

Replicação e Distribuição *Online* em Redes de Distribuição de Conteúdos

**Tiago Araújo Neves, Lúcia M. A. Drummond,
Luiz Satoru Ochi, Célio Albuquerque**

Instituto de Computação – Universidade Federal Fluminense
R: Passo da Pátria 156 - Bloco E - 3º andar, São Domingos, Niterói - RJ - CEP: 24210-240
{tneves, lucia, satoru, celio}@ic.uff.br

Eduardo Uchoa

Departamento de Engenharia de Produção – Universidade Federal Fluminense
R: Passo da Pátria 156 - Bloco E - 4º andar, São Domingos, Niterói - RJ - CEP: 24210-240
uchoa@producao.uff.br

RESUMO

Uma Rede de Distribuição de Conteúdos (RDC) é um sistema de computadores interligados que colaboram para fornecer conteúdos para clientes. Ela mantém réplicas de cada conteúdo em seus servidores, objetivando reduzir atrasos, a carga nos servidores e o congestionamento da rede, melhorando a qualidade do serviço. Em arquiteturas de RDC tradicionais, as requisições são recebidas por um servidor central e redirecionadas para o servidor mais próximo do cliente. Entretanto, devido aos custos de replicação, não é razoável replicar os conteúdos em toda a rede. Neste trabalho são apresentadas abordagens exatas e heurísticas para resolver um problema de gerenciamento das RDC, o Problema de Posicionamento de Réplicas e Distribuição de Requisições (PPRDR) *online*. Este problema consiste em decidir quais réplicas serão colocadas em cada servidor e quais servidores irão atender as requisições. O objetivo final é minimizar o tráfego na rede sem violar as restrições de qualidade de serviço (QoS).

PALAVRAS CHAVE. Redes de distribuição de conteúdos. Método híbrido. Problema de Posicionamento de réplicas e Distribuição de Requisições. Área de classificação principal (escolher uma na janela de áreas)

ABSTRACT

A content distribution network (CDN) is a system of interconnected computers that cooperate to deliver content to clients. It maintains replicas of each content in its servers, with the goal of reducing delays, server load and network congestion, therefore improving the quality of the service. In traditional CDN architectures, requests are received by a central server and then redirected to the server that is closer to the client. However, because of the costs involved in replication process, it is not reasonable to replicate the contents over the entire network. In this work, exact and heuristic approaches are presented to solve a problem that appears in CDN management, called the Request Distribution and Replica Placement Problem online. This problem consists of deciding which replicas will be placed in each server and which servers will handle the requests. The overall objective is to minimize the traffic in the network without violating QoS constraints.

KEYWORDS. Content Distribution Network. Hybrid Method. Request Distribution and Replica Placement Problem. Main area (choose between those in the areas window).

1. Introdução

Com o crescimento da internet, a demanda por conteúdos diversos aumentou consideravelmente em todo o mundo. Alguns destes conteúdos, em especial os conteúdos de multimídia, necessitam algum tipo de Qualidade de Serviço (Quality of Service - QoS) como atraso máximo e banda mínima para que as expectativas dos clientes possam ser satisfeitas. Devido a isso, muitos estudos têm sido feitos na tentativa de encontrar maneiras de servir melhor a crescente demanda por conteúdos com restrições de QoS [Tenzakthi 2004] [Zhou 2007].

Um modo de servir este tipo de conteúdo de maneira eficiente é o uso de Redes de Distribuição de Conteúdos (RDC) [Bakiras 2005][Tenzakthi 2004][Zhou 2007]. RDCs são tipicamente redes sobrepostas usadas para posicionar réplicas dos conteúdos nas proximidades dos clientes, melhorando assim a qualidade percebida e reduzindo o atraso, a carga nos servidores e o congestionamento da rede.

Em arquiteturas de RDC tradicionais, as requisições dos clientes são recebidas por um servidor central e então, cada requisição é redirecionada para um outro servidor, capaz de atender a requisição e que se encontre nas proximidades do cliente. Na realidade, as requisições dos clientes nem sempre serão atendidas pelo servidor mais próximo. Em muitos casos, é mais vantajoso usar um servidor que se encontra um pouco mais distante, mas que esteja menos sobrecarregado do que o que se encontra mais próximo ao cliente.

Empresas como Akamai, [Akamai 2008], Level 3 [Level 3 2009] e Mirror Image [Mirror Image 2009] especializaram-se em prover arquiteturas de RDC para empresas, como Adobe, Aude Ag e Fox Interactive, que querem, ou necessitam distribuir seus conteúdos.

Existem vários problemas de otimização dentro das arquiteturas de RDC. Entre eles estão o Problema de Localização de Servidores, o Problema de Replicação e o Problema de Posicionamento de Réplicas (PPR). O primeiro consiste em encontrar os melhores locais dentro da rede real para posicionar os servidores que irão compor a rede sobreposta da RDC [Bektas 2007]. O segundo consiste em decidir quais conteúdos serão replicados [Zhou 2007]. E o último consiste em encontrar os melhores locais (servidores) dentro da rede sobreposta para colocar os conteúdos replicados [Aioffi 2005][Zhou 2007].

O problema que é abordado neste trabalho é uma variante *online* do PPR e consiste em encontrar os melhores locais para posicionar as réplicas dos conteúdos a fim de que as exigências de QoS das requisições não sejam violadas e o tráfego dentro da rede seja minimizado ao longo do tempo.

