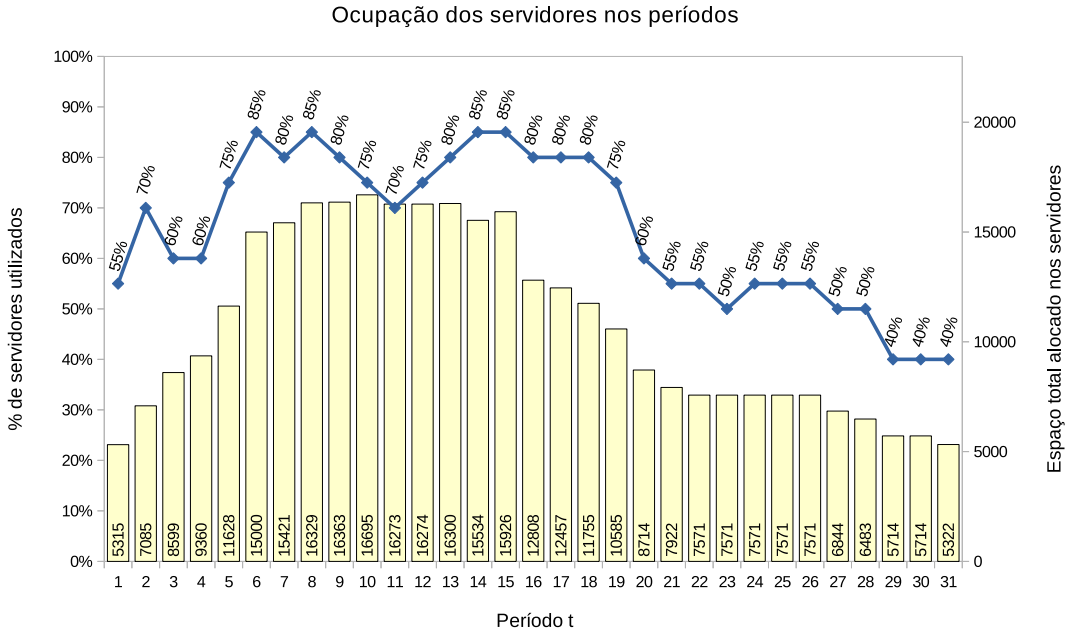

Figura 7.2: Distribuição da ocupação dos servidores nos períodos

sejam desabilitados ou disponibilizados para outro provedor, no caso o uso de servidores compartihados, a fim de reduzir os custos operacionais desta CDN. É possível observar que durante o período de atendimento das requisições desta CDN representada nas tabelas (7.4) e (7.5) houveram mais de 270 lacunas (servidores sem espaço alocado) nas tabelas. Ainda, é possível observar através das Tabelas que os servidores 2, 11, 12 e 19 estiveram sem espaço alocado em mais de 90% dos períodos. O gráfico da Figura 7.2 mostra que em nenhum dos períodos tivemos 100% dos servidores ocupados, havendo casos em que apenas 40% dos servidores disponíveis na rede estavam com pelo menos algum conteúdo alocado.

Vale ressaltar que, para estas intâncias, a solução encontrada tem *status* de melhor solução encontrada dentro do tempo limite estabelecido (10800 segundos), ou seja, mesmo em soluções intermediárias, é possível reduzir o número de servidores da rede sem reduzir a qualidade do atendimento.

7.3 PDCRR com custo de alocação associado

A fim de tornar as instâncias ainda mais próximas da realidade, foi proposta uma modificação na função objetivo do modelo atual de resolução do PDCRR, acrescentando custos para a alocação de espaço nos servidores. Atualmente o modelo não possui um

custo associado à alocação de espaço em disco nos servidores, logo, a utilização ou não de um servidor torna-se indiferente para a função objetivo, mas, de maneira prática, esta alocação gera custos. Além disso, a inclusão destes custos ajuda a diferenciar soluções que antes eram consideradas iguais pelo modelo, privilegiando soluções que utilizam uma fração menor do espaço disponível.

Assim, a função objetivo descrita em pela Equação (5.23), deve ser reescrita como:

$$Min \quad \sum_{i\in R}\sum_{j\in S}\sum_{t\in T} c_{ijt}x_{ijt} + \sum_{i\in R}\sum_{t\in T} q_{it}b_{it} + \sum_{k\in C}\sum_{j\in S}\sum_{l\in S}\sum_{t\in T} h_{kjlt}w_{kjlt} + F\sum_{j\in S}\sum_{t\in T} r_{jt} \quad (7.2)$$

Onde F é a constante que representa o custo de alocação de conteúdo em um servidor, sendo este valor considerado fixo para toda a rede. O uso de técnicas de precificicação dinâmica para o valor de F também pode ser analisado para o caso de formulações *online*, indicando os custos como uma função do estado atual da CDN. Esta técnica não é tratada no escopo deste trabalho, porém pode ser alvo de trabalhos futuros. Outra maneira, seria explorar F como máquinas virtuais com capacidades variadas.

A Tabela (7.6) apresenta os resultados obtidos para função objetivo de algumas instâncias com mais de 30 servidores, considerando F equivalente a 10 unidades, sendo obtido de maneira arbitrária.

Para obter o valor da função objetivo para o PDCRR, foram realizados novos testes computacionais, porém para obter os valores para a função objetivo do PPRDR foram utilizados os resultados previamente obtidos nos primeiros testes realizados (sem custo de alocação associado) e somado a este, a quantidade de espaço alocada em cada servidor multiplicada por F e multiplicada pelo número de períodos.

A coluna *gap* contém a diferença percentual entre as duas abordagens, obtido através do cálculo dado pela equação (7.1).

