

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

**Transferência Sob Demanda de Mídias Não Lineares para
Dispositivos Móveis**

ANDERSON TADEU BRAGIATTO

Florianópolis
2014

Anderson Tadeu Bragiatto

**TRANSFERENCIA SOB DEMANDA DE MÍDIAS NÃO
LINEARES PARA DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso
de Bacharelado em Ciências da
Computação
da Universidade Federal de Santa
Catarina.
Orientador: Prof. Dr. Mário Dantas

Florianópolis
2014

RESUMO

A utilização de dispositivos móveis para leitura tem se tornado cada vez mais comum na sociedade, porém algumas limitações desses dispositivos, em especial limitações sobre a transferência de dados diminuem a utilidade desses dispositivos para estes fins. Alguns tipos de livros adicionam complexidade adicional ao ser transmitido pela rede enquanto o usuário realiza a leitura, como é o caso dos livros não lineares, usados principalmente em entretenimento. O Sistema desenvolvido proporciona uma solução para melhorar a transmissão de livros não lineares através de um modelo estatístico onde assume que o usuário tem a tendência de tomar as mesmas decisões que a maioria das pessoas que já leram aquele mesmo livro.

1	Introdução	7
1.1	Contextualização	7
1.2	Objetivos	7
1.3	Objetivos específicos	7
1.4	Estrutura do documento	8
2	Sistemas distribuídos.....	9
2.1	Arquiteturas computacionais.....	9
2.2	Clusters.....	12
2.3	Multi-cluster.....	14
2.4	Grids computacionais.....	15
2.5	Nuvem computacional	18
3	Computação Móvel.....	21
3.1	Comunicação sem fio.....	22
3.2	Mobilidade	22
3.3	Rede sem fio.....	24
3.4	Desconexão	26
4	Proposta.....	28
4.1	Fluxo linear	29
4.2	Fluxo não linear	30
4.3	Grafo de fluxo	31
4.4	Transferência sob demanda.....	32
5	Resultados	36
5.1	Desenvolvimento dos testes	36
5.2	Análise sobre os dados obtidos	37
6	Conclusões e Trabalhos Futuros	39

6.1 Limitações.....	39
6.2 Trabalhos Futuros	39
Referências.....	40

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Devido ao constante aumento no número de dispositivos móveis utilizados no dia a dia, tornou-se necessária a adaptação de livros para esses tipos de dispositivos devido às limitações de rede, bateria e tamanho.

O problema abordado consiste em uma forma de disponibilizar um livro pela rede enquanto o usuário o lê, tais livros possuem a característica de que a leitura que não é feita página após página e em determinados momentos o usuário pode avançar para outra página. Esses livros são utilizados para o ensino de forma mais interativa e também para entretenimento.

1.2 OBJETIVOS

O objetivo do trabalho é propor um sistema que permita a transferência de um livro pela rede, enquanto o usuário lê. A transferência utiliza um sistema de predição que torna a transferência melhor que se transferido em páginas sequenciais.

1.3 OBJETIVOS ESPECÍFICOS

- Estudar conceitos e limitações de dispositivos móveis;
- Estudar conceitos de computação distribuída;
- Identificar as limitações que afetam o tipo da aplicação a ser desenvolvida;
- Propor um sistema para transferir um livro sob demanda que seja melhor que a transferência linear;

1.4 ESTRUTURA DO DOCUMENTO

O trabalho está organizado da seguinte forma: o capítulo 2 aborda os principais conceitos sobre sistemas distribuídos e como eles são utilizados nos dias de hoje. No capítulo 3 são apresentados conceitos sobre computação móvel, suas limitações e características. No capítulo 4 é apresentada a aplicação desenvolvida e a solução para transferência proposta.

2 SISTEMAS DISTRIBUÍDOS

Na década de 90, verificou-se a disponibilização de conexões mais rápidas, e computadores pessoais tornaram-se configurações mais potentes. Diante desse quadro, a interconexão de computadores, local ou geograficamente distribuídos, torna-se uma forma a maximizar o poder computacional. Popularizam-se as chamadas configurações dos sistemas distribuídos.

Para Tanenbaum e Steen (2012), um sistema distribuído é um conjunto de computadores independentes que se apresentam como um único sistema para seus usuários. A construção de um sistema distribuído deve atender aos seguintes requisitos:

- Tornar os recursos acessíveis: facilitar para o usuário o acesso aos recursos remotos de forma controlada e eficiente;
- Transparência na distribuição: esconder do usuário o fato de que os processos e recursos estão fisicamente distribuídos;
- Abertura: oferecer para o usuário serviços de acordo com as regras padrão que descrevem sua sintaxe e semântica;
- Escalabilidade: adicionar recursos ao sistema de maneira fácil;