O restante deste trabalho está organizado da seguinte maneira: na seção 2 é feita uma descrição detalhada do problema. Alguns trabalhos relacionados são discutidos na seção 3. Na seção 4 é apresentada uma formulação matemática para o problema. Na seção 5 é descrita uma heurística para o problema. Os resultados computacionais e as conclusões são apresentadas respectivamente nas seções 6 e 7.

2. Problema de Posicionamento de Réplicas e Distribuição de Requisições Online

O Problema de Posicionamento de Réplicas consiste em encontrar os melhores servidores para armazenar as réplicas dentro da RDC de modo que toda a demanda seja atendida. Portanto, para resolver este problema, é necessário conhecer as posições dos servidores, bem como sua capacidade de banda e armazenamento.

Como mencionado anteriormente, encontrar o posicionamento ótimo para os servidores também é um problema de otimização. Por simplificação, neste trabalho, considerou-se que este problema já está resolvido, e a solução deste é dada como entrada para o problema tratado.

O problema tratado neste trabalho é uma variante do PPR chamada Problema de Posicionamento de Réplicas e Distribuição de Requisições (PPRDR) dinâmico e *online*, o qual consiste em encontrar o melhor posicionamento para as réplicas dentro da rede sobreposta e redistribuir as requisições entre os servidores ao longo do tempo, com o objetivo de reduzir a carga da rede sem violar as restrições de QoS das requisições.

Requisições com QoS são tipicamente encontradas em ambientes onde existem usuários que pagam por algum tipo de vantagem, como uma melhor qualidade no conteúdo ou uma melhor

velocidade de acesso. Neste trabalho, os cenários foram construídos considerando a possibilidade da existência de tratamento diferenciado de clientes pela arquitetura da RDC. Portanto, o uso de restrições de QoS surge como opção natural para modelar este tratamento diferenciado. Estão sendo tratadas, neste trabalho, duas exigências de QoS: banda mínima e atraso máximo.

Como os arquivos exigidos pelas requisições, no problema abordado neste trabalho, são muito grandes, frequentemente estas requisições se prolongam por vários períodos de tempo baixando uma fração do conteúdo exigido a cada período. Determinar o tamanho da fração é um complicador do problema, pois, transmitir frações pequenas dos conteúdos exigidos fará com que as requisições se estendam por mais períodos de tempo. Este prolongamento exagerado do atendimento das requisições pode acarretar uma sobrecarga no sistema devido à alta concentração de requisições em um mesmo período. Como os conteúdos podem ser segmentados em fragmentos menores, com o objetivo de reduzir o custo operacional e aumentar a velocidade de atendimento, é permitido que múltiplos servidores se encarreguem do atendimento de uma mesma requisição em um mesmo período desde que sejam respeitadas as limitações dos servidores e dos clientes.

Para atender uma requisição no PPRDR alguns fatores precisam ser considerados. Uma requisição pode ser atendida, total ou parcialmente, por um servidor somente se este servidor possuir uma réplica do conteúdo desejado pela requisição. As restrições de QoS da requisição também precisam ser averiguadas. Se as restrições são violadas, redirecionar esta requisição irá produzir uma solução inviável. Em alguns casos, para otimizar a distribuição de requisições, é necessário, primeiramente, alterar o posicionamento das réplicas.

O posicionamento das réplicas está sujeito a fatores relacionados com os servidores como, por exemplo, capacidade de armazenamento e distâncias entre os mesmos. No início, os conteúdos se encontram em seus servidores de origem, podendo um ou mais conteúdos ter o mesmo servidor de origem, e só depois, eles são distribuídos pela rede. Devido à possibilidade da existência de custos associados com o transporte das réplicas destes conteúdos, a relação custo-benefício entre a redução de carga na rede e o custo de transporte das réplicas deve ser analisada antes da distribuição. Estes fatores tornam o posicionamento das réplicas um problema não trivial, no qual uma decisão precipitada pode acarretar grandes custos e/ou inviabilidades nos requisitos de QoS.

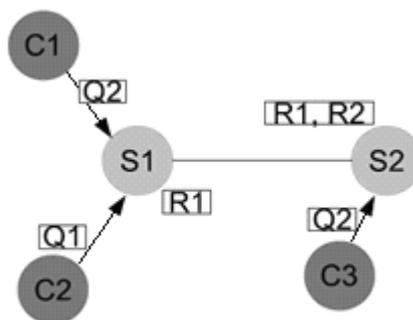


Figura 1 - PPRDR em uma RDC de 2 servidores

A Figura 1 mostra um exemplo do PPRDR em uma RDC em um único instante de tempo. A figura mostra dois servidores e três clientes. O servidor 1 possui uma réplica do conteúdo 1 (R1), o servidor 2 possui uma réplica do conteúdo 1 e uma do conteúdo 2, e cada cliente envia uma requisição (Q) para um conteúdo na RDC. Observe que a solução mostrada pela figura é inviável porque a requisição do cliente 1 é atendida por um servidor que não possui uma réplica do conteúdo exigido pela requisição. Uma possível mudança para tornar a solução viável é a mudança da requisição vinda do cliente 1 do servidor 1 para o servidor 2.

O dinamismo do problema se encontra na modificação que os dados sofrem ao longo dos períodos considerados. Neste trabalho, considerou-se três possíveis tipos de alterações nos dados: Mudanças nos conteúdos, na rede e nas requisições. As Mudanças na rede ocorrem através da mudança no atraso entre servidores, afetando assim os custos de comunicação. Mudanças nos

conteúdos ocorrem através do surgimento de novos conteúdos e mudanças nas requisições ocorrem através do surgimento e finalização de requisições. O fato do problema ser *online* indica que os dados não são conhecidos a priori, ou seja, não é possível determinar em um instante de tempo quais serão as demandas e a configuração da rede nos períodos seguintes.