Tabela 7.6: Comparativo Função Objetivo PPRDR e PDCRR com custo de alocação de capacidade dos servidores

Servidor.Instância	Função Objetivo PPRDR	Função Objetivo PDCRR	*GAP*
30.11	43396530	19309200	55,51%
30.12	54907770	18339800	66,60%
30.13	44393250	18764500	57,73%

Servidor.Instância	Função Objetivo PPRDR	Função Objetivo PDCRR	*GAP*
30.14	43823900	18547700	57,68%
30.15	43459010	18175900	58,18%
30.16	57018850	534271000000	-936907,67%
30.17	56766250	40424300	28,79%
30.18	55961550	38838100	30,60%
30.19	57229350	571263000000	-998099,35%
30.20	472117891400	574082000000	-21,60%
50.11	797020392400	40505900	99,99%
50.12	72226500	36490300	49,48%
50.13	307330472700	415072000000	-35,06%
50.14	87647150	405502000000	-462552,81%
50.15	76961800	453606000000	-589291,10%
50.16	1397829910000	1468210000000	-5,03%
50.17	1423090901550	1374870000000	3,39%
50.18	1616720122450	1615870000000	0,05%
50.19	1732959600250	1625480000000	6,20%
50.20	1615840454600	1591170000000	1,53%

Através dos resultados apresentados na Tabela (7.6), é possível perceber que em alguns casos os resultados são pouco conclusivos, mas são um indicativo forte de que o uso destes custos associados ajuda o processo de busca através de cortes mais robustos, tornando possível a resolução de problemas em instâncias mais próximas ao real.

Em 80% dos casos foi possível observar um *gap* positivo do resultado da função objetivo para o PDCRR em relação ao PPRDR. Isto indica que nesses custos a função objetivo apresentou redução de custo quando utilizada a alocação dinâmica. As instâncias 30.11 e 50.12 possuem solução ótima para o PPRDR. Nestes casos, o ganho foi de aproximadamente 50% em relação ao resultado obtido para a mesma instância no PPRDR.

Comparando as Tabelas (7.6) e (7.1) é possível perceber que para as instâncias de 30 servidores, 70% delas obtiveram melhores resultados de *gap* e para as intâncias com 50 servidores, 30% delas obtiveram *gap* melhor na solução encontrada quando comparado ao problema sem custo de alocação de servidores.

Em todos esses casos foi possível perceber que não foi necessário a utilização de todos os servidores para o atendimento de todas as requisições. Alguns servidores não foram

ocupados durante o horizonte de planejamento. Isto mostra que a alocação dinâmica de espaço nos servidores pode reduzir os custos de replicação de distribuição de conteúdos nos servidores, visto que ela procura alocar de forma ótima todos os recursos necessários para o atendimento ao cliente, sem comprometer a qualidade do serviço.

Ainda, é possível observar um número significativo de instâncias com *gaps* negativos, com uma dispersão consideravelmente alta. As intâncias 30.16, 30.19, 30.20, 50.13, 50.14, 50.15 e 50.16 apresentaram *gaps* negativos, indicando uma perda na solução encontrada para a função objetivo no PDCRR. Além disso, analisando os logs de solução, nota-se que para todas essas instâncias a implementação do CPLEX apresentou o uso de *bounds* negativos. Os *bounds* são os limites ou cortes realizados pelo CPLEX a fim de facilitar a otimização do modelo. Os *bounds* negativos surgem porque o CPLEX não conseguiu resolver a relaxação linear dentro do tempo, visto que o CPLEX trabalha com linhas de execução paralelas. Assim, uma dessas linhas resolve a relaxação linear, procurando o limite dual, e uma das outras busca soluções heurísticamente, encontrando o limite primal. O *gap* fornecido pelo CPLEX utiliza os limites primal e dual para calcular aquela porcentagem. Quando se tem o limite primal, que é mais fácil de ser encontrado e ainda não se tem o limite dual, ele calcula o *gap* mas este não contém informação relevante neste caso. Então, ao invés de seguir reduzindo os *gaps* entre as soluções intermediárias encontradas, segue aumentando. Os valores de *gaps* mostrados pelo CPLEX para estas instâncias são respectivamente: 148.17%, 150.25%, 143.15%, 226,36%, 228.20%, 212.91% e 157.63%.

As instâncias utilizadas neste teste foram aquelas pertencentes às classes C e D, que são classes muito mais próximas da realidade, conforme descrito no Capítulo 6. Para estas instâncias o *status* de solução obtido no PDCRR para todas elas foi de "melhor solução dentro do tempo previsto", estes resultados continuam sendo pouco conclusivos, porém, demonstram que ainda é possível obter resultados com maior *gap* entre o PDCRR e o PPRDR.

Tabela 7.7: Comparativo *GAPs* para a solução da função (7.2) com custo de alocação de capacidade dos servidores - *GAP* 1 e a função (5.23) - *GAP* 2

Servidor.Instância	Status PPRDR	Status PDCRR	*Gap* 1	*Gap* 2
30.11	Ótm. Tol	Tempo	55,51%	2,08%
30.12	Ótimo	Tempo	66,60%	0,85%

30.13	Ótimo	Tempo	57,73%	0,86%
30.14	Ótimo	Tempo	57,68%	1,01%
30.15	Ótimo	Tempo	58,18%	0,88%
50.12	Ótimo	Tempo	49,48%	0,25%

A Tabela 7.7 faz um comparativo entre os *gaps* encontrados entre o PPRDR e PDCRR para os casos em que é acrescentado o custo de alocação nos servidores e os *gaps* encontrados entre as formulações sem custo alocados, mostrados na seção 7.1. Estes resultados destacam as instâncias em que a implementação com custo alocado apresentou *gap* consideravelmente maior que os *gaps* encontrados na primeira solução. Isto demonstra que com a inclusão do custo de alocação na função objetivo torna-se ainda mais lucrativo realizar a alocação dinâmica da capacidade de armazenamento.

