

Avaliação Quantitativa do Desempenho de Leiautes de Disco para Cargas Realistas

Ítalo Cunha, Jussara Almeida, Marcus Rocha¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Pampulha
30123-970 Belo Horizonte, MG

cunha@dcc.ufmg.br, jussara@dcc.ufmg.br, mvrocha@dcc.ufmg.br

Abstract. *The impact of user behavior and workload characteristics on the performance of real streaming media services has not been thoroughly studied yet. Unlike previous work, which considered mainly sequential and bandwidth-intensive synthetic workloads, this paper provides a quantitative performance evaluation of different disk layout strategies using realistic workloads, derived from the analysis of real systems. Striping, randomized I/O as well as sequential layout are thoroughly analyzed, through simulation, for a number of system and workload configurations, with different levels of user interactivity and average content bitrate, observed in real workloads.*

Resumo. *Os impactos do comportamento do usuário e características da carga no desempenho de serviços reais de mídia contínua ainda não foram exaustivamente estudados. Ao contrário de trabalhos prévios, que consideraram apenas cargas sintéticas com altos requisitos de banda e acesso sequencial, este artigo apresenta uma avaliação quantitativa do desempenho de diferentes estratégias de leiautes de disco, usando cargas realistas derivadas da análise de sistemas reais. Os leiautes por faixas, aleatório e sequencial, são profundamente analisados, através de simulação, para várias configurações do sistema e da carga, com diferentes níveis de interatividade do usuário e taxas de codificação dos arquivos, observadas em cargas reais.*

1. Introdução

O desempenho de um servidor multimídia depende do desempenho de seu sistema de armazenamento. Em particular, o número de requisições de clientes que o servidor consegue atender simultaneamente em um dado momento depende da banda efetiva que seu sistema de disco provê, o que, por sua vez, depende não só das características da carga (ex: tamanho e taxa de codificação do arquivo, padrão de acesso do usuário) mas também do leiaute de alocação de dados usado para armazenar os arquivos nos discos.

Há três principais estratégias de alocação de dados em sistemas de armazenamento para servidores de mídia. O esquema sequencial tradicional, que armazena o arquivo inteiro em apenas um disco, é a estratégia mais simples. Porém ela não escala muito bem, gerando contenção nos discos que contêm os arquivos mais populares.

Para resolver este problema, o leiaute por faixas (*striping*) [1, 8, 13] e o leiaute aleatório [7, 11, 12] foram propostos. Em ambas as estratégias, cada arquivo é dividido em blocos de tamanho igual. No leiaute por faixas estes blocos são distribuídos nos discos de uma maneira *round-robin*; no leiaute aleatório o disco onde cada bloco será armazenado é selecionado aleatoriamente, possibilitando a replicação de alguns blocos em múltiplos discos. Os leiautes por faixas e aleatório já foram analisados no passado [1, 8, 10, 12], sendo que o resultado principal foi de que o leiaute aleatório é mais simples e possui desempenho competitivo. Esses estudos consideraram principalmente cargas sintéticas sequenciais, onde as requisições dos clientes são para a mídia completa, os arquivos são grandes e possuem alta taxa de codificação (ex: 1 Mbit/s). Assim, cada requisição do cliente é atendida por múltiplos discos, em ambos os esquemas.

Porém, apesar de todos os esforços de pesquisa com o intuito de desenvolver mecanismos custos-efetivos, para distribuição escalável de mídia contínua de alta qualidade através da Internet [4, 7, 13], quase todos os arquivos de mídia transferidos atualmente na Internet possuem baixa taxa de codificação. Isto é verdade não só para conteúdo ao vivo [6], mas também para conteúdo previamente armazenado [2, 6], provavelmente porque a banda disponível na rede e no cliente são ainda limitadas. Em [2], nós mostramos que apesar de vídeos educacionais distribuídos a estudantes, através de uma rede de um campus com alta banda, serem codificados com uma taxa média de apenas 300 Kbit/s, vídeo e áudio de entretenimento distribuídos, através da Internet por dois dos maiores distribuidores de conteúdo da América Latina, consistem em sua maior parte de arquivos pequenos com taxa de codificação ainda menor (no máximo 100 Kbit/s). Além disto, um alto grau de interatividade foi observado para vídeos de longa duração.

Assim, a premissa de acesso sequencial e completo a arquivos com alta taxa de codificação não é verdadeira para vários servidores de mídia *reais* na Internet atual. Este fato levanta a questão de qual o desempenho quantitativo que cada leiaute de disco existente é capaz de prover para cargas que são *realistas* para sistemas *correntes*.

Este artigo revisita o problema de leiautes de alocação de dados para servidores de mídia e provê uma avaliação quantitativa de desempenho dos três esquemas existentes, especificamente, alocação sequencial, alocação por faixas e alocação aleatória, com diferentes estratégias de replicação e balanceamento de carga. Nosso objetivo é mostrar resultados práticos. Assim, ao contrário de trabalhos prévios, consideramos configurações de sistema e da carga que são realistas para sistemas atuais.

Nós desenvolvemos um simulador de sistema de disco de alta-fidelidade, que implementa cada um destes três esquemas. As simulações são dirigidas por cargas sintéticas realistas criadas usando o GENIUS [3], um gerador de cargas realistas para servidores de mídia, que é parametrizado com resultados de uma extensa caracterização de cargas de mídia reais [2]. Nossos resultados principais mostram que alocação sequencial provê latência inicial competitiva para cargas leves e médias. Alocação aleatória é robusta a variações na configuração da carga e do servidor, enquanto o leiaute por faixas apresenta, como esperado, latência muito mais alta para as cargas consideradas (ex: 5 vezes maior) do que os outros leiautes.

O restante deste artigo é organizado como segue. A seção 2 descreve os três leiautes analisados bem como outros trabalhos relacionados. A seção 3 apresenta nosso

modelo do sistema de disco. Nosso ambiente de simulação e as cargas usadas são descritos na seção 4. A seção 5 apresenta os resultados obtidos mais relevantes. As principais conclusões estão sumarizadas na seção 6.

2. Leiautes de Disco

Alocação seqüencial é o leiaute mais simples que alguém pode usar em servidores de mídia. Este leiaute causa contenção, se a carga não for balanceada, nos discos com os objetos de mídia mais populares. Como alternativas a este leiaute, os seguintes foram propostos:

O leiaute por faixas para servidores de mídia foi originalmente proposto e analisado em [13]. No leiaute por faixas, vídeos são divididos em blocos de tamanho igual. No leiaute por faixas de grão-grosso, blocos de objetos de mídia são alocados de forma *round-robin* através dos discos do servidor. Se o bloco b_i está no disco d_k , então b_{i+x} está em $d_{(k+x) \bmod D}$, onde D é o número total de discos no servidor. No leiaute por faixas de grão-fino cada disco recebe uma pequena parte de cada bloco, de modo que uma requisição a um bloco resulta em acesso a todos os discos. Ambos leiautes garantem que clientes assistindo a mídia de forma seqüencial irão distribuir a carga entre todos os discos do servidor. O leiaute por faixas trabalha em ciclos. A cada ciclo, o servidor envia para o cliente um bloco de mídia. Ignorando quaisquer atrasos, que podem ser contornados com o uso de *buffers*, um ciclo de duração $\frac{\text{tamanho do bloco}}{\text{taxa de codificação}}$ garante uma exibição contínua [13]. Uma propriedade do leiaute por faixas de grão-grosso é o fato dos clientes e da banda disponível circularem através dos discos do servidor: dado que no ciclo atual o cliente recebeu um bloco do disco d_i , no próximo ciclo ele receberá um bloco do disco d_{i+1} no caso de acesso seqüencial aos dados. Requisições chegando num dado momento precisam esperar até o início do próximo ciclo para serem servidas, mesmo que o disco onde está o bloco tenha banda disponível para servi-lo. Se, no próximo ciclo, não houver banda disponível no disco onde está o bloco a ser recuperado, um cliente pode esperar mais de um ciclo até que a banda disponível chegue no disco com o bloco requisitado. Se o bloco requisitado está no disco d_r e apenas o disco d_b possui banda disponível, então o cliente precisa esperar $r - b \bmod D$ ciclos, onde D é a quantidade de discos no servidor, para receber o bloco.

Staggered striping [1] é uma generalização do leiaute por faixas. No *staggered striping* os blocos de uma mídia podem se estender por mais de um disco e o próximo bloco pode estar a qualquer distância, chamada passo, do bloco corrente¹. Estas modificações na alocação em disco, juntas de *buffers* em memória, habilitam o *staggered striping* a servir mais fluxos de mídia do que o leiaute por faixas original, em cenários onde as mídias possuem taxa de codificação superior à banda efetiva de um disco. O *staggered striping* também suporta mídias com diferentes taxas de codificação.

Como o *staggered striping* trata de vídeos com alta taxa de codificação, o que não é o assunto deste estudo, nós analisamos apenas o leiaute por faixas original. Além disto, como [13] mostrou que o leiaute por faixas de grão-grosso possui desempenho superior ao leiaute por faixas de grão-fino, nós avaliamos o desempenho apenas do primeiro.

¹Se o servidor opera com passos de tamanho p e o bloco b_i começa no disco d_k , então o bloco b_{i+1} começa no disco $d_{k+p \bmod D}$, onde D é o total de discos no servidor.

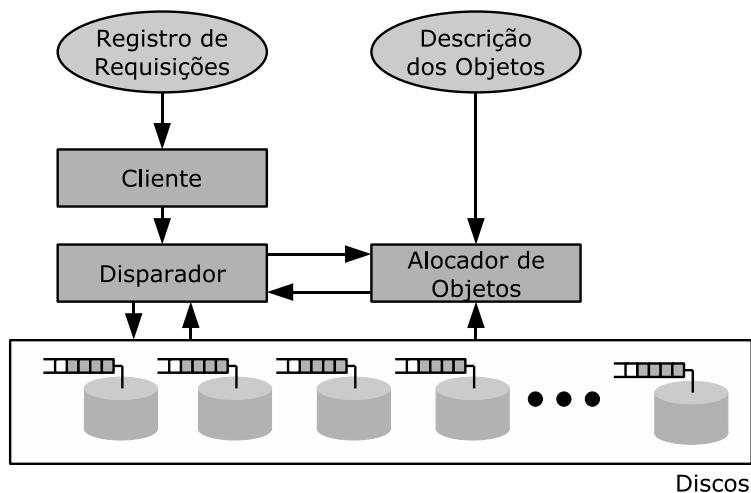


Figura 1: Modelo do Sistema de Armazenamento de Mídia.

A alocação aleatória foi inicialmente proposta em [7]. Objetos de mídia são novamente divididos em pequenos blocos. Cada bloco de mídia é colocado em um disco aleatório, réplicas de um bloco são colocadas em outros discos e são usadas para melhor balancear a carga através dos discos. Ao contrário do leiaute por faixas, a alocação aleatória não assume nenhum padrão de acesso aos dados e conseqüentemente é mais apropriada para aplicações que não fazem acesso seqüencial aos dados, como mundos virtuais. Comparações prévias entre o leiaute aleatório e o leiaute por faixas [12] mostraram que, com replicação homogênea de dados, o leiaute aleatório é competitivo com o leiaute por faixas. Este trabalho vai além daquele, pois nós experimentamos com cargas realistas e avaliamos o desempenho de diferentes métodos de replicação. [11] mostra que o leiaute aleatório é eficiente mesmo usando discos heterogêneos, através do uso de replicação de dados para balancear a carga no disco proporcionalmente à sua banda efetiva.

[10] compara o leiaute aleatório e o leiaute por faixas num serviço distribuído de mídia com até 8 servidores, seus resultados mostram que os leiautes têm desempenho competitivo quanto ao número de clientes atendidos, mas o leiaute por faixas possui latência superior. Eles analisam o impacto da quantidade de *buffers* no cliente, chegando à conclusão que os ganhos são pequenos para mais do que dois blocos de *buffer*. Por último eles analisam algumas otimizações possíveis num cenário onde clientes utilizam seu espaço de armazenamento e banda para transmitir vídeos para seus vizinhos.

Uma abordagem diferente ao problema de alocação de dados é tomada em [5]. Dados um conjunto de objetos de tamanho (em bytes) e quantidade conhecida de clientes requisitando-os, eles estudam o problema de maximizar a quantidade de clientes atendidos sujeitos à capacidade de armazenamento dos discos. Eles apresentam um algoritmo de alocação com limites teóricos de desempenho.

3. Modelo do Sistema de Armazenamento de Mídia

Esta seção descreve o modelo do sistema de armazenamento que construímos para avaliar o desempenho do leiaute por faixas, aleatório e seqüencial para várias cargas realistas atu-

almente. O modelo consiste de quatro componentes principais. Um *cliente*, um *alocador de objetos*, um *disparador de requisições* e uma quantidade de *discos*, com banda e espaço limitados. Suas principais entradas são o *tamanho de bloco em disco*, a *descrição dos objetos*, com o tamanho e taxa de codificação de cada arquivo, e um *registro de requisições*, especificando o momento de chegada, arquivo requisitado e posições de início e fim dentro do arquivo.

O alocador de objetos é responsável por alocar todos os objetos de mídia nos discos do sistema, de acordo com a descrição dos objetos e tamanho de bloco bem como o leiaute escolhido. Duas variações da alocação seqüencial são consideradas. Uma, à qual nos referimos como leiaute seqüencial, simplesmente coloca cada arquivo inteiro em um disco escolhido aleatoriamente, desde que ele tenha espaço livre. A outra variação também coloca os arquivos inteiros em apenas um disco, mas seleciona os discos de modo a balancear a carga total (ex: quantidade de bytes recuperados) nos discos. Esta variação é chamada de alocação seqüencial com balanceamento de carga.