2.1 ARQUITETURAS COMPUTACIONAIS

A classificação de arquitetura de computadores mais aceita é conhecida como taxonomia de Flynn. Proposta por Michael J. Flynn em 1966, esta taxonomia leva em consideração a quantidade de instruções executadas em paralelo pelo conjunto de dados para os quais as instruções são submetidas. (DANTAS, 2005)

As classificações de computadores são divididas em:

- SISD (Single Instruction Single Data): executa uma instrução de um programa por vez, ou seja, são computadores que apresentam apenas um processador. Este modelo pode ser representado pelo computador pessoal, onde as instruções são executadas sequencialmente. (Figura 1)

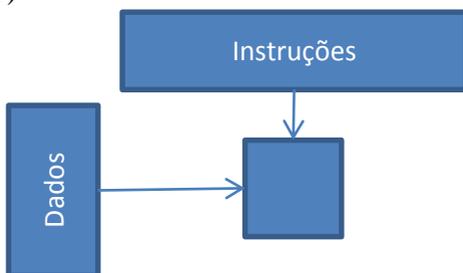


Figura 1 - SISD segundo a taxonomia de Flynn

- SIMD (Single Instruction Multiple Data): executa uma única instrução, porém esta instrução é processada sobre diferentes itens e dados. Isso ocorre devido à existência de facilidades de hardware para armazenamento de múltiplos dados.

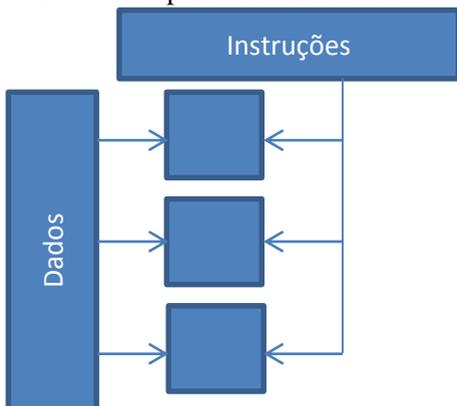


Figura 2 - SIMD segundo a taxonomia de Flynn

- MISD (Multiple Instruction Single Data): nesse tipo de arquitetura são executadas múltiplas instruções sobre um único conjunto de dados. No entanto, não se tem

conhecimento da existência de computadores com essa classificação.

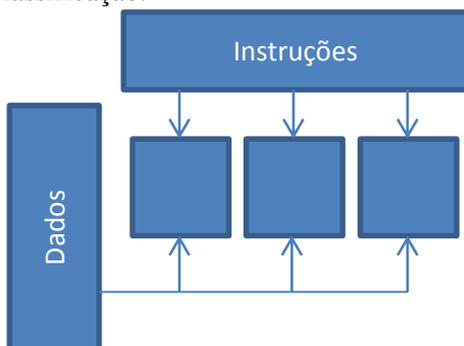


Figura 3 - MISD segundo a taxonomia de Flynn

- MIMD (Multiple Instruction Multiple Data): executa múltiplas instruções sob múltiplos dados. Computadores com essa arquitetura possuem múltiplos processadores em que cada um pode executar instruções independente dos demais.

Arquiteturas MIMD, de modo geral, são classificadas como multiprocessadores e multicomputadores, nas quais se diferenciam pelo compartilhamento ou não de memória. (MEFFE; MUSSI; MELLO, 2006).

- Multiprocessadores: arquitetura conhecida como fortemente acoplada, uma vez que processadores e memórias são interligados através de um sistema local de interconexão. Estas arquiteturas possuem características de possuir diversos processadores compartilhando uma ou um conjunto de memórias. A escalabilidade dessa arquitetura varia entre algumas centenas de processadores.
- Multicomputadores: Arquitetura conhecida como fracamente acoplada. Esta configuração caracteriza-se por obter centenas de processadores com suas próprias memórias locais. A comunicação entre os processadores é

efetuado através da troca de mensagens entre os processos que estão sendo executados.

2.2 CLUSTERS

A ideia inicial de cluster computacional foi desenvolvida na década de 60 pela IBM como uma forma de conectar grandes mainframes para prover paralelismo comercial de baixo custo (BUYYA, 1999).

Diante disso, clusters são definidos por Buyya (1999) como um sistema de processamento paralelo ou distribuído, formado por um conjunto de computadores interligados que trabalham como um recurso computacional integrado.

Dantas (2005) entende que as configurações de clusters podem ser como um agregado de computadores, de forma dedicada, ou não, para a execução de aplicações específicas de uma organização.

Um cluster é composto por dois ou mais computadores (nós), interconectados por uma rede de alto desempenho. Seu principal objetivo é realizar o processamento de aplicações de maneira distribuída e transparente ao usuário. Em outras palavras, distribuir o processamento entre os nós, mas de forma que pareça estar sendo processado em um único computador.

Segundo Dantas (2005), clusters podem ser classificados de acordo com o limite geográfico, utilização dos nós, tipo de hardware, aplicação alvo, tipos de nós e sistema operacional.