Para as instâncias da Tabela 7.7, apesar do *status* encontrado para o PDCRR ser de melhor resultado dentro do tempo disponível, os *status* para o PPRDR foram de Ótimo ou Ótimo dentro da tolerância, ou seja, os melhores resultados encontrados para o PDCRR dentro do tempo limite estabelecido, já se encontram melhores que as soluções para o PPRDR.

Tabela 7.8: Comparativo de *GAPs* da constante F

Cenários	*GAP* 30.11	*GAP* 50.12
F = 0	2,08%	0,25%
F = 1	19,15%	19,77%
F = 5	42,05%	42,99%
F = 10	55,51%	49,48%

Ambas instâncias *30.11* e *50.12* obtiveram o *status* de solução ótima para a função objetivo da formulação FD. Quando acrescentado um custo para a alocação de servidores na rede, observa-se que os *gaps* encontrados em relação à solução para o PPRDR também com custo alocado vão cada vez aumentando mais conforme o custo aumenta. Na Tabela 7.8 é possível perceber que o *gap* de solução aumenta quase 10 vezes mais quando alocado o custo de pelo menos 1 unidade. Este fato indica que a alocação dinâmica dos servidores nas CDNs pode reduzir consideravelmente o custo de solução de entrega de conteúdos sem comprometer a qualidade do serviço e que o custo de alocação de espaço nos servidores pode potencializar a necessidade do uso da alocação dinâmica do espaço total de uma

CDN.

Tabela 7.9: Análise de sensibilidade da constante F para a instância 30.11

Comparativo	Redução no espaço alocado
F=0 x F=1	673808
F=1 x F=5	631
F=5 x F=10	463654

Através da Tabela 7.9 é possível realizar uma análise de sensibilidade da constante F na função objetivo representada na Equação (7.1) para a instância 30.11. Nesta tabela é possível observar uma redução significativa na quantidade de espaço alocado nos servidores quando incluído o custo de alocação destes servidores na função objetivo. O problema busca minimizar o custo da função objetivo, logo, ao tentar minimizar ele reduz a quantidade de espaço alocada nos servidores. Assim, fazendo um comparativo entre as Tabelas 7.8 e 7.9 nota-se que além da redução do espaço alocado no servidor, também aumenta o *gap* entre as soluções dos problemas quando comparado o resultado obtido para o PDCRR com o PPRDR para a mesma instância, mostrando a redução do custo.

Tabela 7.10: Análise de sensibilidade da constante F para a instância 50.12

Comparativo	Redução no espaço alocado
F=0 x F=1	215921
F=1 x F=5	253
F=5 x F=10	-253

A mesma análise realizada para a instância 30.11 pode ser realizada para a instância 50.12. A Tabela 7.10 traz essas informações, porém, neste caso, nota-se uma redução de grau de grandeza de seis quando incluído um custo de alocação na função objetivo, porém, esta diferença pouco muda para os demais cenários. Comparando a Tabela 7.10 com a Tabela 7.8, pode-se estabelecer o mesmo paralelo mostrado acima, visto que os *gaps* para *F=5* e *F=10* são similares.

A Figura 7.3 faz uma comparação gráfica do total de conteúdo alocado em cada servidor conforme variação da constante F. É possível perceber que para a instância 30.11,

ao colocar-se um custo F, já se observa uma redução de 6,74e+5 unidades alocadas nos servidores, e, conforme aumentado este custo, a quantidade de unidades alocadas também pode variar, conforme observado para *F=10*.

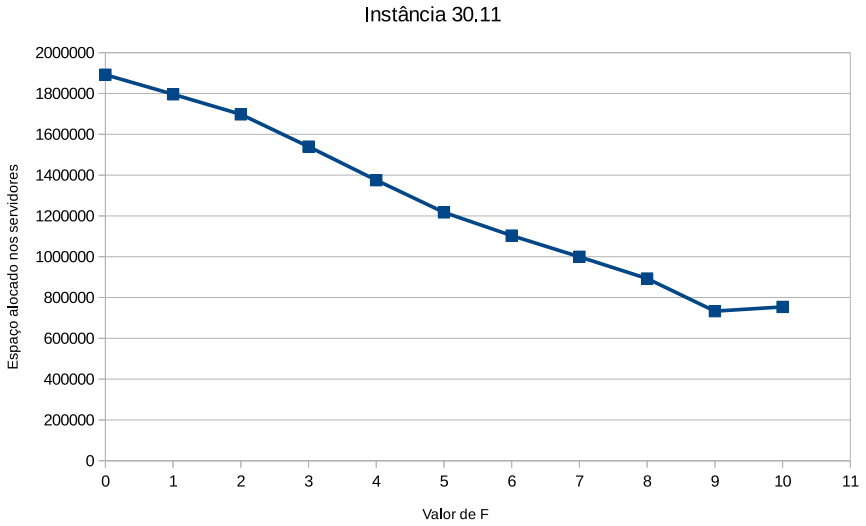

Figura 7.3: Análise Sensibilidade da constante F para instância 30.11

Na Figura 7.4, que faz o mesmo comparativo para a intância 50.12, nota-se que a diferença entre os cenários em que F é diferente de zero é quase nula. Porém, ao comparar-se o cenário em que F é nulo, com qualquer um dos demais cenários, observa-se uma diferença aproximada de 2,16e+5