Assim sendo, as características do problema a ser abordado neste trabalho são:

- Capacitado - existência de limites, heterogêneos, de banda e espaço nos servidores.
- Multiproduto - existência de mais de um conteúdo na rede.
- Dinâmico - mudanças na rede, conteúdos e demandas ocorrem ao longo tempo.
- Online – dados sobre o futuro não são conhecidos a priori. Os dados vão sendo conhecidos durante a execução e nada se pode supor sobre dados futuros.
- Demanda divisível - possibilidade de atendimento de uma demanda por mais de um servidor.
- Demanda fracionária - possibilita a divisão de conteúdos em frações. Os conteúdos são divididos em Mega Bytes, existindo a possibilidade de um servidor entregar uma fração do conteúdo.
- QoS - presença de requisitos mínimos de banda e atraso nas requisições.
- Atendimento otimizado - tentativa de atender melhor os clientes sempre que possível, utilizando limites máximos nas requisições.

3. Trabalhos Relacionados

Várias abordagens já foram feitas para problemas relacionados à RDC. Em [Bektas 2007], [Almeida 2004] e [Huang 2004] o Problema de Posicionamento de Réplicas estático é abordado. O primeiro apresenta um algoritmo exato baseado na decomposição de Bender, e o segundo, apresenta 6 heurísticas. Em ambos os trabalhos, são apresentados formulações matemáticas para o problema. Em [Huang 2004] os autores afirmam que apesar do grande número de trabalhos relacionados com RDC, pouca atenção foi dada às questões relacionadas com a QoS percebida pelos clientes. Uma modelagem por grafos é utilizada e uma abordagem distribuída baseada na estratégia de dominação de grafos é proposta para o problema.

Em [Bartolini 2003] [Zhou 2007] o Problema de Posicionamento de Réplicas em sua forma dinâmica é abordado através de abordagens centralizadas. Em [Bartolini 2003] os autores lidam com o problema modelando-o como um processo de Markov. Já em [Zhou 2007] os autores lidam com o problema usando uma abordagem heurística, compondo três heurísticas para o problema de replicação de conteúdos e duas para o posicionamento das réplicas. Pelo conhecimento dos autores, este trabalho é o único que apresentou o uso de uma metaheurística, a saber, *Simulated Annealing*.

Já em [Aioffi 2005] [Tenzakthi 2004] [Wauters 2005], foram feitas abordagens distribuídas para o problema em sua versão dinâmica. Em [Aioffi 2005] é proposta uma formulação matemática para a versão *offline* e uma heurística distribuída, baseada em um método de previsão, usada para a versão *online*. As soluções obtidas são comparadas com um outro algoritmo proposto na literatura, mostrando que as soluções encontradas pelos métodos propostos superam o algoritmo da literatura e que as soluções encontradas pela estratégia distribuída estão próximas da solução ótima. Em [Tenzakthi 2004], os algoritmos levam em consideração taxas calculadas com base nos custos de comunicação, processamento e na frequência de acesso às réplicas. uma abordagem centralizada é proposta, na qual um servidor central recolhe dados e executa um algoritmo guloso de posicionamento. Na abordagem distribuída, as decisões são tomadas localmente de acordo com as informações armazenadas em cada servidor. Os algoritmos propostos são comparados entre si tendo o distribuído apresentado melhores resultados. Em [Wauters 2005], os autores utilizam conceitos de redes P2P, fazendo com que os servidores se comuniquem entre si, trocando informação sobre a carga e tráfego. Duas heurísticas são propostas. Apenas os resultados da primeira heurística foram mostrados e comparados com uma série de outras heurísticas expostas na literatura, mostrando que a heurística proposta obteve a melhor relação custo benefício em termos de qualidade de solução e tempo de execução.

Em Coppens et al [Coppens 2006], um método híbrido é proposto para resolver o problema de posicionamento de réplicas dinâmico. Segundo os autores as abordagens centralizadas podem atingir o posicionamento ótimo, porém, estas estratégias não são escaláveis para grandes redes. Na abordagem proposta, todos os servidores executam uma cópia do algoritmo levando em conta apenas dados locais e algumas informações enviadas pela arquitetura da RDC. Depois disso, cada servidor procura replicar os conteúdos de maior custo de comunicação, trazendo para si os conteúdos mais desejados pelos clientes ligados a este servidor. Esta abordagem híbrida apresenta resultados que são compatíveis com as abordagens centralizadas e ainda mantém as qualidades de uma abordagem distribuída como, por exemplo, escalabilidade e tolerância à falhas.

4. Formulação Matemática para o problema

Tendo observado o uso de métodos exatos nos trabalhos relacionados e já tendo feito alguns estudos preliminares sobre o tema [Neves 2008b] [Neves 2008], foi proposta, neste trabalho, uma formulação matemática considerando as várias características do problema. Apesar do fato de que as formulações matemáticas não são adequadas para a resolução de problemas *online* seu uso ainda sim é importante para obtenção de limites, os quais podem ser usados como guias para as demais abordagens. Assim, a formulação apresentada nesta seção é utilizada como métodos para resolução da versão *offline* do PPRDR e os resultados obtidos pela mesma usados como guias para as demais abordagens. Vale a pena lembrar que, na versão *offline*, todas as mudanças são conhecidas a priori, fato este que possibilita uma alocação muito mais eficiente das réplicas, reduzindo assim, os custos operacionais.