No caso específico do leiaute aleatório, nós consideramos três estratégias de replicação de dados. Na replicação homogênea [12], o alocador de objetos aleatoriamente seleciona uma fração fixa dos blocos de cada arquivo para serem replicados e aloca-os aleatoriamente através dos discos. Os outros dois esquemas são novos. Dada uma quantidade fixa de espaço disponível para replicação, eles criam uma quantidade de réplicas proporcional à popularidade do conteúdo. Na replicação por popularidade do arquivo, a fração de blocos do arquivo selecionada para replicação é proporcional à popularidade do arquivo (ex: quantidade de bytes requisitados). A replicação por popularidade do bloco cria uma quantidade de réplicas para cada bloco que é proporcional à quantidade de requisições disparadas para ele, e assim, pode apresentar melhor desempenho para cargas interativas, onde a popularidade dos blocos muda significativamente dentro do mesmo arquivo.

O módulo cliente lê o registro da carga e dispara uma série de requisições para o disparador. No leiaute por faixas, requisições de clientes são para segmentos de dados. É responsabilidade do disparador sincronizar a recuperação de cada bloco de dado, dentro do segmento requisitado, dos discos e enviar estes blocos para o cliente na taxa de exibição do arquivo [13]. Nos leiautes aleatório e seqüencial a sincronização é feita pelo cliente, como descrito em [7]. Dois *buffers* são usados para cada requisição do cliente: um deles para recebimento e outro para exibição. Quando a exibição de um bloco termina, os *buffers* trocam de papel e o cliente faz uma nova requisição de bloco para o disparador.

Para cada requisição recebida do cliente, o disparador pergunta ao alocador de objetos a posição dos blocos requisitados e dispara requisições de bloco para os devidos discos. No leiaute por faixas, o disparador funciona em ciclos [13]. No início de cada ciclo, ele dispara uma nova requisição de bloco para cada requisição de cliente pendente. A duração de um ciclo é calculada como descrita na seção 2, usando a maior taxa de codificação em casos de cargas com diferentes taxas de codificação. No leiaute aleatório, se existe replicação, o disparador envia a requisição de bloco para o disco que possui a menor fila.

Uma requisição de bloco enviada para um disco entra na fila do disco. A política de escalonamento da fila do disco pode ser tanto Primeiro a Chegar Primeiro a Sair (PCPS) ou B-SCAN, que ordena as requisições pela posição do cilindro na superfície do disco, mi-

nimizando o tempo de *seek* por mover a cabeça em apenas uma direção [12]. O tempo de serviço de uma requisição de bloco é calculado como a soma do tempo de *seek*, da latência rotacional e do tempo de transferência. O tempo de *seek* é computado dinamicamente como segue: se a cabeça de leitura precisa se mover x cilindros para alcançar o dado requisitado, o tempo de *seek* é dado por $\frac{x-1}{cilindros-1} \times (seek\ max - seek\ min) + seek\ min$, onde *cilindros* é a quantidade de cilindros no disco, *seek min* é o tempo de movimentação da cabeça entre dois cilindros adjacentes e *seek max* é o tempo de passar por todos os cilindros (ex: de um lado a outro do disco). Assumimos que a latência rotacional é uniformemente distribuída dentro do período de uma rotação. Finalmente, o tempo de transferência é calculado como a razão do tamanho da requisição e da taxa média de transferência do disco. Foi mostrado que este modelo do tempo de serviço de um disco subestima o tempo médio de resposta em apenas 15% em [9].

Outros parâmetros de disco são a quantidade de *cabeças*, *trilhas*, *setores por trilha* e *bytes por setor*. Estes parâmetros definem a capacidade total de armazenamento do disco. A figura 1 mostra os principais componentes do nosso modelo do sistema de armazenamento de mídia.

4. Ambiente de Simulação

Nós construímos um simulador detalhado do modelo de sistema de armazenamento apresentado na seção 3. O simulador é dirigido por eventos, disparado pela chegada de novas requisições de clientes (do registro de requisições). Uma descrição dos parâmetros do simulador e sua validação é apresentada na seção 4.1. A seção 4.2 introduz as cargas realistas usadas no nosso estudo.

4.1. Simulador do Sistema de Armazenamento

Nós parametrizamos nosso simulador com blocos de tamanho variando desde 128 KB até 512 KB, mas mostramos resultados apenas para 512 KB, a não ser que explicitamente dito o contrário, pois os resultados são qualitativamente os mesmos. A quantidade de replicação de dados é variada de 0 até 25% da quantidade total de dados armazenados, quando não explicitado a quantidade de replicação usamos o valor de 10% como padrão.

Os discos são configurados com parâmetros do Hitachi Ultrastar 36Z15, mostrados na tabela 1. O número de cilindros no disco é a razão do número de trilhas pelo número de cabeças. Nós experimentamos com ambas, PCPS e B-SCAN, políticas de escalonamento da fila do disco mas mostramos resultados apenas para a última, pois PCPS é significativamente menos eficiente, independente do leiaute utilizado. A quantidade de discos utilizada foi variada de 2 até 4, mas mostramos resultados para quatro discos, a não ser quando explicitamente dito o contrário.

Nosso simulador foi validado através de uma comparação do número de clientes atendidos simultaneamente em função do tamanho do bloco, para o leiaute por faixas e leiaute aleatório. Usamos uma carga tradicional com 100 arquivos com duração entre 90 e 120 minutos, codificados a 1,5 Mbit/s. A popularidade de um arquivo é dada por uma distribuição Zipf [14] com parâmetro $\alpha = 1$. Requisições dos clientes são para arquivos completos. Nossos resultados, mostrados na figura 2, são consistentes com [12] e mostram que o leiaute por faixas consegue atender 11% mais clientes que o leiaute aleatório para blocos de 512KB, num servidor com 4 discos.

Número de bytes por setor	512
Número de setores por trilha	222
Número de trilhas	27000
Número de cabeças	12
Período rotacional	4 ms
Taxa média de transferência	44,7 MB/s
Tempo mínimo de <i>seek</i>	0,65 ms
Tempo máximo de <i>seek</i>	8,9 ms

Tabela 1: Parâmetros do Hitachi Ultrastar 36Z15.

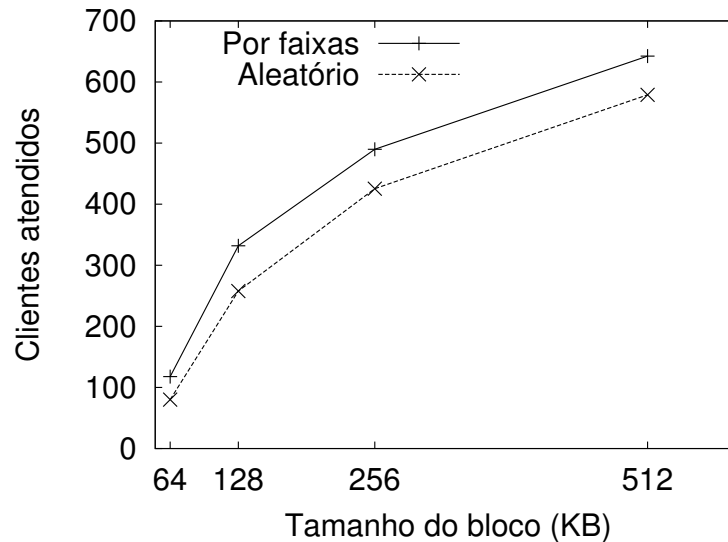


Figura 2: Validação do simulador.