- Limite geográfico: baseado em sua localização e quantidade, os clusters podem ser classificados como: clusters de grupo, com capacidade de 2 a 99 nós interconectados por dispositivos especiais; clusters departamental suportando de dezenas a centenas de nós; clusters com capacidade de mais de cem nós.
- Aplicação Alvo: Comumente existem dos alvos principais às aplicações comuns, aplicações que necessitam de alto

desempenho e aplicações que necessitam de alta disponibilidade. Aplicações que precisam de alto desempenho estão procurando maior quantidade de processadores, maior quantidade de memória e capacidade de armazenamento, enquanto as que precisam de alta disponibilidade não toleram interrupções. Vale ressaltar que uma aplicação pode exigir ambos os tipos.

- Utilização de nós: pode ser estabelecida através da participação de nós dedicados ou não dedicados que compõem o cluster. Em cluster não dedicados, as aplicações são executadas em ciclos ociosos dos nós, enquanto clusters dedicados são projetados para executarem exclusivamente as aplicações.
- Tipo de hardware: os tipos de hardware empregados em clusters são classificados como: Nows, CoPs, PoPs, Cows e Clumps.
 - As Nows (Network of Workstations) são caracterizadas por usarem estações de trabalho distribuídas na rede local para compor o cluster. De forma análoga, CoPs (Clusters of PCs) ou PoPs (Pile of PCs) utilizam computadores pessoais do tipo PCs.
 - Configurações Cows (Cluster of Workstations) são geralmente construídas por máquinas dedicadas e homogêneas para a execução de aplicações específicas. Além disso, dispõem de uma rede de interconexão específica.
 - Clumps (Clusters of SMPs) são compostos de máquinas com arquiteturas SMP.
- Tipos de nós: Clusters podem ser classificados como do tipo Homogêneo ou Heterogêneo.
 - Nos clusters homogêneos todos os nós possuem arquiteturas semelhantes e o mesmo sistema operacional.
 - Em clusters heterogêneos, possuem arquiteturas diferentes ou sistemas operacionais distintos.

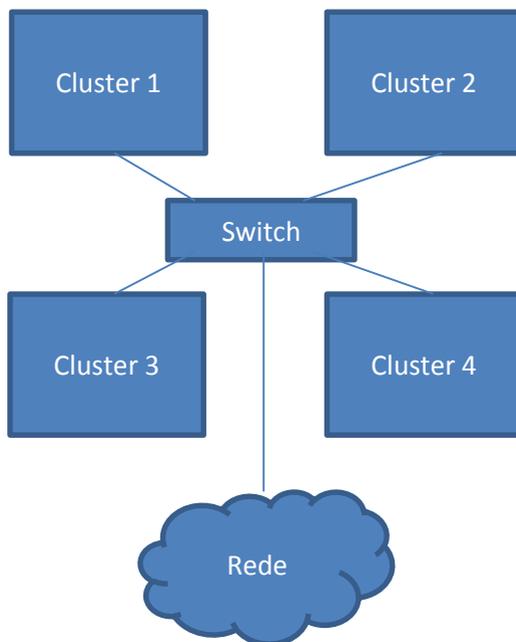
Um único cluster, muitas vezes não possui capacidade de resolver problemas que demandam grande poder computacional.

Nesse caso, múltiplos clusters interconectados formam um ambiente chamado de multi-cluster. Multi-clusters podem aumentar o poder computacional para a resolução de problemas desse tipo.

2.3 MULTI-CLUSTER

Conforme já abordado, os clusters possuem a localização restrita á um domínio. Porém é possível e possivelmente necessário que clusters de múltiplos domínios se conectem para formar um sistema único de maior escala. Tais sistemas são conhecidos como multi-cluster. Diferentemente de clusters, os multi-clusters são formados por um conjunto de clusters conectados por uma rede WAN (World-Area-Network). Os sistemas multi-clusters podem ser classificados como super-cluster e cluster-de-cluster (JAVADI; AKBARI; ABAWAJY, 2006). Super-clusters podem ser caracterizados por possuir um grande número de processadores homogêneos e redes de comunicação heterogêneas. Cluster-de-cluster no entanto, é construído pela conexão de múltiplos clusters, com heterogeneidade tanto na rede de comunicação quanto nos processadores. Entretanto, existe um alto grau de complexidade nos ambientes de multi-clusters, no que diz respeito á problemas de escalabilidade de tarefas, pois seus recursos são heterogêneos, distribuído e altamente compartilhado no tempo e no espaço. Por conta disso, o recebimento contínuo de tarefas e mudanças dinâmicas da disponibilidade da capacidade de CPU dificulta a utilização de algoritmos que normalmente são executados em ambiente de clusters (ABAWAJY; DANDAMUDI, 2003).

A configuração de sistemas multi-clusters forma uma visão unificada do compartilhamento de recursos e serviços do ambiente (DANTAS, 2005).