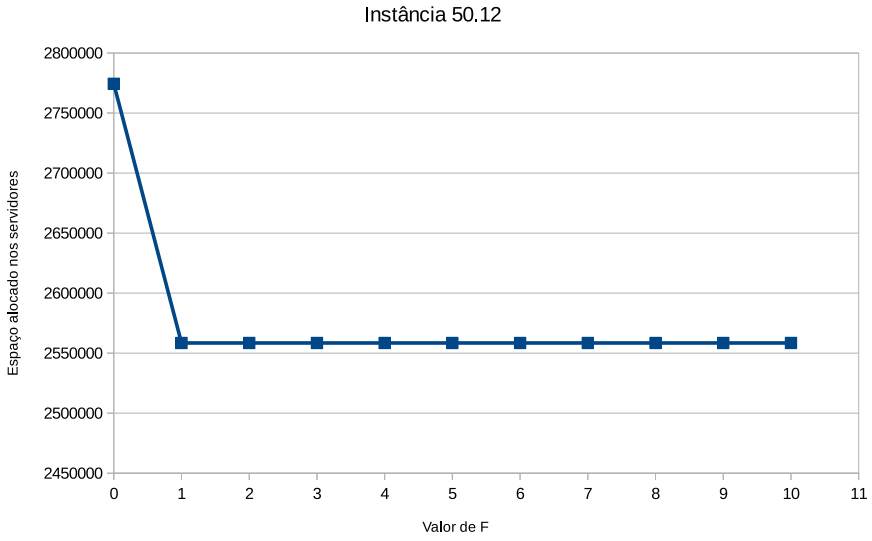

Figura 7.4: Análise de Sensibilidade da constante F para instância 50.11

O valor real de alocação de espaço nos servidores na rede foi estimado através dos dados obtidos de um simulador da Amazon, disponível em [2], em que o custo de 1 *terabyte* é de aproximadamente 100 dólares por mês ou 188 reais, conforme cotação em 2009, disponível em [3]. Atualmente, calculado após aplicação do IPCA [4], este valor seria estimado em 315 reais por *terabyte* utilizado, aproximadamente 31 centavos/*gigabyte*. Como os conteúdos estão representados em *megabytes*, o valor de F na função objetivo seria de 0.03.

7.4 Utilização de heurísticas na solução do PDCRR

Devido aos resultados pouco conclusivos para algumas instâncias obtidos pela formulação proposta para PDCRR, fez-se necessário conceber uma heurística, fazendo algumas modificações na formulação proposta, conforme mostrado no Capítulo 5.4, com o objetivo de reduzir o espaço de busca, reduzindo o consumo de memória e tempo computacional necessários à resolução do problema e permitindo encontrar soluções melhores que as obtidas durante a primeira fase dos testes para as instâncias mencionadas.

Através da aplicação da heurística matemática espera-se encontrar valores ótimos na resolução do modelo. Para as instâncias pertencentes às Classes A e B, que são geralmente utilizadas para testes, há a expectativa de que os resultados sejam similares aos obtidos com os testes da formulação exata. Já para as demais instâncias, espera-se que os custos sejam menores que os valores obtidos para a formulação exata.

Tabela 7.11: Comparativo entre resultado obtidos para formulação exata e heurística matemática para o PPRDR

Instância	Exato	Status	Heurística	Status	Comparativo
30.11	6,70e+006	Ótm. Tol.	6,70e+06	Ótimo	Resultado Similar
30.12	6,39e+006	Ótimo	6,39e+06	Ótimo	Resultado Similar
30.13	6,61e+006	Ótimo	6,61e+06	Ótimo	Resultado Similar
30.14	6,53e+006	Ótimo	6,53e+06	Ótimo	Resultado Similar
30.15	6,52e+006	Ótimo	6,52e+06	Ótimo	Resultado Similar
30.16	2,66e+007	Tempo	4,42e+08	Tempo	FO Maior
30.17	2,59e+007	Tempo	6,23e+08	Tempo	FO Maior
30.18	2,53e+007	Tempo	1,13e+09	Tempo	FO Maior
30.19	2,67e+007	Tempo	6,15e+08	Tempo	FO Maior

Instância	Exato	Status	Heurística	Status	Comparativo
30.20	4,72e+011	Tempo	6,15e+08	Tempo	FO Menor
50.11	7,97e+11	Tempo	1,13e+07	Ótm. Tol.	FO Menor
50.12	1,09e+07	Ótimo	1,09e+07	Ótimo	Resultado Similar
50.13	3,07e+11	Tempo	1,15e+07	Tempo	FO Menor
50.14	1,49e+07	Tempo	1,18e+07	Tempo	FO Menor
50.15	1,51e+07	Tempo	1,12e+07	Ótm. Tol.	FO Menor
50.16	1,40e+12	Tempo	0	Inviável	Inviável
50.17	1,42e+12	Tempo	1,17e+09	Tempo	FO Menor
50.18	1,62e+12	Tempo	0	Inviável	Inviável
50.19	1,73e+12	Tempo	2,02e+09	Tempo	FO Menor
50.20	1,62e+12	Tempo	0	Inviável	Inviável

A Tabela 7.11 compara os resultados encontrados com a heurística matemática para o PPRDR com os resultados da função objetivo para a formulação exata. A última coluna mostra um resumo desses resultados. Observa-se que os resultados em que foi possível encontrar uma solução ótima na formulação exata, mostraram resultados similares na heurística matemática, mostrando coerência entre o modelo exato e heurístico matemático.

Porém, para as instâncias de Classe D, que são mais próximas à realidade e com 30 servidores, a heurística matemática não foi capaz de encontrar um resultado ótimo e para as intâncias de 30.16 a 30.19, os resultados encontrados pela heurística foram ainda maiores que o resultado obtido para a formulação exata. A instância 30.20 obteve redução da função objetivo com a utilização da heurística, porém este foi ainda o melhor resultado encontrado dentro do período de tempo estabelecido.