A formulação proposta é baseada na formulação apresentada em [Aioffi 2005], adaptando alguns detalhes relativos à diferença entre os problemas e acrescentado o conceito de *backlog* [An 2005] para modelar a demanda residual entre os períodos de tempo. Para modelar a atualização de conteúdos, foi proposto o uso da atribuição de um tempo de vida para os conteúdos como meio de incorporar este novo aspecto. Esta formulação também leva em consideração o fato de que os conteúdos podem surgir de servidores diferentes.

Sejam as funções origem(i) que retorna o servidor de origem da requisição i , $ld(i)$ que retorna o atraso local da requisição i , $atraso(j_1, j_2)$ que retorna o atraso entre dois servidores e $RTT(j_1, j_2)$ que retorna o valor do tempo que uma mensagem leva para percorrer o caminho de um servidor j_1 para um outro servidor j_2 e voltar ao primeiro (*Round Trip Time* - RTT). Para calcular a demanda de um cliente em um determinado período de tempo (D_i) são feitas duas operações. Uma divisão inteira e um resto de divisão, ambos usando o tamanho do conteúdo como dividendo e a banda máxima do cliente como divisor. Com o quociente da primeira operação descobre-se o número de períodos em que a requisição deve ser atendida na plena capacidade, com o resultado da segunda, o valor que deve ser enviado no último período de atendimento da requisição.

Variáveis:

- x_{ijt} fração do conteúdo solicitado pela requisição i entregue pelo servidor j no período t .
- y_{kjt} 1 se o conteúdo k está replicado no servidor j no período de t . 0, caso contrário
- b_{it} *backlog* da requisição i no período t .
- $w_{kjlt} = 1$ se o conteúdo k é copiado pelo servidor j a partir do servidor l no período t . 0, caso contrário.

Constantes:

- R conjunto de requisições a serem atendidas.
- S conjunto de servidores da RDC.
- C conjunto de conteúdos a serem replicados.
- T conjunto de períodos de tempo.
- L_k o tamanho do conteúdo k .
- B_k período em que o conteúdo k é disponibilizado.
- E_k período em que o conteúdo k é removido da RDC.
- O_k servidor origem do conteúdo k .
- AS_j espaço em disco disponível no servidor j .

- MB_j banda máxima do servidor j .
- D_{it} demanda da requisição i no período t .
- BR_i banda mínima exigida pela requisição i .
- BX_i banda máxima aceita pela requisição i .
- $g(i)$ conteúdo exigido pela requisição i .
- c_{ijt} custo de atendimento da requisição i no servidor j , no período t , calculado pela seguinte equação $c_{ijt} = (RTT_{origem(i),j,t} + Delay_{origem(i),j,t} + ld(i)) BR_i$.
- p_{it} penalidade por usar *backlog* da requisição i no período t .

$$Min \quad \sum_{i \in R} \sum_{j \in S} \sum_{t \in T} c_{ijt} x_{ijt} + \sum_{i \in R} \sum_{t \in T} p_{it} b_{it} + \sum_{k \in C} \sum_{j \in S} \sum_{l \in S} \sum_{t \in T} L_k w_{kjl} t \quad (1)$$

S.a.

$$\sum_{j \in S} L_{g(i)} x_{ijt} - b_{i(t-1)} + b_{it} = D_{it} \quad \forall i \in R, \forall t \in [BG_i, EG_i] \quad (2)$$

$$\sum_{i \in R} L_{g(i)} x_{ijt} \leq MB_j \quad \forall j \in S, \forall t \in T \quad (3)$$

$$\sum_{j \in S} L_{g(i)} x_{ijt} \leq BX_i \quad \forall i \in R, \forall t \in T \quad (4)$$

$$\sum_{j \in S} \sum_{t \in T} x_{ijt} = 1 \quad \forall i \in R \quad (5)$$

$$y_{g(i)jt} \geq x_{ijt} \quad \forall i \in R, \forall j \in S, \forall t \in T \quad (6)$$

$$\sum_{j \in S} y_{kjt} \geq 1 \quad \forall k \in C, \forall t \in [B_k, E_k] \quad (7)$$

$$y_{kjt} = 0 \quad \forall k \in C, \forall j \in S, \forall t \notin [B_k, E_k] \quad (8)$$

$$y_{kOk} = 1 \quad \forall k \in C \quad (9)$$

$$y_{kjB_k} = 0 \quad \forall k \in C, \forall j \in \{S \mid j \neq O_k\} \quad (10)$$

$$y_{kj(t+1)} \leq \sum_{l \in S} w_{kjl} t \quad \forall k \in C, \forall j \in S, \forall t \in T \quad (11)$$

$$y_{kjt} \geq w_{kjl} t \quad \forall k \in C, \forall j \in S, \forall l \in S, \forall t \in T \quad (12)$$

$$\sum_{k \in C} L_k y_{kjt} \leq AS_j \quad \forall j \in S, \forall t \in T \quad (13)$$

$$x_{ijt} \in [0,1] \quad \forall i \in R, \forall j \in S, \forall t \in T \quad (14)$$

$$y_{kjt} \in \{0,1\} \quad \forall k \in C, \forall j \in S, \forall t \in T \quad (15)$$

$$b_{it} \geq 0 \quad \forall i \in R, \forall t \in T \quad (16)$$

$$w_{kjl} t \in \{0,1\} \quad \forall j, l \in S, \forall k \in C, \forall t \in T \quad (17)$$