2.4 GRIDS COMPUTACIONAIS

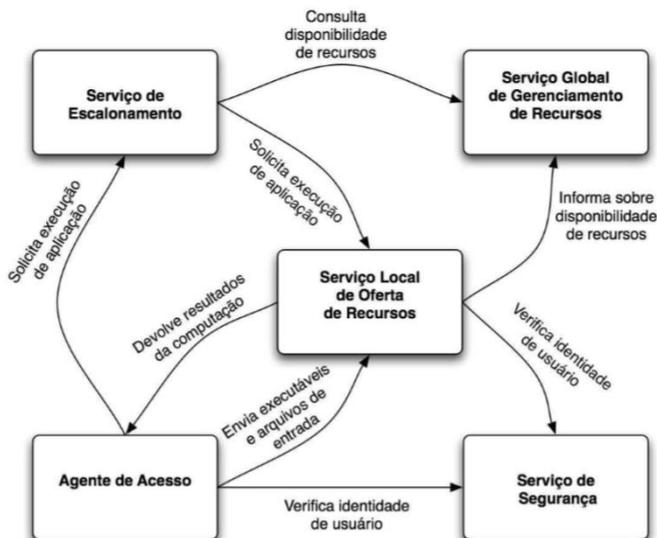
Na década de 90, devido a maior disponibilidade de conexões mais rápidas e com a necessidade de poder computacional por parte de aplicações, pesquisadores começaram a imaginar uma infraestrutura computacional que conectasse computadores e clusters geograficamente distribuídos para atender essa necessidade. Surge então a proposta chamada de grade computacional, em inglês *computational grid*. A nomenclatura é devido a analogia à rede elétrica (power grid), onde a energia é disponibilizada sem se preocupar sobre onde ela está sendo gerada e como está sendo destinada. A ideia de uma grade computacional é criar um ambiente computacional distribuído que possua mecanismos que permitam o processamento, armazenamento e uso de recurso de forma transparente para o usuário (DANTAS, 2005).

Grid computacional pode ser entendida como uma plataforma heterogênea de computadores geograficamente distribuídos, onde usuário acessam seus recursos através de uma interface única (DANTAS, 2005).

Segundo proposto por Krauter, Buyya e Maheswaran (2012), os sistemas de grids existentes podem ser divididos nas seguintes categorias:

- Grid Computacional (Computational Grid): Unem recursos computacionais geograficamente distribuídos a fim de obter alta capacidade de processamento. Dependendo da forma que essa capacidade é utilizada, essa categoria pode ser subdividida em: supercomputação distribuída e alta taxa de transferência.
- Grid de dados (Data Grid): Composto por uma infraestrutura para acesso, pesquisa e processamento de informações a partir de repositórios de dados.
- Grid de Serviço (Service Grid): Normalmente administradas por grandes instituições, se enquadra nessa categoria sistemas que reúnem alto desempenho, recursos computacionais dedicados e grandes sistemas de armazenamento de dados (BRASILEIRO et al., 2008). Como exemplo desse tipo de grid pode-se citar: SINAPAD no Brasil, e NGS no Reino Unido.

Uma configuração de grid é composta por organizações virtuais, que podem ser indivíduos ou entidades que compartilham recursos, porém, podem apresentar diferentes políticas de acesso e uso de algumas restrições quanto a disponibilidade dos mesmos. Diante disso, foi proposto arquiteturas que permitam que diferentes organizações virtuais se comuniquem. Camargo et. al. apresenta uma arquitetura implementada pela maioria dos sistemas de grid (CAMARGO et al., 2006) ilustrada a seguir.



- **Agente de acesso:** é a ponte de acesso para os usuários, está presente em cada nó que solicite a execução de aplicações.
- **Serviço local de oferta de recursos:** responsável por iniciar a execução da aplicação, reportar erros e retornar resultados. É executado em cada máquina que disponibiliza recursos para a grid.
- **Serviço global de gerenciamento de recursos:** responsável por responder a solicitação de uso de recursos, também tem função de monitoramento, a fim de combinar as requisições com recursos oferecidos.
- **Serviço de escalonamento:** escalonando aplicações para recursos disponíveis. Recebe solicitações de execução, obtém informações quanto à disponibilidade, e determina onde cada aplicação será executada.
- **Serviço de segurança:** é responsável por evitar que os recursos compartilhados sofram ataques de aplicações maliciosas e por autenticar usuários.

2.5 NUVEM COMPUTACIONAL

Nuvem computacional (Cloud Computing) vem sendo utilizada quando um produto ou serviço necessita obter recursos computacionais sob demanda. Embora sua acensão esteja acontecendo nos dias de hoje, o conceito básico de nuvem computacional não é tão novo, tendo sido prevista nos anos 60 pelo cientista da computação John McCarthy, quando mencionou que a computação poderia um dia ser organizada como de utilidade pública, exatamente como o sistema de telefonia. (GARFINKEL, 1999).