No caso das instâncias com 50 servidores, a instância 50.12 que foi a única instância que obteve solução ótima na formulação exata, também obteve um resultado similar com a utilização da heurística. As instâncias 50.16, 50.18 e 50.20 não foram capazes de serem resolvidas pela heurística proposta. As demais instâncias foram resolvidas, porém continuaram com *status* de melhor solução dentro do tempo limite, mas os resultados encontrados pela heurística para a função objetivo foram maiores que os obtidos pela formulação exata.

Tabela 7.12: Comparativo entre resultado obtidos para formulação exata e heurística matemática para o PDCRR

Instância	Exato	Status	Heurística matemática	Status	Comparativo
30.11	6,59e+06	Tempo	1,93e+07	Ótm. Tol.	FO Maior
30.12	6,34e+06	Ótimo	1,83e+07	Ótm. Tol.	FO Maior
30.13	6,56e+06	Ótimo	1,88e+07	Ótm. Tol.	FO Maior
30.14	6,46e+06	Ótimo	1,85e+07	Ótm. Tol.	FO Maior
30.15	6,46e+06	Ótimo	1,82e+07	Ótm. Tol.	FO Maior
30.16	4,07e+11	Tempo	4,56e+08	Tempo	FO Menor
30.17	4,20e+11	Tempo	0	Inviável	Inviável
30.18	4,86e+11	Tempo	1,14e+09	Tempo	FO Menor
30.19	5,09e+11	Tempo	0	Inviável	Inviável
30.20	4,10e+11	Tempo	6,13e+08	Tempo	FO Menor
50.11	6,59e+11	Tempo	3,14e+07	Tempo	FO Menor
50.12	1,09e+07	Tempo	2,99e+07	Tempo	FO Maior
50.13	3,48e+11	Tempo	3,22e+07	Tempo	FO Menor
50.14	1,10e+07	Tempo	3,34e+07	Tempo	FO Maior
50.15	3,71e+11	Tempo	3,13e+07	Tempo	FO Menor
50.16	1,34e+12	Tempo	0	Inviável	Inviável
50.17	1,18e+12	Tempo	0	Inviável	Inviável
50.18	1,49e+12	Tempo	0	Inviável	Inviável
50.19	1,48e+12	Tempo	0	Inviável	Inviável
50.20	1,42e+12	Tempo	0	Inviável	Inviável

A Tabela 7.12 faz uma comparação entre os resultados do PDCRR Exatos e Heurístico Matemáticos, incluindo na última coluna um resumo desta comparação. Os resultados para o PDCRR mostram um cenário diferente daquele encontrado para o PPRDR. Neste caso, as instâncias 30.12, 30.13, 30.14 e 30.15 que obtiveram *status* ótimo na formulação exata, para o caso heurístico, obtiveram *status* de ótimo na tolerância, porém, apresentaram uma piora no resultado da função objetivo. O PDCRR também apresentou instâncias com inviabilidade nas soluções já para aquelas com 30 servidores, são elas: 30.17, 30.19. As instâncias 30.16 e 30.20 apresentaram redução no custo da função objetivo. Quando comparado o *status*, observa-se que na maioria das instâncias com *status* Tempo na solução exata, obtiveram redução na função objetivo da heurística matemática.

Para as instâncias de 50 servidores, em que todas elas obtiveram o *status* 'Tempo' em ambas as formulações, podemos dizer que todas elas tiveram redução no custo da função objetivo ou pelo menos resultados similares, no caso da 50.12 e 50.14 em que não aumentaram sua ordem de grandeza. Porém, para todas as instâncias pertencentes à Classe D, que são as instâncias assimétricas e com maior escassez de recurso, não foi possível obter solução viável através da utilização da heurística matemática proposta.

Tabela 7.13: Comparativo entre resultado obtidos para formulação exata e heurística matemática para o PDCRR - Segunda Tentativa

Instância	Exato	Status	Heurística matemática	Status	Comparativo
30.17	4,20e+11	Tempo	7,46e+08	Tempo	FO Menor
30.19	5,09e+11	Tempo	5,72e+09	Tempo	FO Menor
50.17	1,18e+12	Tempo	1,04e+10	Tempo	FO Menor

Devido a identificação de inviabilidades na heurística, foi restabelecido um limite com maior folga para as variáveis de *backlog* b_{it}, definindo que o limite máximo de períodos em que a requisição pode ser atrasada é o triplo, e não mais o dobro, da soma do tempo mínimo de entrega com o tempo de chegada do conteúdo. O teste foi aplicado na formulação do PPRDR, e somente a instância 50.16 apresentou uma solução viável, no valor de 2,44e+09, com *status* de melhor solução no tempo, porém com resultado da função objetivo menor que o encontrado na formulação exata. Na Tabela 7.13 é possível observar que as instâncias 30.17, 30.19 e 50.17 foram capazes de serem resolvidas após esta alteração para o limite da variável de *backlog*. Porém, a inviabilidade ainda se mantém para as demais instâncias mostradas na Tabela 7.12.

Para encontrar a viabilidade na solução das demais instâncias, foram realizados novos testes, estabelecendo-se o limite do backlog para de 3,5 vezes a soma do tempo mínimo de entrega com o tempo de chegada do conteúdo. As Tabelas 7.14 e 7.15 mostram as soluções obtidas nesta terceira tentativa.