4.2. Cargas Realistas

As cargas usadas em nossas simulações, geradas com o GENIUS [3], capturam a interatividade e heterogeneidade encontradas em cargas reais [2]. Nós experimentamos com dois diferentes tipos de cargas:

Vídeo Educacional: 1000 vídeos com duração entre 5 e 50 minutos, codificados a 300 Kbits/s. 25% das sessões para arquivos grandes possuem mais do que 10 requisições. A duração média de uma requisição é 15 minutos. A popularidade dos arquivos é dada pela concatenação de duas distribuições Zipf [2].

Vídeo de Entretenimento: 10000 vídeos, a maioria com menos de 5 minutos e codificados a 50Kbit/s ou 100Kbit/s. Há um grau médio de interatividade, com 15% das sessões tendo mais do que uma requisição. A popularidade do arquivo é dada por uma simples distribuição Zipf com parâmetro α variando entre 0,75 e 1,23.

Estas características foram observadas na carga de um servidor de mídia real que transmitia conteúdo educacional numa universidade nos EUA e numa carga de entretenimento num dos maiores provedores de conteúdo da América Latina. Uma descrição mais detalhada das cargas pode ser encontrada em [2].

5. Resultados

Esta seção apresenta os resultados mais relevantes da nossa avaliação do leiaute por faixas, aleatório e seqüencial, para cargas realistas. O desempenho destes leiautes foi analisado com relação a duas métricas: latência inicial média e número máximo de clientes atendidos simultaneamente.

A latência inicial é medida como sendo o tempo necessário para que o sistema envie ao cliente os blocos necessários para que ele inicie a exibição de seu vídeo e, desta forma, está diretamente ligada com a qualidade de serviço (QoS) percebida pelo cliente. Note que, no leiaute por faixas, a latência do cliente depende da duração do ciclo, o que, por sua vez, depende do tamanho do bloco e da taxa de codificação da mídia (veja seção 2). Requisições chegando num dado momento precisam esperar até o início do próximo ciclo para serem servidas, mesmo que o disco onde está o bloco tenha banda disponível para servi-lo. Se, no próximo ciclo, não houver banda disponível no disco onde está o bloco a ser recuperado, um cliente pode esperar mais de um ciclo até que a banda disponível chegue no disco com o bloco requisitado. Em nossa análise nós favorecemos o leiaute por faixas, desconsiderando este tempo de espera até que a banda disponível circule até o disco com o bloco requisitado. Ao contrário do leiaute por faixas, na alocação aleatória e seqüencial, novas requisições para blocos podem ser disparadas assim que a requisição é feita pelo cliente.

Nos leiaute aleatório e seqüencial, quando a carga no sistema de discos aumenta, a recuperação de blocos de dados pode sofrer atrasos. Alguns blocos podem chegar no cliente depois do momento que deveriam ser tocados, comprometendo o serviço. No leiaute por faixas, sobrecarga do sistema acontece quando o tempo necessário para recuperar todos os blocos pendentes das requisições dos clientes é maior que o tempo do ciclo. Nós usamos uma abordagem probabilística para determinar a quantidade máxima de clientes atendidos pelos leiautes. Para os leiautes aleatório e seqüencial, nós assumimos que uma dada quantidade de clientes, de uma carga, foi atendida com sucesso se a probabilidade de um bloco chegar atrasado é menor que 10^{-6} . Para que tenhamos uma comparação justa, nós dizemos que o leiaute por faixas também pode suportar uma dada quantidade de clientes, de uma carga, se a probabilidade de um ciclo não recuperar todos os blocos pendentes for menor do que a mesma fração. Esta abordagem é a mesma que a usada em [12] e favorece o leiaute por faixas, pois independente da quantidade de blocos atrasados num ciclo, consideramos que ocorreu apenas um atraso.

5.1. Carga educacional

A figura 3 mostra a latência inicial média em função do número de clientes servidos no leiaute por faixas, aleatório e seqüencial. As curvas são obtidas variando a taxa de chegada de sessões de 1 a 11 sessões por segundo. Note que, para o leiaute por faixas, cada ponto representa a latência *mínima* que o leiaute pode prover para servir o dado número de clientes. Isto significa que cada ponto é obtido com o tamanho de bloco (anotado ao lado de cada ponto) que minimiza a latência para aquela dada quantidade de clientes. A tabela 2 mostra a quantidade máxima de clientes que o sistema conseguiu suportar para cada um dos leiautes, utilizando 4 discos e blocos de 512 KB.

Com relação à quantidade máxima de clientes atendidos, o leiaute por faixas serve 7% mais clientes que o leiaute aleatório e 40% mais clientes que o leiaute seqüencial, para

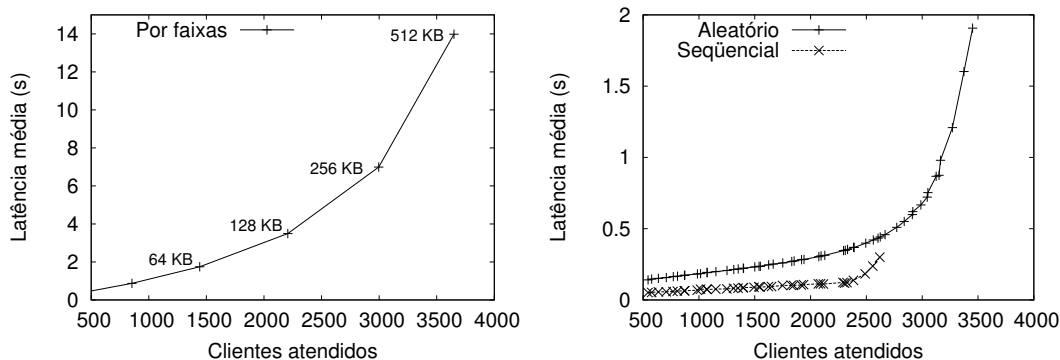


Figura 3: Latência inicial X número de clientes atendidos: carga educacional.

Leiaute	Máx. Clientes
Por faixas	3690
Aleatório	3450
Seqüencial	2620

Tabela 2: Quantidade máxima de clientes atendidos: carga educacional.

a carga educacional. Porém, este ganho vem com um custo de uma latência inicial muito alta (ex: mais de 5 vezes maior). Esta é uma latência proibitiva, especialmente para cargas interativas, o que foi visto ser comum para cargas educacionais [2]. Se mantermos a latência inicial abaixo de 1 segundo, o leiaute por faixas atende uma quantidade máxima de clientes que é 65% e 58% menor do que o leiaute aleatório e seqüencial, respectivamente. Assim, mesmo o leiaute seqüencial, que sofre de sérios problemas de balanceamento de carga nos discos (veja abaixo), apresenta melhor desempenho do que o leiaute por faixas.