O conceito de nuvem computacional pode ter diferentes percepções, devido ao fato que não é uma tecnologia nova e sim uma junção de tecnologias já consolidadas executadas de maneira diferenciada.

Segundo Foster et al. (2008) nuvem computacional é um paradigma de computação distribuída de larga escala, impulsionado pela economia, na qual um conjunto de recursos abstratos, virtualizados, escalonáveis dinamicamente, com poder computacional gerenciável e armazenável, proporcionado aos usuários externos através da Internet. Segundo a NIST (MELL; GRANCE, 2011), responsável por desenvolver padrões e diretrizes, uma nuvem computacional é um modelo para permitir acesso à rede sob demanda, de forma ubíqua e conveniente para o compartilhamento de recursos computacionais.

O objetivo da nuvem computacional é fazer melhor uso de recursos distribuídos, além da capacidade para resolver problemas computacionais de larga escala. (RIMAL; CHOI; LUMB, 2009).

De acordo com Zhang, Cheng e Boutaba (2010) a arquitetura de uma nuvem computacional está dividida em modelos de níveis, modelo de serviços e tipos de nuvens. Cada divisão será detalhada a seguir.

- Modelo de Níveis: a arquitetura de nuvem, de modo geral, pode ser subdividida em quatro níveis. Fracamente

acoplados com os outros, permitindo que sejam desenvolvidos separadamente.

- Nível de aplicação: contém a aplicação que será executada na nuvem.
 - Nível de Plataforma: adiciona uma coleção de ferramentas, middleware e serviços especializados para proporcionar uma plataforma de desenvolvimento (FOSTER et al., 2008)
 - Nível de infraestrutura ou virtualização: responsável por criar um *pool* de armazenamento e recursos computacionais através do particionamento de recursos físicos utilizando tecnologias de virtualização.
 - Nível de hardware: responsável por gerenciar recursos físicos, como servidores, roteadores e switches.
- Modelo de serviços: os níveis da arquitetura da nuvem computacional podem ser implementados como um serviço para o nível acima. Consequentemente todos os níveis podem ser percebidos como clientes do nível abaixo (ZHANG;CHENG;BOUTABA,2010). Em geral, as nuvens computacionais oferecem serviços que podem ser agrupados em três categorias.
 - Software como Serviço (SaaS): refere-se ao fornecimento de aplicações acessadas pelos usuários através da internet.
 - Plataforma como Serviço (PaaS): oferece ao usuário um ambiente para desenvolver aplicações personalizadas. Geralmente desenvolvedores precisam respeitar algumas restrições, como os softwares que podem ser utilizados. Nessa categoria, o usuário não gerencia ou controla a infraestrutura da nuvem, no entanto, lhe é permitido controlar aplicativos implementados.

- Infraestrutura como Serviço (IaaS): refere-se ao fornecimento de processamento, armazenamento, rede e outros serviços computacionais fundamentais capazes de executar software, incluindo sistemas operacionais e aplicativos. No entanto, o usuário não gerencia a infraestrutura da nuvem, o que lhe impede de controlar sistemas operacionais, armazenamento e desenvolvimento de aplicações (MELL; GRANCE, 2011).
- Tipos de Nuvens: segundo a NIST (MELL; GRANCE, 2011), existem quatro tipos de nuvens computacionais:
 - Nuvens Privadas: projetadas para uso exclusivo de uma única organização. Podendo ser construídas e gerenciadas pela organização ou por provedores externos.
 - Nuvens de Comunidades: a infraestrutura é compartilhada por várias organizações que possuem as mesmas preocupações, tais como missão ou requisitos seguros.
 - Nuvens Híbridas: a infraestrutura é composta com duas ou mais nuvens unidas por tecnologia que permite a portabilidade de dados e aplicações.

3 COMPUTAÇÃO MÓVEL

A computação móvel evidenciou uma significativa expansão nos últimos anos, possibilitando que usuários de dispositivos móveis tenham acesso a um ambiente de rede e compartilhem dados e recursos. A crescente utilização deste paradigma é originada principalmente por avanços obtidos nas tecnologias de rede sem fio, que proporcionam aos usuários acesso a uma infraestrutura compartilhada, independente de sua localização física (Forman; Zahorjan, 1994). Outro aspecto que deve ser observado é a disseminação da utilização destes equipamentos devido a redução de custo, melhorias no transporte, tamanho e peso, além de funcionalidades extras incorporadas aos aparelhos. Um exemplo é o *smartphone* que reúne funções de computação e de telefone celular.