A função objetivo, exposta em (1), tenta minimizar o custo de entrega dos conteúdos para os clientes bem como a quantidade de atrasos na entrega dos conteúdos (*backlog*) feitos ao longo do tempo e ainda tenta reduzir o número de replicações dos conteúdos através da rede. As restrições (2) associam as variáveis x_{ijt} e b_{it} , dizendo que a soma das quantidades entregues no dia atual, mais a quantidade que se ficará devendo para o dia posterior é igual à demanda do dia atual mais o que se ficou devendo do dia anterior. As restrições (3) exigem que o fluxo total entregue por um servidor seja menor ou igual à sua capacidade máxima. As restrições (4) impedem que seja entregue ao cliente uma banda maior do que ele suporta. As restrições (5) exigem que uma requisição seja plenamente atendida. As restrições (6) condicionam que uma requisição só possa ser atendida por um servidor que possua uma réplica do conteúdo exigido. As restrições (7) e (8) controlam o número de réplicas de um conteúdo, afirmando que no mínimo uma réplica deve existir durante o tempo de vida do conteúdo e que nenhuma réplica pode existir fora do período

de vida. As restrições (9) e (10) fazem com que no período de surgimento de um conteúdo apenas o servidor origem deste conteúdo possua uma réplica. As restrições (11) garantem que toda replicação crie uma nova réplica. As restrições (12) exigem que uma replicação ocorra a partir de um servidor que possua o conteúdo replicado. As restrições (13) dizem que a soma dos tamanhos dos conteúdos em um servidor não pode exceder o espaço em disco disponível. As demais restrições são as de integralidade e não negatividade.

Esta formulação, chamada a partir deste ponto de *FDI*, é usada para obter limites de qualidade para comparações com as demais abordagens.

5. Heurística para o Problema

Com o objetivo de apresentar uma primeira forma de abordar a versão *online* do problema foi proposta neste trabalho uma derivação da heurística CORA [Tenzekhtil (2004)] introduzida para um problema semelhante ao tratado neste trabalho. A heurística CORA posiciona gulosamente réplicas em servidores baseando-se no custo de comunicação dos servidores, ou seja, número de requisições feitas a eles. Tendo o posicionamento das réplicas ela então resolve heurísticamente a associação das requisições aos servidores.

Observando a formulação proposta para o problema apresentada na seção 4, pode-se observar claramente que esta formulação trata de um problema inteiro misto, ou seja, existem variáveis contínuas e variáveis inteiras. Ao separar estes dois conjuntos de variáveis, percebe-se que as variáveis contínuas estão relacionadas à associação de requisições a servidores e que as variáveis inteiras estão relacionadas com a replicação de conteúdos em servidores.

Assim sendo, dado um esquema de replicação, resolver o problema de associação de requisições aos servidores torna-se um problema linear. Considerando os fatos apresentados a seguinte heurística foi proposta para a versão *online* do problema:

Algoritmo CH

1: Resolver o posicionamento para o período inicial

2: **Para** cada período de tempo restante **faça**

3: Resolver de forma exata a associação requisição-servidor

4: Montar heurísticamente o esquema de replicação para o próximo período

5: **Fim Para**

Algoritmo 1 Heurística CH

O algoritmo CH, mostrado acima, é um algoritmo construtivo que resolve de forma exata a associação entre requisições e servidores e de maneira heurística o esquema de replicação. Para cada período de tempo, o algoritmo resolve a associação entre requisições e servidores usando uma formulação matemática extraída da formulação *FDI*. Para posicionar as réplicas para o período seguinte, a heurística utiliza uma média das demandas dos períodos anteriores para tentar prever onde serão os pontos de maior demanda no próximo período.

Para a resolução da associação entre requisições e servidores as seguintes notações tiveram sua semântica alterada:

Variáveis:

- x_{ij} fração do conteúdo solicitado pela requisição i entregue pelo servidor j no período atual.
- b_i *backlog* da requisição i para o período seguinte.

Constantes:

- B_i *backlog* anterior da requisição i .
- Y_{kj} indica se o conteúdo k está replicado no servidor j . 1 se verdadeiro e 0 caso contrário.
- D_i demanda da requisição i no período atual.
- c_{ij} custo de atendimento da requisição i no servidor j , calculado pela seguinte equação $c_{ij} = (\text{RTT}_{\text{origem}(i),j,t} + \text{Delay}_{\text{origem}(i),j,t} + \text{ld}(i)) \text{BR}_i$.
- p_i penalidade por usar *backlog* da requisição i fazendo com que parte da demanda mínima seja atendida no período seguinte.

As demais notações que aparecem em trechos da formulação mantêm a semântica apresentada na seção 4.

$$\text{Min} \quad \sum_{i \in R} \sum_{j \in S} c_{ij} x_{ij} \quad \sum_{i \in R} p_i b_i \quad (18)$$

S.a.

$$\sum_{j \in S} L_{g(i)} x_{ij} + b_i = D_i + B_i \quad \forall i \in R \quad (19)$$

$$\sum_{i \in R} L_{g(i)} x_{ij} \leq MB_j \quad \forall j \in S \quad (20)$$

$$\sum_{j \in S} L_{g(i)} x_{ij} \leq BX_i \quad \forall i \in R \quad (21)$$

$$x_{ij} \leq Y_{g(i)j} \quad \forall i \in R \forall j \in S \quad (22)$$

$$x_{ij} \in [0,1] \quad \forall i \in R \forall j \in S \quad (23)$$

$$b_i \geq 0 \quad \forall i \in R \quad (24)$$

É importante mencionar que esta formulação para o subproblema é linear, ou seja, não possui variáveis inteiras. Este fato facilita muito a resolução do problema, visto que o maior complicador para a resolução de sistemas lineares é justamente a presença de variáveis inteiras.