Tabela 7.14: Comparativo entre resultado obtidos para formulação exata e heurística matemática para o PPRDR- Terceira Tentativa

Instância	Exato	Status	Heurística matemática	Status	Comparativo
50.18	1,62e+12	Tempo	2,76e+09	Tempo	FO Menor
50.20	1,62e+12	Tempo	1,39e+09	Tempo	FO Menor

Tabela 7.15: Comparativo entre resultado obtidos para formulação exata e heurística matemática para o PDCRR- Terceira Tentativa

Instância	Exato	Status	Heurística matemática	Status	Comparativo
50.16	1,34e+12	Tempo	2,19e+12	Tempo	FO Maior
50.18	1,49e+12	Tempo	3,54e+10	Tempo	FO Menor
50.19	1,48e+12	Tempo	2,42e+12	Tempo	FO Maior
50.20	1,42e+12	Tempo	3,12e+10	Tempo	FO Menor

Conforme pode ser observado nas Tabelas 7.14 e 7.15, todas as demais instâncias foram possíveis de ser resolvidas através da heurística matemática. Apesar de algumas instâncias ainda apresentarem, a função objetivo maior para a heurística matemática, o resultado ainda se mostra na mesma ordem de grandeza. Comparando os resultados das instâncias

Tabela 7.16: *Gaps* entre resultado obtidos nas heurísticas matemáticas para o PPRDR e PDCRR

Instância	PPRDR	Status	PDCRR	Status	*Gaps*
30.11	6,70e+06	Ótm. Tol.	1,93e+07	Ótm. Tol.	188,33%
30.12	6,39e+06	Ótimo	1,83e+07	Ótm. Tol.	186,84%
30.13	6,61e+06	Ótimo	1,88e+07	Ótm. Tol.	183,76%
30.14	6,53e+06	Ótimo	1,85e+07	Ótm. Tol.	184,11%
30.15	6,52e+06	Ótimo	1,82e+07	Ótm. Tol.	178,96%
30.16	4,42e+08	Tempo	4,56e+08	Tempo	3,08%
30.17	6,23e+08	Tempo	7,46e+08	Tempo	19,65%
30.18	1,13e+09	Tempo	1,14e+09	Tempo	1,18%

Instância	PPRDR	Status	PDCRR	Status	*Gaps*
30.19	6,15e+08	Tempo	5,72e+09	Tempo	830,77%
30.20	6,15e+08	Tempo	6,13e+08	Tempo	-0,39%
50.11	1,13e+07	Tempo	3,14e+07	Tempo	177,67%
50.12	1,09e+07	Ótimo	2,99e+07	Tempo	173,74%
50.13	1,15e+07	Tempo	3,22e+07	Tempo	180,83%
50.14	1,18e+07	Tempo	3,34e+07	Tempo	181,84%
50.15	1,12e+07	Tempo	3,13e+07	Tempo	179,25%
50.16	2,44e+09	Tempo	2,19e+12	Tempo	89708,16%
50.17	1,17e+09	Tempo	1,04e+10	Tempo	785,36%
50.18	2,76e+09	Tempo	3,54e+10	Tempo	1184,07%
50.19	2,02e+09	Tempo	2,42e+12	Tempo	119646,34%
50.20	1,39e+09	Tempo	3,16e+10	Tempo	2139,83%

Tabela 7.16 compara os resultados obtidos na heurística matemática para o PPRDR e o PDCRR e calcula o *gap* encontrado entre as soluções para ambas formulações, conforme Equação 7.1. Os resultados de *gaps* apresentam que a solução da função objetivo para o PDCRR foram menores que para o PPRDR, indicando redução de custo. Porém, ainda assim os *status* de solução das instâncias de Classe C com 50 servidores e Classe D com 30 e 50 servidores continuam apresentando *status* de melhor solução encontrada dentro do tempo limite estabelecido. Esses resultados continuam sendo pouco conclusivos mesmo com a aplicação da heurística.

Esses resultados mostram que com o aumento do número de servidores, a aplicação desta heurística não é o melhor método de resolução. A utilização de um limite de backlog superior a 4 vezes a soma do tempo mínimo de entrega com o tempo de chegada do conteúdo já aproximaria significativamente a heurística à formulação matemática, e possivelmente faria a utilização de servidores apenas para alocação de pequenas unidades de conteúdo, podendo causar a subutilização destes, não respeitando ao objetivo da formulação inicial.

Capítulo 8

Conclusões e Trabalhos Futuros

8.1 Conclusões

Conforme objetivo geral do presente trabalho, foi apresentado um novo problema nomeado de PDCRR, presente em CDNs, que tem por objetivo a integração do PACA e do PPRDR para resolver o problema de otimização do espaço de armazenamento além da distribuição de conteúdos, réplicas e requisições. Além disso, também foi apresentado um modelo de resolução do problema que faz a associação entre variáveis presentes nos modelos de resolução dos problemas estudados por Uderman [32] e Neves [28], e criado um novo modelo de resolução único e integrado.

Como objetivo específico para obtenção do resultado final proposto foi realizada uma análise das formulações matemáticas para o PACA e o PPRDR a fim de comparar as características destes problemas. A análise dessas duas formulações permitiu cumprir o objetivo específico de criação de uma formulação matemática para o PDCRR, capaz de tratar de maneira integrada os dois problemas já existentes (PACA e PPRDR). Além de atender às características dos dois problemas já existentes, essa nova formulação ainda mostrou-se capaz de resolver de maneira dinâmica a questão da alocação de capacidade nos servidores e também o Problema de Posicionamento dos Servidores (PPS).

Para encontrar a solução do PDCRR proposto, foram realizados testes em instâncias com até 50 servidores, aplicando a formulação matemática. Além disso, foram identificados alguns resultados pouco conclusivos com a aplicação da formulação exata, para isto foi criada uma heurística matemática afim de facilitar a resolução do problema.