Se por um lado, dispositivos móveis propiciam o acesso a informações de qualquer lugar e a qualquer momento (Perry et al. 1999), de outro, diversos desafios surgiram juntamente com essa tecnologia. Alguns atores (Borges, 2006b), (Bruneo et al., 2003), (Clarke; Humphrey, 2002), citam diversos problemas que são impostos pelo paradigma da computação móvel. Do mesmo modo, os avanços na computação móvel tornam possíveis mais usuários acessarem serviços através da rede, independente de sua localização, entretanto, características inerentes revelam alguns obstáculos. Algumas dessas características são:

- Menor capacidade de processamento e memória;
- Baixa autonomia devida à duração da bateria;
- Grande número de dispositivos diferentes no mercado;
- Qualidade inferior da rede;
- Menor capacidade de armazenamento;

O trabalho de (Forman; Zahorjan, 1994) mostra que os desafios da computação móvel não são triviais, e fogem do paradigma conhecido de sistemas distribuídos. O desenvolvimento de sistemas para ambientes que envolvem dispositivos móveis e redes sem fio, necessitam de um tratamento diferenciado de projeto

convencional de computação distribuída (Chen; Kotz, 2001). Os elementos que devem ser considerados envolvem, principalmente, as características desses ambientes, o padrão de comunicação e rede.

3.1 COMUNICAÇÃO SEM FIO

Comunicação sem fio, ou *wireless*, estabelece troca de mensagens entre dois pontos ou dispositivos utilizando de ondas eletromagnéticas para transmissão de dados. Em função do acesso sem fio, a comunicação na computação móvel encontra muitos obstáculos, devido ao fato que o ambiente interage com o sinal (Pitoura; Samaras, 1998). Além disso, diferentemente de computadores estacionários, dispositivos móveis encontram múltiplas conexões de rede heterogêneas em diversos caminhos (Forman; Zahorjan, 1994).

Problemas decorrentes de comunicação sem fio envolvem: largura de banda limitada ou variável, alta taxa de erros, maior latência, retransmissão e *timeout*, largura de banda dividida entre usuários de uma mesma célula, movimentação do usuário e desconexões frequentes.

Observa-se ainda, que problemas de segurança na comunicação são mais facilmente expostos que em uma rede cabeada, principalmente se a área de transmissão for grande, exigindo que projetos de software incluam medidas para aumentar a segurança, geralmente usando criptografia (Forman; Zahorjan, 1994).

3.2 MOBILIDADE

Mobilidade é a capacidade de mudar sua posição enquanto conectado a uma rede (Forman; Zahorjan, 1994). Essa capacidade é característica inerente de sistemas de computação móvel, no entanto, a partir dessa característica, surgem outros fatores relevantes, tais

como informações que em ambientes fixos eram considerados estáticos, em ambientes móveis tornam-se dinâmicos. Por exemplo, em um computador fixo, o servidor a ser usado pode ser configurado de maneira estática, enquanto em um dispositivo móvel, é necessário um mecanismo para determinar qual servidor usar, dependendo do local. Podemos assim observar que em um ambiente móvel, a localização do usuário pode ser estimada com um item de dado no qual valores mudam a cada movimento. (Imielinski; Viswanathan; Badrinath, 1994).

Para (Jing; Abdelsalam; Elmagarmid, 1999) a computação móvel se distingue da clássica conexão fixa devido a dois fatores.

1. A mobilidade de usuários nômades e seus computadores, implicando que os usuários podem conectar-se de diferentes *access points*, e podem ficar conectados enquanto se movem, apesar da possibilidade de desconexão.
2. Restrições de recursos móveis, tais como largura de banda escassa e tempo de vida de bateria limitado.

A mobilidade apresenta benefícios para usuários móveis na medida em que estes não precisam conhecer a localização do servidor, dos dados ou de recursos que desejam acessar. Sistemas devem proporcionar mecanismos que deem suporte a essa mobilidade de maneira transparente ao usuário final, como apresenta (Ferreira et al., 2007).

Segundo (Imielinski; Viswanathan; Badrinath, 1994), mobilidade é um comportamento com implicações para ambas as redes, fixas e sem fio. Na rede fixa, os usuários móveis podem estabelecer uma conexão com diferentes tipos de portas em locais diferentes. Já a conexão sem fio permite mobilidade e conectividade irrestrita de qualquer local dentro da área de cobertura.

Segundo (Satyanarayanan, 1996) a mobilidade possui restrições intrínsecas:

- Elementos móveis são recursos pobres em relação a elementos estáticos, considerando diversos aspectos tais

como tamanho, energia, ergonomia e recursos computacionais.

- Existe uma restrição de mobilidade referente ao aspecto de vulnerabilidade quanto à perda ou roubo, ou seja, além dos benefícios da mobilidade, herdamos as fragilidades.
- Conectividade é altamente variável e impacta no desempenho e confiabilidade.
- Elementos móveis dependem de uma fonte de energia finita. Enquanto não estiverem disponíveis melhorias quanto às fontes de energia, será necessário considerar esse aspecto em projetos de software ou hardware.