Para resolver o posicionamento de réplicas foi utilizada uma heurística gulosa. A heurística ordena cada tupla conteúdo/servidor de acordo com o custo de comunicação. Após esta ordenação a heurística tenta inserir o conteúdo da tupla de maior custo no servidor, desta mesma tupla. Caso haja espaço disponível no servidor, a réplica é simplesmente posicionada. Caso contrário, é necessário liberar espaço em disco para que o conteúdo possa ser inserido, assim, é feita uma tentativa de remoção de uma das réplicas presentes no servidor baseada no custo de comunicação das réplicas existentes. Caso o conteúdo que estamos tentando inserir, possua um custo de comunicação maior do que algum dos conteúdos presentes no servidor, o conteúdo presente no servidor é removido em função do novo conteúdo.

6. Resultados computacionais

Os algoritmos apresentados nas seções anteriores foram implementados na linguagem C++ com o uso do compilador g++ versão 4.3 e executados em um computador Quad-Core com 2.83 GHz por core, 8 Gigabytes de RAM rodando um sistema operacional Linux com kernel 2.6. Para resolver as formulações, tanto a FD1 quanto a presente na heurística, foi utilizado o resolvidor CPLEX 11.2 [ILOG S.A. (2008)].

Devido ao desconhecimento, por parte dos autores, acerca da existência de instâncias que contemplem todas as características do problema, foram geradas 60 instâncias com números variados de servidores, conteúdos e requisições. Estas instâncias foram utilizadas nos testes e estão disponíveis no site do projeto LABIC [Labic (2009)].

A Tabela 1 mostra os resultados obtidos pela formulação FD1e pela heurística CH. Na primeira coluna estão os nomes das instâncias, na segunda o número de servidores, na terceira, o número de requisições, na quarta está o número de conteúdos. Na quinta e sexta colunas estão os tempos de execução da FD1 e CH respectivamente. Na sétima e oitava estão os valores das funções objetivo alcançados pelos métodos. A nona coluna mostra o GAP entre os dois métodos. Note que para uma única instância (2011) a heurística CH não foi capaz de encontrar uma solução viável para o problema.

Uma análise importante a ser feita quando se comparam algoritmos dentro de uma RDC é o número de cópias geradas. Para este quesito foram usadas instâncias menores para uma coleta detalhada de informações, mostrando que a heurística proposta produz, na maioria das vezes, o mesmo número de réplicas que a formulação matemática. Foram usadas 20 instâncias neste teste e em apenas 5 delas o número de réplicas geradas pela heurística foi diferente da formulação.

A Tabela 2 mostra as diferenças no número de replicações da Formulação FD1 e da heurística CH. A primeira coluna mostra o nome da instância, a segunda e a terceira mostram o

Artigo apresentando e publicado nos Anais do XLI Simpósio Brasileiro de Pesquisa Operacional – XLI SBPO - 2009

número de réplicas geradas pela formulação FD1 e pela heurística CH, respectivamente. A quarta coluna mostra a diferença percentual entre a segunda e terceira colunas.