Comparando os leiautes aleatório e seqüencial, é interessante notar que o número máximo de clientes atendidos pelo leiaute aleatório é 31% maior, enquanto, se fixarmos a quantidade de clientes atendidos, o leiaute seqüencial apresenta uma latência que é até 65% menor do que a do leiaute aleatório. Isto acontece porque os dois primeiros blocos requisitados, para preenchimento do *buffer*, pelo cliente antes de começar a exibição são recuperados seqüencialmente com apenas um acesso ao disco no leiaute seqüencial, enquanto que no leiaute aleatório a recuperação de dois blocos envolve dois acessos a disco. Desta forma, o simples leiaute seqüencial tem desempenho muito competitivo, especialmente para cargas altamente interativas para as quais uma baixa latência é necessária, para cargas médias e baixas no servidor.

Para melhor compreender o desempenho de cada um dos leiautes, as figuras 4-a, 4-b e 4-c mostram a carga normalizada nos discos típica para o leiaute por faixas, aleatório e seqüencial, respectivamente. Nós medimos a carga normalizada em cada disco como sendo a quantidade de bytes servidos por ele dividido pela quantidade de bytes servidos pelo disco com a maior carga. O leiaute por faixas provê o melhor balanceamento de carga, como esperado. O leiaute seqüencial apresenta a pior distribuição de carga através dos discos. O disco com menor carga apresenta uma carga que é 20% menor do que a do

disco com maior carga. Este desbalanceamento de carga é sinônimo de banda de disco desperdiçada, o que leva a uma menor quantidade máxima de clientes atendidos.

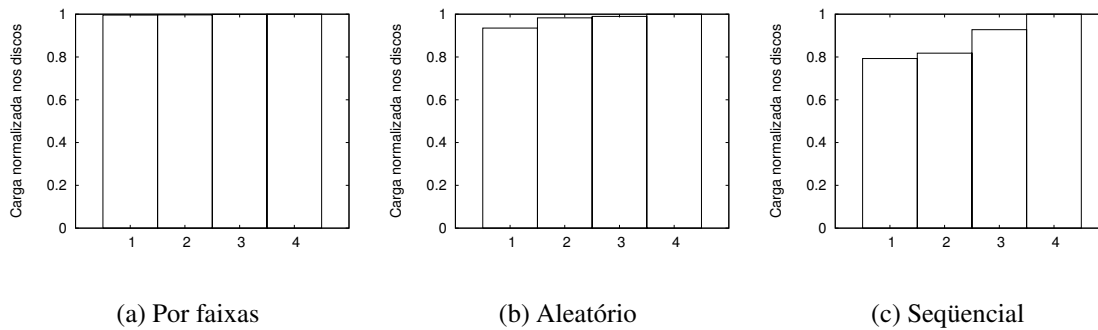


Figura 4: Carga normalizada nos discos: carga educacional.

5.2. Carga de entretenimento

A figura 5 mostra a latência inicial média como função do número de clientes atendidos para os três leiautes analisados. Novamente, para o leiaute por faixas, mostramos a latência *mínima* que o leiaute pode prover para a dada quantidade de clientes. As curvas foram obtidas variando a taxa de chegada de sessões entre 10 e 280 sessões por segundo. A tabela 3 mostra a quantidade máxima de clientes atendidos por um servidor de 4 discos e blocos de 512 KB.

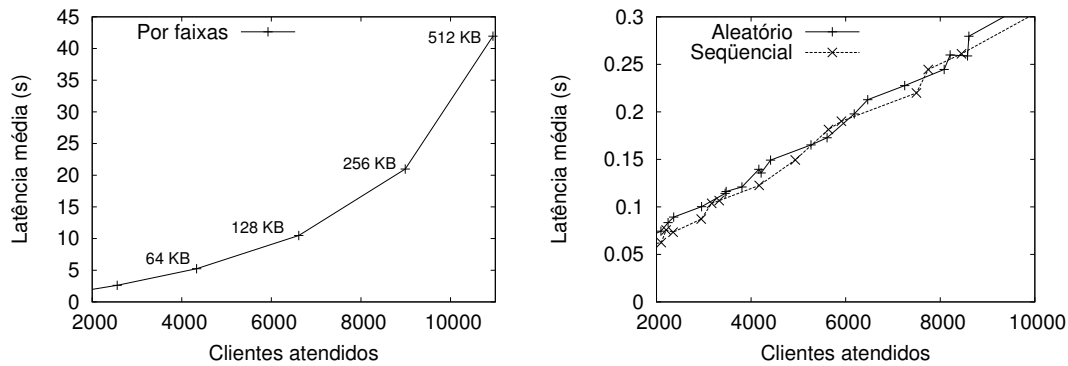


Figura 5: Latência inicial X número de clientes atendidos: carga de entretenimento.

Leiaute	Máx. Clientes
Por faixas	11096
Aleatório	14305
Seqüencial	10910

Tabela 3: Quantidade máxima de clientes atendidos: carga de entretenimento.

Para estas cargas de entretenimento, com arquivos cuja taxa de codificação é ainda menor que no caso educacional, a latência para o leiaute por faixas aumenta ainda mais.

O leiaute seqüencial ainda apresenta uma latência até 12% menor que o leiaute aleatório. O ganho reduzido de latência neste cenário, quando comparado ao cenário educacional, deve-se ao fato de que muitas requisições para a carga de entretenimento são menores que um bloco, fazendo que ambos leiautes aleatório e seqüencial recuperem os dados com apenas um acesso a disco. Novamente, para cargas leves e médias no servidor, o leiaute seqüencial é uma alternativa competitiva.

Com relação à quantidade de clientes atendidos, o leiaute aleatório consegue servir 29% mais clientes que o leiaute por faixas e 31% mais clientes que o leiaute seqüencial. Isto deve-se ao fato do leiaute por faixas ser ineficiente para cargas onde os vídeos possuem diferentes taxas de codificação e ao fato do leiaute seqüencial apresentar grande desbalanceamento de carga nos discos, como mostrado na figura 6-c.

A figura 6 mostra o desbalanceamento de carga nos discos para os três leiautes. Até o leiaute por faixas apresenta um considerável desbalanceamento de carga nos discos, isto deve-se a duas características da carga: primeiramente uma distribuição de popularidade dos arquivos muito mais desbalanceada e arquivos muito pequenos que cabem em menos discos do que os disponíveis, impossibilitando a distribuição da carga através de todos os discos.