3.3 REDE SEM FIO

O objetivo das redes sem fio é fornecer conectividade em uma infraestrutura sem fios e cabos. Estas redes proporcionam aos usuários móveis meios de comunicação com outras entidades (usuários, recursos, redes). (Coulouris; Dollimore; Kindberg, 2007) apresenta quatro tipos de redes sem fio: WPAN, WLAN, WMAN e WWAN.

- WPAN (Wireless Personal Area Network) fornece conectividade com uma pequena abrangência para dispositivos móveis. É interessante para interligar teclados, impressoras, telefones móveis, agendas eletrônicas, computadores de mão e outros. Os padrões mais utilizados para estabelecer a comunicação são ondas de raio infravermelho. Exemplo: Bluetooth (IEEE 802.15.1)
- WLAN (Wireless Local Area Network) fornece conectividade para dispositivos móveis dentro de uma área (casa, prédio) e prove acesso a internet ou outra rede fixa. Possui variantes de padrões, oferecendo largura de banda de 10 e 100 Mbps com abrangência de até 1,5 quilômetros. Exemplo: Wifi (IEEE 802.11)

- WMAN (Wireless Metropolitan Area Network) possuem, geralmente, maior abrangência que as WLAN e são utilizadas para interligar uma cidade ou região metropolitana. Exemplo: WiMax (IEEE 802.16)
- WWAN (Wireless Wide Area Network) provê aos usuários de telefonia móvel acesso com ampla cobertura, seja por antenas ou sistemas de satélite. É importante salientar que estas redes possuem taxa de transmissão de dados relativamente baixos comparado as demais redes. Exemplos: GSM, 3G

Conforme (Pitoura; Bhargava, 1994), uma estrutura de sistemas distribuídos que suporta mobilidade possui uma arquitetura que consiste em dois tipos distintos de hosts: hosts móveis e fixos. Os hosts móveis são considerados computadores ou dispositivos portáteis interligados em rede por meio de uma conexão sem fio, possibilitando a mobilidade. Já hosts fixos são computadores que estão em uma rede cuja localização e conectividade são constantes. Alguns dos hosts fixos, chamados de estação base ou estação de suporte móvel, são acrescidos com uma interface wireless para comunicar com hosts móveis.

Enquanto em sistemas distribuídos não móveis, um host opera de dois modos diferentes, conectado ou desconectado. Em ambientes móveis, o host pode operar de diversas formas, sendo um dos fatores determinantes a disponibilidade de largura de banda. Desconexões frequentes, sejam elas totais ou parciais, não devem ser tratadas como falhas, ou seja, um host móvel deve ser capaz de operar em casos de impossibilidade de conexão. Os autores (Pitoura; Bhargava, 1994) ainda comentam que outro fator que pode determinar o modo de operação do dispositivo é o tempo de vida de bateria. Nesse caso, o dispositivo poderia entrar em um estado de economia de energia.

Também é destacado que a mobilidade também pode causar a perda ou degradação da conexão wireless, visto que os usuários podem mover-se entre as células. Nesse contexto, o número de

dispositivos em uma célula varia dinamicamente, podendo causar também a sobrecarga da capacidade da rede wireless.

3.4 DESCONEXÃO

No cenário do ambiente móvel, entende-se que um host móvel está conectado quando está em completo modo de operação. Por outro lado, considera desconectado o host móvel que se encontra inacessível ao restante da rede (Imielinski; Badrinath, 1993). As desconexões em um ambiente com características de mobilidade podem ocorrer por diversas razões, por exemplo, a alta variabilidade da qualidade da conexão, necessidade de economia de energia ou término da mesma, interferências, sombreamento (variação de sinal causada pela obstrução do sinal por outros objetos tais como montanhas ou prédios)

Tendo em vista as propriedades e os problemas citados anteriormente é possível afirmar que o ambiente móvel é suscetível a desconexões, o que se pode considerar como uma característica inerente da computação móvel. Segundo (Mateus; Loureiro, 1998) e (Pitoura; Samaras, 1998) desconexões podem ser classificadas como voluntárias ou forçadas; previsíveis ou repentinas; curtas ou longas.

- Voluntárias quando a desconexão é intencional, por exemplo, o usuário pode evitar o acesso à rede para diminuir o custo da tarefa de comunicação, o consumo de energia, para economizar no uso da largura de banda.
- Forçadas quando a região onde o usuário está não tem cobertura para acesso à rede.
- Previsíveis podem ser de natureza voluntária, em função da variação na taxa sinal ruído (SNR – Signal-to-Noise Ratio), ou ainda quanto a energia disponível na bateria.
- Repentinias acontecem de forma abrupta;