Instância	# Servidores	# Requisições	# Conteúdos	Tempo FD1(s)	Tempo CH (S)	FO FD1	FO CH	GAP(%)
1001	10	613	5	1,23	0,09	300268	304670	1,466
1002	10	526	6	1,08	0,09	272965	283353	3,806
1003	10	519	5	1,05	0,11	253842	261003	2,821
1004	10	632	6	1,32	0,08	349568	358468	2,546
1005	10	445	5	0,87	0,06	265562	269060	1,317
1006	10	627	12	4,76	0,4	2,00E+006	2,09E+006	4,520
1007	10	675	11	5,03	0,42	2,14E+006	2,23E+006	4,178
1008	10	641	13	4,84	0,36	2,07E+006	2,18E+006	5,160
1009	10	659	14	4,98	0,42	2,08E+006	2,17E+006	4,115
1010	10	649	15	5,12	0,42	2,10E+006	2,24E+006	6,529
1011	10	627	12	12,54	0,4	2,10E+006	2,32E+006	10,450
1012	10	675	11	7,59	0,41	2,16E+006	2,27E+006	5,300
1013	10	641	13	13	0,43	2,15E+006	2,31E+006	7,635
1014	10	659	14	19,96	0,4	2,22E+006	2,41E+006	8,586
1015	10	649	15	19,3	0,41	2,24E+006	2,56E+006	14,612
2001	20	715	4	3,46	0,24	463848	477181	2,874
2002	20	1525	4	7,02	0,59	821229	822871	0,200
2003	20	1602	5	8,71	0,63	878728	892866	1,609
2004	20	1224	5	6,26	0,56	649137	664332	2,341
2005	20	1366	4	6,48	0,41	754288	763235	1,186
2006	20	1289	15	29,06	1,41	4,20E+006	4,45E+006	5,991
2007	20	1356	11	26,93	1,6	4,33E+006	4,49E+006	3,704
2008	20	1314	13	29,44	1,49	4,29E+006	4,48E+006	4,438
2009	20	1352	14	29,18	1,47	4,43E+006	4,66E+006	5,055
2010	20	1367	12	27,31	1,43	4,48E+006	4,65E+006	3,898
2011	20	1289	15	449,06	1,65	4,78E+006	1,26E+008	2541,409
2012	20	1356	11	175,67	1,57	4,42E+006	4,67E+006	5,532
2013	20	1314	13	97,79	1,52	4,50E+006	4,90E+006	8,910
2014	20	1352	14	113,46	1,49	4,67E+006	5,06E+006	8,143
2015	20	1367	12	76,74	1,96	4,55E+006	4,80E+006	5,316
3001	30	2190	5	17,8	0,98	1,24E+006	1,25E+006	0,655
3002	30	1793	6	14,5	0,73	1,11E+006	1,14E+006	2,911
3003	30	1845	6	17,47	0,89	1,10E+006	1,12E+006	1,900
3004	30	1896	6	15,36	0,87	1,05E+006	1,08E+006	2,834
3005	30	2384	6	18,52	1,05	1,43E+006	1,46E+006	1,697
3006	30	2007	12	73,59	3,34	6,56E+006	6,86E+006	4,595
3007	30	1963	12	80,75	3,14	6,34E+006	6,63E+006	4,563
3008	30	2021	11	72,06	3,64	6,56E+006	6,85E+006	4,507
3009	30	1991	11	72,18	4,4	6,46E+006	6,70E+006	3,712
3010	30	1998	11	74,19	3,1	6,46E+006	6,72E+006	4,019
3011	30	2007	12	185,24	3,3	6,70E+006	7,13E+006	6,472
3012	30	1963	12	161,67	3,06	6,39E+006	6,76E+006	5,660
3013	30	2021	11	148,64	3,38	6,61E+006	7,00E+006	5,871
3014	30	1991	11	132,08	3,31	6,53E+006	6,86E+006	5,087
3015	30	1998	11	138,92	3,1	6,52E+006	6,87E+006	5,394
5001	50	3360	4	62,38	2,78	2,08E+006	2,10E+006	0,554
5002	50	3231	6	46,87	3,2	1,92E+006	1,97E+006	2,940
5003	50	3534	6	52,93	2,66	2,23E+006	2,27E+006	1,769
5004	50	3646	4	62,62	2,64	2,77E+006	2,96E+006	6,950
5005	50	3762	5	56,35	3,12	2,04E+006	2,08E+006	1,773
5006	50	3391	11	270,36	9,9	1,12E+007	1,16E+007	3,752
5007	50	3329	11	275,63	8,9	1,09E+007	1,14E+007	4,603
5008	50	3214	15	348,16	9,12	1,05E+007	1,11E+007	5,611
5009	50	3303	15	355,51	9,11	1,10E+007	1,16E+007	5,724
5010	50	3295	13	304,85	9,83	1,10E+007	1,15E+007	4,971
5011	50	3391	11	504,87	9,73	1,13E+007	1,19E+007	5,073
5012	50	3329	11	396,97	8,84	1,09E+007	1,15E+007	5,312
5013	50	3214	15	1254,76	8,8	1,15E+007	1,28E+007	11,851
5014	50	3303	15	788,06	9,07	1,19E+007	1,33E+007	11,732
5015	50	3295	13	673,54	11,5	1,12E+007	1,20E+007	7,242

Tabela 1- Resultados

Instância	# Rep. FD1	# Rep. CH	Dif. Perc. (%)
1002	49	56	14,285
1005	38	40	5,263
3002	152	160	5,263
3003	148	150	1,351
5003	245	260	6,122

Tabela 2 Replicações

Outro teste importante é o teste que mede o impacto que o espaço disponível nos servidores da RDC tem sobre os algoritmos. Para este teste foram geradas 40 instâncias, 20 delas sem restrição de espaço nos servidores, e 20 com restrições nos servidores e idênticas nos outros aspectos. O Gráfico 1 mostra o GAP mediano dos resultados obtidos pela heurística CH. O impacto da presença de restrições pode ser claramente percebido visto que os GAPs das instâncias com restrição são sempre superiores.

Outra informação que pode ser observada no gráfico é o fato de que os GAPs permanecem relativamente estáveis independente do tamanho das instâncias.

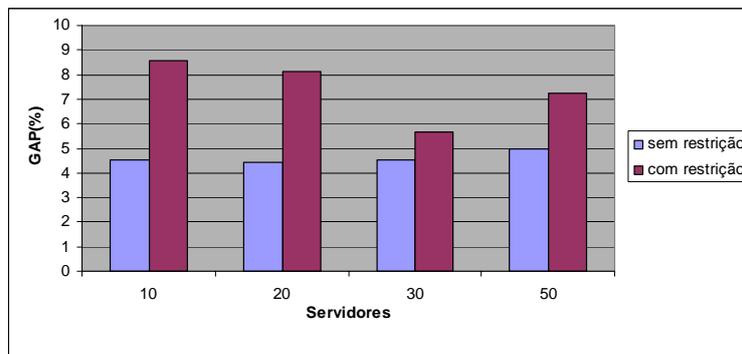


Gráfico 1 Resultados - Medianas

A Gráfico 2 mostra um comparativo dos tempos médios obtidos pela formulação FD1 e pela heurística CH. Como pode ser observado, os tempos de execução da heurística CH são ínfimos quando comparado com os da formulação.