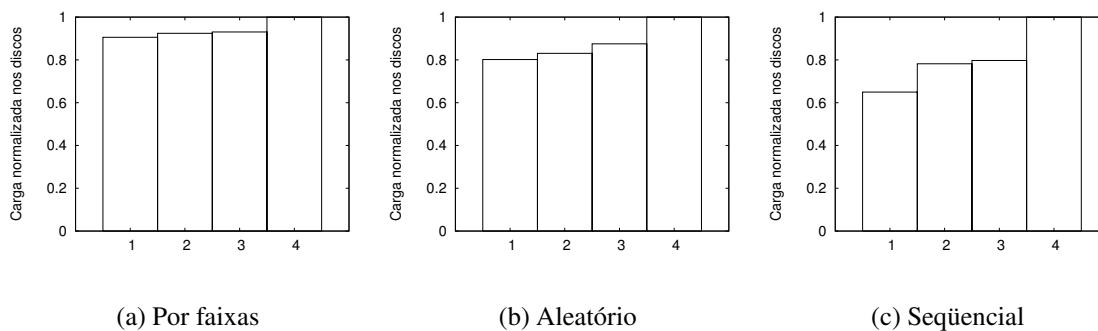


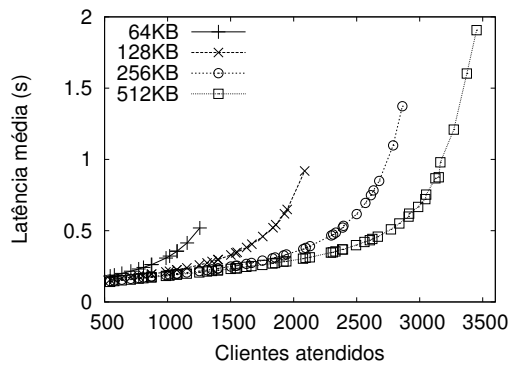
Figura 6: Carga normalizada nos discos: carga de entretenimento.

5.3. Impacto do tamanho do bloco

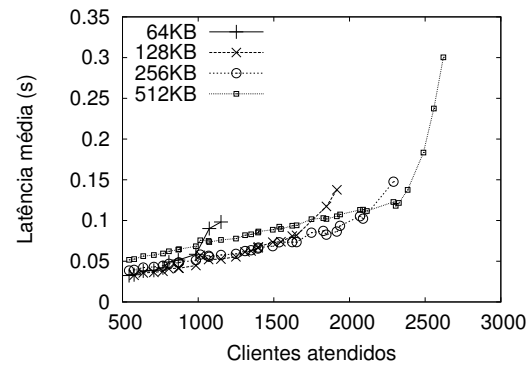
As figuras 7-a e 7-b mostram o impacto do tamanho do bloco na latência inicial média de serviço para os leiautes aleatório e seqüencial, respectivamente, para cargas educacionais e servidores com 4 discos. Resultados para a carga de entretenimento são qualitativamente iguais. A tabela 4 mostra a quantidade máxima de clientes atendidos para diferentes tamanhos de bloco.

Leiaute	Tamanho do bloco			
	64KB	128KB	256KB	512KB
Por faixas	1440	2205	2995	3690
Aleatório	1255	2085	2860	3450
Seqüencial	1150	1915	2290	2620

Tabela 4: Quantidade máxima de clientes atendidos para diferentes tamanhos de bloco.



(a) Aleatório



(b) Seqüencial

Figura 7: Latência inicial X número de clientes atendidos: diferentes tamanhos de blocos.

Em todos os leiautes, aumentar o tamanho do bloco resulta numa maior quantidade de clientes atendidos. Isto deve-se ao fato de uma maior eficiência na utilização dos discos, pois blocos maiores implicam menos operações de *seek* e menos atrasos devido à latência rotacional.

A latência no leiaute aleatório diminui com o aumento do tamanho do bloco, pois a eficiência do disco aumenta, mas o comportamento segue o mesmo padrão para qualquer tamanho de bloco. Em contrapartida, no leiaute seqüencial blocos maiores resultam em um perceptível (até 25%) aumento na latência inicial. No leiaute aleatório a recuperação dos dois primeiros blocos necessários para preencher o buffer do cliente é independente e sofre menos impacto do tempo de transferência do bloco do disco do que no leiaute seqüencial, que recupera os dois blocos com apenas um acesso seqüencial ao disco. Para o leiaute por faixas, a latência inicial média para os diferentes tamanhos de bloco pode ser vista nas figuras 3 e 5: blocos menores diminuem a duração de um ciclo, implicando uma redução na latência inicial.

5.4. Impacto da quantidade de discos

As figuras 8-a e 8-b mostram o impacto da quantidade de discos na latência inicial do serviço para os leiautes aleatório e seqüencial, respectivamente, para a carga educacional e blocos de 512 KB. Os resultados para as cargas de entretenimento são qualitativamente iguais. A tabela 5 mostra a quantidade máxima de clientes atendidos para cada um dos leiautes para diferentes quantidades de discos.

Leiaute	2 discos	4 discos
Por faixas	1850	3690
Aleatório	1685	3450
Seqüencial	1495	2620

Tabela 5: Quantidade máxima de clientes atendidos para diferentes quantidades de discos.

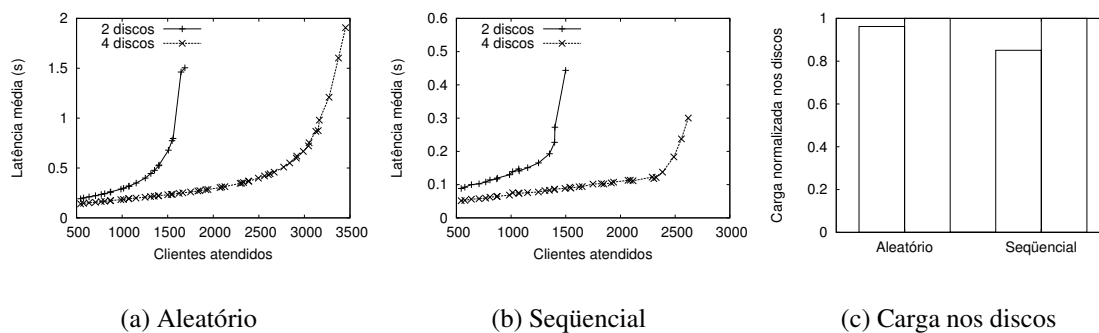


Figura 8: Impacto da quantidade de discos no desempenho dos leiautes aleatório e seqüencial.

O aumento do número de discos no servidor resulta numa maior quantidade de clientes atendidos: 100%, 104% e 75% mais clientes para os leiautes por faixas, aleatório e seqüencial, respectivamente. Além deste ganho esperado, vemos que a latência inicial do serviço diminui quando uma maior quantidade de discos é utilizada para os leiautes aleatório e seqüencial, isto deve-se à diminuição do tamanho médio das filas dos discos, mostrado na figura 9. O leiaute por faixas apresenta a mesma latência para qualquer quantidade de discos: o aumento na quantidade de discos diminui o tamanho médio das filas dos discos, mas isto traz ganhos desprezíveis visto que a latência imposta pela duração do ciclo é ordens de grandeza maior que o atraso da fila. Note que uma maior quantidade de discos no leiaute por faixas aumenta a quantidade média de ciclos que um cliente precisa esperar até que a banda disponível chegue ao disco onde está o bloco requisitado, o que causa um aumento da latência. Nós desconsideramos este efeito, favorecendo o leiaute por faixas.

É importante notar que para apenas dois discos no servidor, o leiaute seqüencial consegue atender uma quantidade de clientes que é apenas 11% menor que a atendida pelo leiaute aleatório. Isto deve-se ao fato de que o desbalanceamento relativo de carga num servidor com 2 discos é menor do que para quatro discos, como pode-se perceber comparando a figura 4 e a figura 8-c. Este mesmo fato explica porque o leiaute seqüencial tem um aumento de apenas 75% na quantidade de clientes atendidos quando passamos de 2 para 4 discos no servidor.