A aplicação da formulação exata mostrou ganho de redução de custo de até 4,55%, obtendo resultado ótimo para quase todas as instâncias pertencentes às Classes A, B e

C. Os resultados apresentados pelas instâncias mais simples, definidas principalemente para testes (pertencentes às classes A e B), mostraram pouca ou nenhuma redução no custo da função objetivo comparado do PPRDR. A grande maioria das instâncias com 10 e 20 servidores foram resolvidas otimamente pela formulação exata, sendo que aquelas pertencentes às Classes C e D, que são mais próximas da realidade de funcionamento das CDNs, obtiveram redução no custo da função objetivo.

Algumas instâncias apresentaram resultados inconclusivos quando comparado os resultados da função objetivo, nestas instâncias foi observado que o CPLEX não foi capaz de resolver o sistema linear. Para estes casos, foi aplicada a heurística matemática de resolução do modelo proposto tanto para o PPRDR e o PDCRR, a qual apresentou resultados coerentes aos da formulação exata para a maioria das instâncias com resultados ótimos e pertencentes às Classes A e B. A heurístuca ainda não foi capaz de resolver otimamente todas as instãncias pertencentes às Classes C e D, mostrando inviabilidade para alguns casos. Após alguns ajustes realizados no limite das varáveis de *backlog*, tornou-se possível a resolução de todas as instâncias. Em todos os casos, os resultados foram comparáveis àqueles obtidos pela formulação exata, ou ainda obtiveram redução na função objetivo quando aplicada a heurística matemática, porém, observou-se que, com o aumento do número de servidores, a utilização desta heurística pode não ser o melhor método de resolução, podendo em alguns casos, ativar servidores que seriam pouco utilizados.

Através da análise de ocupação dos servidores período a período, foi possível identificar que alguns servidores não tiveram espaço em disco alocado durante alguns momentos da implementação da formulação. Logo, este fato indica que a utilização do PDCRR pode determinar que alguns servidores sejam desabilitados ou disponibilizados para outros provedores, reduzindo o custo operacional e sem afetar a qualidade de serviço e confiabilidade.

Foi identificado a possibilidade de inclusão do custo de alocação de espaço nos servidores no problema, visto que, na realidade, a manutenção dos servidores nas CDNs gera custos à rede. Desta maneira, foi proposta ainda uma análise de sensibilidade da inclusão deste custo na função objetivo. Através desta análise foi identificado que no caso de considerar-se a inclusão do custo de armazenamento de dados nos servidores pode potencializar a necessidade do uso da alocação dinâmica do espaço total de uma CDN.

O modelo utilizado foi capaz de resolver de forma integrada o PACA e o PPRDR, obtendo sucesso significativo para grande parte das instâncias utilizadas. O modelo foi capaz de provar que a alocação dinâmica de espaço nos servidores foi capaz de reduzir o

custo sem prejudicar o atendimento dos padrões de qualidade exigidos.

8.2 Trabalhos Futuros

O uso de modelos exatos para resolução da formulação não foi capaz de encontrar a solução ótima e a aplicação de uma heurística matemática não foi suficiente para aproximar estas soluções. Assim, uma das frentes de trabalho é explorar novas ferramentas de otimização, bem como o uso de meta-heurísticas para encontrar o valor mais próximo do ótimo. O uso de algoritmos genéticos poderia facilitar a busca por resultados melhores, no qual os resultados são apresentados como um conjunto de soluções. O estudo de outras técnicas de resolução como local-branching ou geração de colunas também deve ser analisado. Outra opção para encontrar resultados ótimos é trabalhar no reescalonamento das instâncias, visto a limitação computacional que a formulação pode atingir quando utilizados algarismos com grande quantidade de caracteres.

Ainda é possível atribuir variáveis de balanceamento de cargas a fim de determinar o número máximo de requisições por unidade de tempo, desta maneira, identificando o máximo carregamento que poderá ser atribuído a um servidor, prevenindo o congestionamento nos servidores da rede.

Uma análise de sensibilidade mais detalhada também pode ser realizada a fim de identificar os pontos críticos para o custo de alocação nos servidores que poderia afetar positiva ou negativamente a resolução do modelo. O custo de alocação em servidores pode variar conforme localização, modelo, entre outras características. Diante disso faz-se interessante uma análise do uso de precificação dinâmica desta constante.

Outro ponto que poderá ser abordado é o estudo de uma política de descarte de requisições, a fim de tornar o problema capaz de encontrar a solução ótima para as demais instâncias com 30 e 50 servidores.

A reimplementação da metodologia proposta por Uderman, para que os resultados da integração total feita neste trabalho e da integração parcial proposta por ele possam ser comparados de forma coerente.

A mesma técnica utilizada para a distribuição de capacidades em disco neste trabalho, ou seja, inclusão de variáveis para representar estas capacidades, também pode ser usada para determinar as demais capacidades dos servidores. Isto leva à concepção de um modelo mais geral de planejamento dentro do ambiente das CDNs, onde todas as

capacidades podem ser otimizadas. No modelo proposto por Neves e Uderman, as capacidades analisadas foram espaço em disco e banda nos servidores. Contudo, existem na literatura modelos que consideram outros limites de capacidade como processamento e número máximo de requisições. Estes outros limites de capacidade podem ser otimizados com a técnica utilizada neste trabalho, o que leva ao raciocínio de que existe um modo uniforme para planejar de maneira ótima estas capacidades. Este estudo também será alvo de um estudo futuro.

Referências

[1] AKAMAI. World Wide Web, www.akamai.com. Acessado em 03/2014.

[2] Amazon Web Services Simple Monthlky Calculator . World Wide Web, http://calculator.s3.amazonaws.com/index.html. 05/2017.