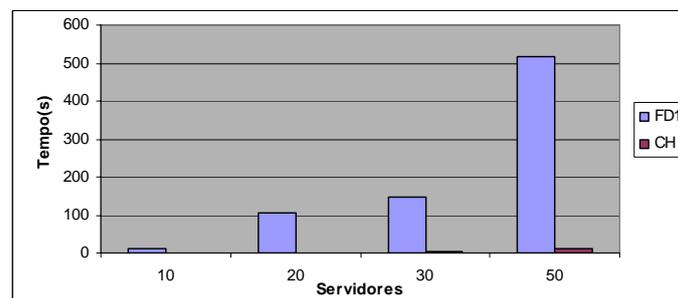


Gráfico 2 Resultados - Tempos

7. Conclusões

Neste trabalho foram apresentadas uma descrição do Problema de Posicionamento de Réplicas e Distribuição de Requisições, presente em Redes de Distribuição de Conteúdos, bem como uma formulação matemática para sua versão *offline* e uma heurística gulosa para a versão *online*.

A heurística proposta é um método híbrido, que resolve aproximadamente o problema de posicionamento das réplicas e de forma exata a associação de requisições a servidores. Os

resultados mostram que a heurística consegue atingir bons resultados com menos tempo computacional, tendo em vista que a formulação é aplicada à versão *offline* do problema.

Uma análise mais detalhada se faz necessária para descobrir se o GAP, apresentado pela heurística, é decorrente da diferença entre as versões *online* e *offline*, ou se é decorrente da diferença natural entre métodos exatos e heurísticos.

Agradecimentos

Os autores gostariam de agradecer a CAPES, o CNPQ e a FAPERJ.

Referências

- Akamai**(2008), website da companhia, www.akamai.com. Acessado em 03/2008.
- Level 3** (2009), website da companhia, www.level3.com. Acessado em 01/2009.
- Labic** (2009), website do projeto, <http://labic.ic.uff.br>, Acessado em 02/2009
- Mirror Image** (2009), website da companhia, www.mirror-image.com. Acessado em 01/2009.
- Aioffi, W., Mateus, G., Almeida, J., Loureiro, A.** (2005), Dynamic Content Distribution for Mobile Enterprise Networks. *IEEE Journal on Sel. Areas in Communications*, 23(10), 2022-2031.
- Almeida, J. M., Eager, D. L., Vernon, M. K., Wright, S. J.** (2004), Minimizing delivery cost in scalable streaming content distribution systems, *IEEE Transactions on Multimedia*, 6, 356-365.
- An, C., Fromm, H.** Eds., Supply Chain Management on Demand: Strategies and Technologies, Applications, Springer, 2005.
- Bakiras, S., Loukopoulos, T.** (2005), Combining replica placement and caching techniques in content distribution networks, *Computer Communications*, 28, 1062-1073.
- Bartolini, N., Presti, F., Petrioli, C.** (2003), Optimal dynamic replica placement in content delivery networks, *The 11th IEEE Intern. Conference on Networks 2003 ICON2003*, 125-130.
- Bektas, T., Oguz, O., Ouveysi, I.** (2007), Designing cost-effective content distribution networks, *Computers & Operations Research*, 34, 2436-2449.
- Coppens, J., Wauters, T., Turck, F. D., Dhoedt, B., Demeester, P.** (2006), Design and performance of a self-organizing adaptive content distribution network, *Network Operations and Management Symposium, 2006. NOMS 2006, 10th IEEE/IFIP*, 534-545.
- Dias, J., Captivo, M. E., Clímaco, J.** (2006), Capacitated dynamic location problems with opening, closure and reopening of facilities, *Journal of Management Mathematics*, 17, 317-348.
- Galvão, R. D., Acosta Espejo, L. G., Boffey, B.** (2002), A hierarchical model for the location of perinatal facilities in the municipality of Rio de Janeiro, *European Journal Of Operational Research*, 138, 495-517.
- Huang, C., Abdelzaher, T.** (2004), Towards content distribution networks with latency guarantees, *Twelfth IEEE Intern. Workshop on Quality of Service, 2004. IWQOS 2004*, 181-192.
- ILOG S.A.**, CPLEX 11 user's manual, 2008.
- Korupolo, M. R., Plaxton, C.G.** (2000), Analysis of a local search heuristic for facility location problems, *Journal of Algorithms*, 37, 146-188.
- Leighton, F. T., Lewin, D.** (2000), Global hosting system, US Patent:US006108703, Aug 2000.
- Neves, T., Drummond, L., Uchoa, E., Albuquerque, C.** (2008), Otimização em redes de distribuição de conteúdos, *Anais do XL Simp. Brasileiro de Pesquisa Operacional*, 2345-2356.
- Neves, T., Ochi, L., Drummond, L., Uchoa, E., Albuquerque, C.** (2008), Optimization in content distribution networks, *Proc. of Intern. Conf. on Engineering Optimization EngOpt2008*.
- Pirkul, H., Jayaraman, V.** (1998), A multi-commodity, multi-plant capacitated facility location problem: Formulation and Efficient Heuristic Solution, *Comp. & Operat. Research*, 25, 869-878.
- Tenzekhti, F., Day, K., Ould-Khaoua, M.** (2004), Replication algorithms for the world-wide web, *Journal of Systems Architecture*, 50, 591-605.
- Wauters, T., Coppens, J., Dhoedt, B., Demeester, P.** (2005), Load balancing through efficient distributed content placement, *Next Generation Internet Networks*, 99-105.
- Wolfson, O., Jajodia, S., Huang, Y.** (1997), An adaptive Data Replication Algorithm, *ACM Transactions on Database Systems*, 22(2), 255-314.
- Zhou, X., Xu, C.-Z.** (2007), Efficient algorithms of video replication and placement on a cluster of streaming servers, *Journal of Network and Computer Applications*, 30, 515-540.