5.5. Impacto da replicação de dados no leiaute aleatório

Para minimizar a degradação de desempenho devido ao desbalanceamento de carga nos discos, o leiaute aleatório pode usar replicação de dados. A figura 10-a mostra a latência inicial como função da quantidade de clientes atendidos para diferentes quantidades de replicação de dados no cenário educacional, utilizando-se replicação homogênea. As linhas são muito próximas, indicando o pequeno impacto da replicação neste cenário, dado que a carga é bem balanceada nos discos mesmo sem replicação. A figura 10-b mostra a latência inicial como função da quantidade de clientes atendidos para os diferentes métodos de replicação, com replicação de 10% dos dados, para a carga de entretenimento. A tabela 6 mostra a quantidade máxima de clientes atendidos utilizando-se 10% de replicação de dados.

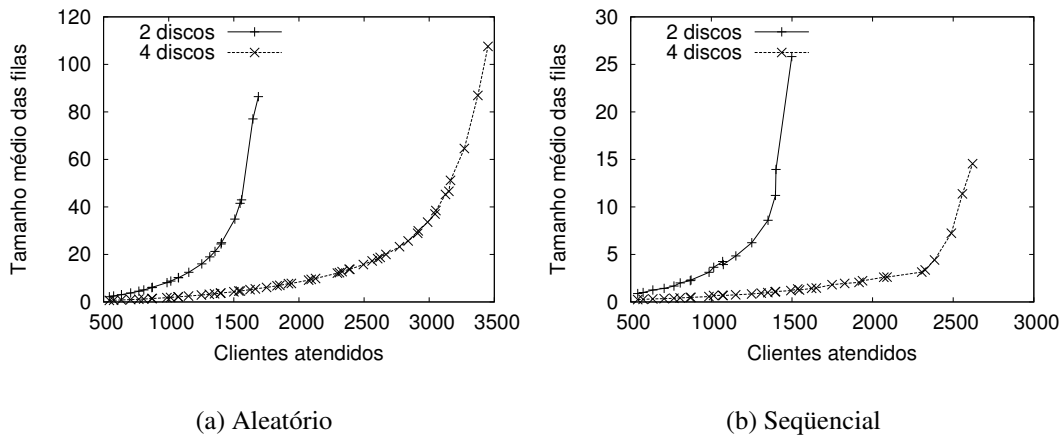


Figura 9: Impacto da quantidade de discos no tamanho médio das filas dos discos.

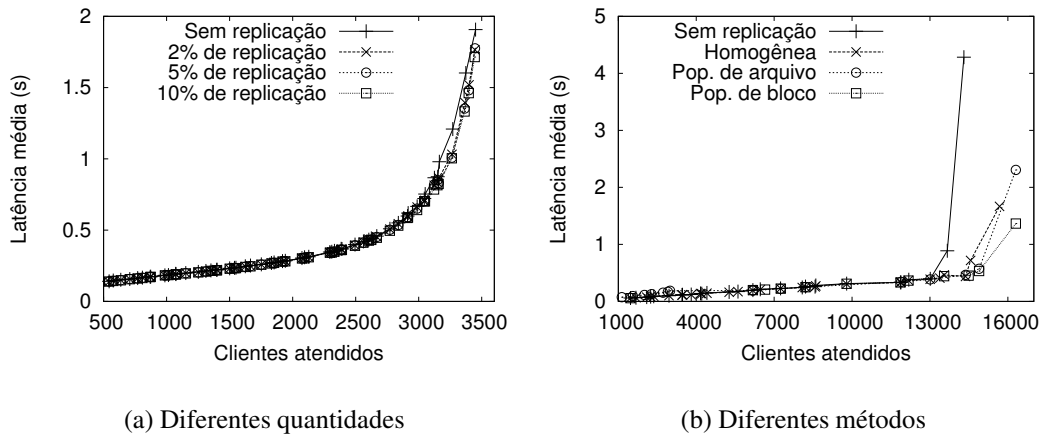


Figura 10: Impacto do uso de replicação de dados no leiaute aleatório.

Na carga educacional, a replicação de dados aumenta a quantidade máxima de clientes atendidos em somente 2%, já que a carga nos discos já está relativamente balanceada entre os discos (veja a figura 4-b). Já para o cenário de entretenimento, onde o desbalanceamento de carga é maior, o aumento na quantidade de clientes atendidos é de até 14%.

Além disto, em termos da quantidade máxima de clientes atendidos, a replicação de dados baseada na popularidade do conteúdo não apresenta nenhuma melhora significativa na quantidade de clientes atendidos, a não ser que a fração de dados replicados seja muito pequena (menos do que 5%). Isto deve-se ao fato do conteúdo mais popular acabar sendo replicado em qualquer um dos três métodos, sendo isto suficiente para atingir o balanceamento da carga nos discos. A figura 11-a mostra a carga normalizada nos discos para replicação homogênea de 2% dos dados para a carga educacional e a figura 11-b mostra a carga normalizada nos discos para replicação de 10% dos dados por popularidade de bloco para a carga de entretenimento. Além disto, replicar mais do que 10% dos

Método de replicação	Cenário	
	Educacional	Entretenimento
Nenhum	3450	14305
Homogêneo	3510	15675
Pop. do arquivo	3510	16305
Pop. do bloco	3510	16305

Tabela 6: Quantidade máxima de clientes atendidos para diferentes métodos de replicação.

dados não traz ganhos significativos.

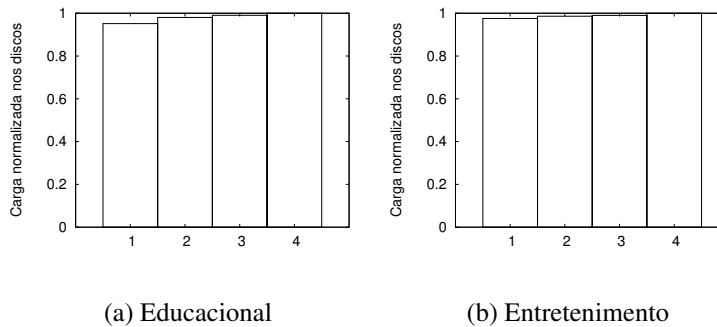


Figura 11: Carga normalizada nos discos com utilização de replicação.

Com relação à latência inicial do serviço nos casos onde a latência provida é similar para os três métodos, reduções de 2%, 8% e 9% foram observadas para a replicação homogênea, replicação baseada na popularidade do arquivo e replicação baseada na popularidade do bloco, respectivamente. Assim, a replicação de dados pode ser utilizada para prover limitada melhora na latência inicial do serviço e ganhos consideráveis na quantidade máxima de clientes atendidos para a carga de entretenimento.

5.6. Impacto do balanceamento de carga no leiaute seqüencial

Assim como o leiaute aleatório, o leiaute seqüencial pode fazer balanceamento de carga através dos discos como explicado na seção 3, para melhorar seu desempenho. A tabela 7 mostra que, para a carga educacional, o número máximo de clientes atendidos aumenta em 29%. Para a carga de entretenimento o aumento é de 33%, o que deve-se à eliminação do grande desbalanceamento apresentado nas figuras 4-c e 6-c.