[3] Associação Comercial Industrial de Serviços Novo Hamburgo - Cotação Dolar. World Wide Web, http://www.acinh.com.br/servicos/cotacao-dolar. 07/2017.

[4] Inflação | IPCA. World Wide Web, https://br.advfn.com/indicadores/ipca. 09/2017.

[5] Olimpíadas terão forte impacto sobre a Internet - Jornal O Globo, por Carlos Alberto Teixeira. World Wide Web, https://oglobo.globo.com/sociedade/tecnologia/olimpiadas-terao-forte-impacto-sobre-internet-5599261. 07/2017.

[6] Support and Services | BlueCoat . World Wide Web, https://www.bluecoat.com/support-services. 10/2017.

[7] Virtualização VMware. World Wide Web, http://www.vmware.com/br/solutions/virtualization.html. 04/2017.

[8] What's Network Virtualization? World Wide Web, http://searchservervirtualization.techtarget.com/definition/network-virtualization. 04/2017.

[9] What's Server Virtualization? World Wide Web, http://searchservervirtualization.techtarget.com/definition/server-virtualization. 04/2017.

[10] LABIC. World Wide Web,http://labic.ic.uff.br/, 2005. 02/2009.

[11] IBM ILOG CPLEX Optimization Studio V12.5.1, user's manual, 2013.

[12] Aioffi, W., Mateus, G., Almeida, J., Loureiro, A. Dynamic content distribution for mobile enterprise networks. *IEEE Journal on Selected Areas in Communications 23*, 10 (2005).

[13] Bektas, Tolga, O. O., Ouveysi, I. Design cost-effective content distribution networks. *Computers and Operations Research 34*, 8 (2007), 2436–2449.

[14] Bjorkqvist, M., Chen, L. Y., Zhang, X. Minimizing retrieval cost of multi-layer content distribution systems. *Communications (ICC), 2011 IEEE International Conference on* (2011), 1–6.

[15] Cassimiri, A. Virtualização: da teoria a soluções. 173–207.

[16] Chen, F., Guo, K., Lin, J., Porta, T. L. Intra-cloud lightning: Building cdns in the cloud. *IEEE Infocom* (Março 2012), 433–441.

[17] Coppens, J., W. T. D. T. F. D. B. . D. P. Design and performance of a self–organizing adaptative content distribution network. *IEEE/IFIP Network Operations and Management Symposium NOMS* (2006), 534–545.

[18] Gerhardt, R. Análise de redundância em formulações matemáticas para um problema de gerenciamento ligado a redes de comunicação do tipo cdn. *Trabalho de Conclusão de Curso - UFF* (2014).

[19] Gerhardt, R., Neves, T., Albuquerque, C. Análise de redundância em uma formulação matemática para o problema de posicionamento de réplicas e distribuição de requisições em redes de distribuição de conteúdos. *SBPO - Simpósio Brasileiro de Pesquisa Operacional 8* (Agosto 2015).

[20] Hu, H., Wen, Y. Joint content replication and request routing for social video distribution over cloud cdn: A community clustering method. *IEEE Transactions on Circuits and Systems for Video Technology 26*, 7 (Julho 2016), 1320–1333.

[21] Ibrahim, A. S., Hamlyn-Harris, J., Grundy, J. Emerging security challenges of cloud virtual infrastructure.

[22] Laoutaris, N., Zissimopoulos, V., Stavrakakis, I. On the optimization of storage capacity allocation for content distribution. 409–428.

[23] Leighton, F. T., Lewin, D. Global hosting system, Agosto 2000. US Patent:US006108703.

[24] Li, B., mordecai J. Golin, Italiano, G. F., Deng, X. On the optimal placement of web proxies inthe internet. *INFOCOM 3* (1999), 1282–1290.

[25] Li, W., Chan, E., Wang, Y., Chen, D., Sanglu. Cache placement optimization in hierarchical networks: Analysis and performance evaluation. *International Conference on Research in Networking* (2008), 385–396.

[26] Li, W. e. a. *Analysis and performance study for coordinated hierarchical cache placement strategies*, 1 ed. Elsevier, 2010.

[27] Lin, C. F., Leu, M. C., Chang, C. W., Yuan, S. M. The study and methods for cloud based cdn. *Cyber-Enabled Distributed Computing and Knowledge Discovery (Cyber C)* (2011), 469–475.

[28] Neves, T. Redes de distribuição de conteúdos: Abordagens exatas, heurísticas e híbridas. *Tese de Doutorado, Universidade Federal Fluminense.*

[29] Neves, T. Redes de distribuição de conteúdos: Abordagens exatas e heurísticas. Relatório Técnico, UFF, 2011.

[30] Skapinetz, K. *Virtualisation as a Blackhat Tool.* 2007.

[31] Tang, X., Xu, J. On replica placement for qos-aware content distribution. Em *Proc. of INFOCON2004* (2004), p. 806–815.

[32] Uderman, F., Neves, T., Albuquerque, C. Optimizing server storage capacity on content distribution networks. Em *Anais do Simpósio Brasileiro de Redes de Computadores - SBRC2011.* (2011).

[33] Wu, J., kaliappa Ravindran. optimization algorithms for proxy server placement in content distribution networks. *Integrated Network Management-Workshops, 2009. IM'09. IFIP/IEEE International Symposium on* (2009), 193–198.

[34] Wu, Y., Wu, C., Li, B., Qiu, X., Lau, F. Cloud media: When cloud on demand meets video on demand. *Conf. Distrib. Comput. Syst. (ICDCS)* (Junho 2011), 268–277.

[35] Yang, M., Fei., Z. A model for reclica placement in content distribution networks for multimedia applications. *Communications 1* (2003), 557–561.