Carga	Sem balanceamento	Com balanceamento
Educacional	2620	3390
Entretenimento	10910	14540

Tabela 7: Quantidade máxima de clientes atendidos com balanceamento de carga no leiaute seqüencial.

As figuras 12-a e 12-b mostram a latência inicial média em função da quantidade de clientes atendidos, para a carga educacional e de entretenimento, respectivamente.

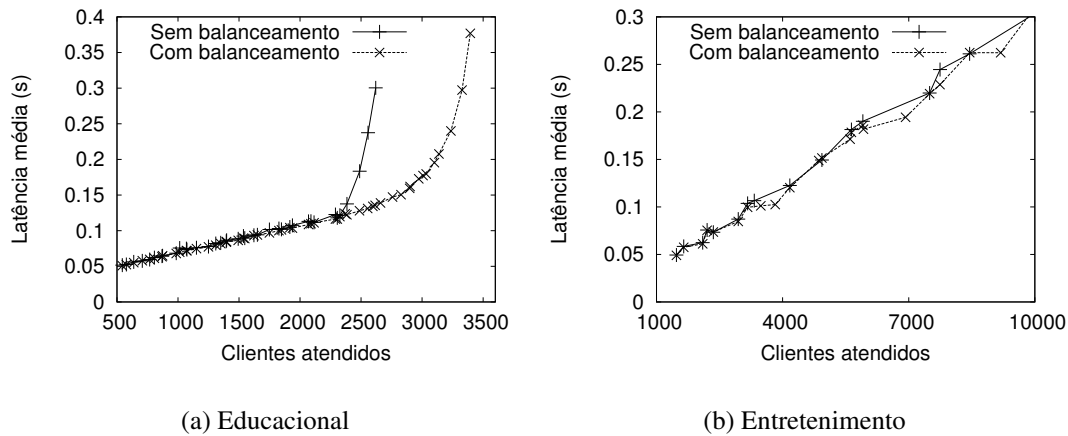


Figura 12: Impacto do uso de balanceamento de carga no leiaute seqüencial.

Estas figuras mostram que os ganhos na latência inicial são pouco significativos (menores que 5%), para a maior parte do número de clientes atendidos.

Assim, o leiaute seqüencial com balanceamento de carga é capaz de atender a mesma quantidade de clientes do que o leiaute aleatório (menos de 1% de diferença), com latência menor. Nós salientamos que estes resultados são obtidos assumindo que a popularidade do arquivo é conhecida de antemão. Como trabalho futuro, temos a intenção de analisar a sensibilidade deste mecanismo para diferentes estimativas da popularidade dos arquivos.

6. Conclusões e trabalhos futuros

Nós apresentamos uma comparação quantitativa entre o leiaute por faixas, o leiaute aleatório e a alocação seqüencial de dados, focando em cenários realistas. Analisamos a quantidade máxima de clientes atendidos simultaneamente e a latência inicial média. Experimentamos com dois tipos distintos de cargas: educacional e entretenimento.

Para a carga educacional, nossos resultados mostram que o leiaute por faixas atende mais clientes do que o leiaute aleatório e a alocação seqüencial. No cenário de entretenimento, o leiaute aleatório atende mais clientes do que qualquer um dos outros leiautes. A alocação seqüencial atende menos clientes em ambos cenários, um resultado do desbalanceamento de carga nos discos. A latência média para o leiaute por faixas é ordens de magnitude maior do que a latência para os outros leiautes, para nossas cargas com baixa taxa de codificação. Além disto, mostramos que a alocação seqüencial apresenta latência inicial média competitiva quando a utilização do servidor é média ou baixa. Replicação de dados no leiaute aleatório e balanceamento de carga na alocação seqüencial aumentam a quantidade máxima de clientes atendidos, mas têm impacto limitado na latência inicial média. Estes resultados mostram que o leiaute aleatório é robusto a variações nas características da carga e configurações do servidor e que a alocação seqüencial é uma alternativa competitiva se for capaz de atender a quantidade de clientes requisitando serviço.

Possíveis direções para trabalhos futuros incluem uma comparação mais extensiva

usando diferentes cargas e configurações do servidor, comparar outros leiautes como o *staggered striping* e analisar cenários específicos como configurações com discos heterogêneos.

Referências

- [1] S. Berson, S. Ghandeharizadeh, R. Muntz, and X. Ju. Staggered striping in multimedia information systems. In *Proc. ACM SIGMOD*, Minneapolis, MN, May 1994.
- [2] C. Costa, Í. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro-Neto. Analyzing Client Interactivity in Streaming Media. In *Proc. 13th WWW Conference*, New York, NY, May 2004.
- [3] C. Costa, C. Ramos, Í. Cunha, and J. Almeida. Genius: Generator of interactive user media sessions. In *Proc. Workshop on Workload Characterization*, Austin, TX, October 2004.
- [4] D. L. Eager, M. K. Vernon, and J. Zahorjan. Bandwidth Skimming: A Technique for Cost-Effective Video-on-Demand. In *Proc. Multimedia Computing and Networking*, San Jose, CA, January 2000.
- [5] L. Golubchik, S. Khanna, S. Khuller, R. Thurimella, and A. Zhu. Approximation algorithms for data placement on parallel disks. In *Proc. Symposium on Discrete Algorithms*, pages 223–232, 2000.
- [6] M. Li, M. Claypool, R. Kinicki, and J. Nichols. Characteristics of streaming media stored on the web. *accepted to ACM Transactions on Internet Technology*, 5(4), November 2005.
- [7] R. Muntz, J. Santos, and S. Berson. A parallel disk storage system for real-time multimedia applications. *Int'l Journal Intelligent Systems, Special Issue on Multimedia Computing System*, 13(12):1137–74, December 1998.
- [8] M. Reisslein, K. Ross, and S. Shrestha. Striping for interactive video: Is it worth it? In *Proc. Int'l Conf. on Multimedia Computing and Systems*, Florence, Italy, June 1999.
- [9] Chris Ruemmler and John Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3):17–28, 1994.
- [10] D. Santos, A. Borges, B. Ribeiro-Neto, and S. Campos. Performance analysis and optimization of a distributed video on demand service. In *Proc. Int'l Symposium on Performance Analysis of Systems and Software*, Austin, TX, March 2003.
- [11] J. Santos and R. Muntz. Performance analysis of the RIO multimedia storage system with heterogeneous disk configurations. In *Proc. ACM Multimedia*, Bristol, UK, 1998.
- [12] J. Santos, R. Muntz, and B. Ribeiro-Neto. Comparing random data allocation and data striping in multimedia servers. In *Proc. ACM SIGMETRICS*, Santa Clara, CA, June 2000.
- [13] B. Özden, R. Rastogi, and A. Silberschatz. Disk striping in video server environments. In *Proc. Int'l Conf. on Multimedia Computing and Systems*, Hiroshima, Japan, June 1996.
- [14] G. K. Zipf. *Human Behavior and the Principle of Least-Effort*. Addison-Wesley, 1949.