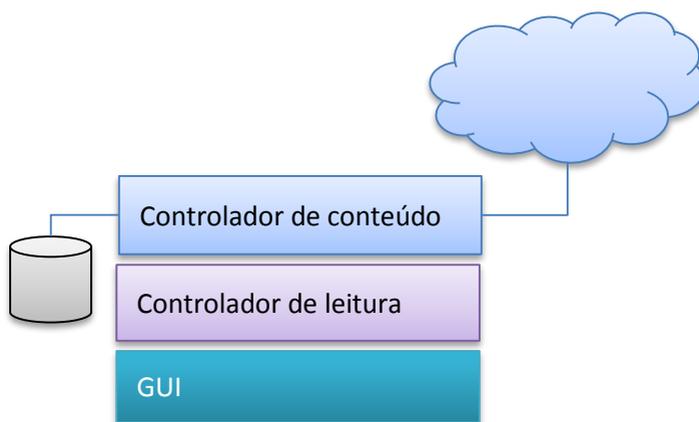
- Curtas ou longas são relativos ao tempo, por exemplo, um obstáculo no sinal pode ser considerada uma desconexão curta, já a falta de energia na bateria pode durar por um longo tempo.

4 PROPOSTA

O aplicativo desenvolvido consiste em um sistema para distribuição de livros interativos não lineares para smartphones, o livro deverá ser distribuído pela rede enquanto o usuário o lê. O aplicativo possui um sistema que busca reduzir a quantidade de informações trafegadas pela rede.

O usuário inicia sua leitura e segue até um momento que seja perguntado qual fluxo ele deve seguir, o livro então é exibido páginas do fluxo escolhido.

O aplicativo foi separado em três camadas para seu funcionamento. A figura abaixo ilustra essas camadas



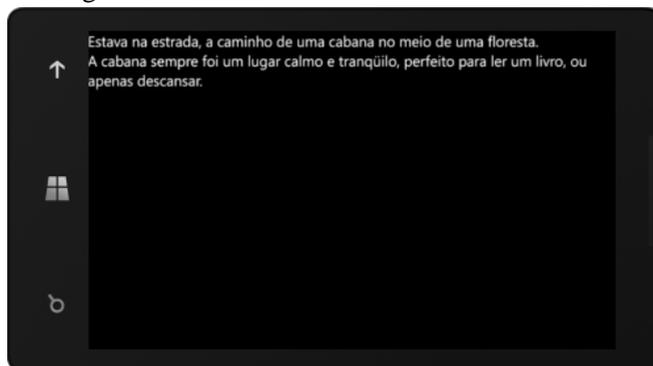
- GUI é responsável por exibir informações e interação com o usuário;
- Controlador de leitura fornece a GUI informações sobre a página a ser exibida, e armazena informações sobre o caminho tomado pelo usuário localmente, ao final da leitura ele envia essas informações para a aplicação remota que a utiliza para melhorar a previsão para os próximos acessos;
- Controlador de conteúdo é responsável por obter informações de páginas e prover ao controlador de leitura

quando solicitado. Essa camada utiliza do grafo de fluxo para priorizar download de páginas que será realizado em plano de fundo, enquanto o usuário lê o livro.

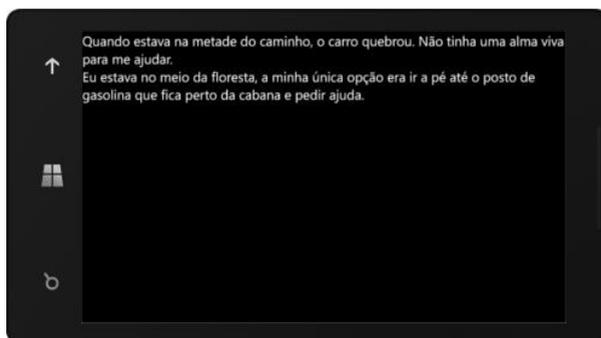
A seguir será mostrado dois tipos de fluxos distintos que o controlador de conteúdo leva em consideração para priorizar a transferência.

4.1 FLUXO LINEAR

Fluxo linear pode ser previsto com exatidão, pois não existe a possibilidade do usuário avançar para outro ponto da leitura. As imagens a seguir ilustram esse fluxo.

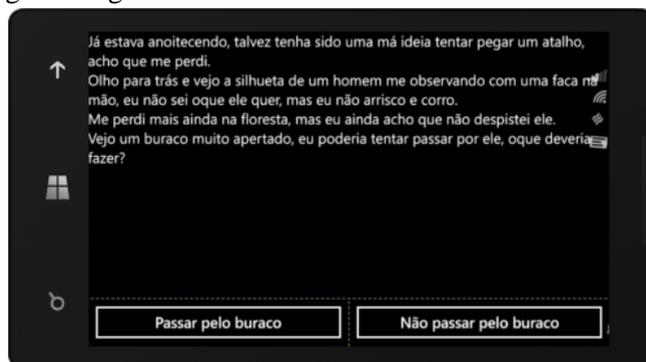


Quando o usuário tocar em qualquer local da tela, a tela a seguir é apresentada, não permitindo ao usuário tomar alguma ação imprevista.

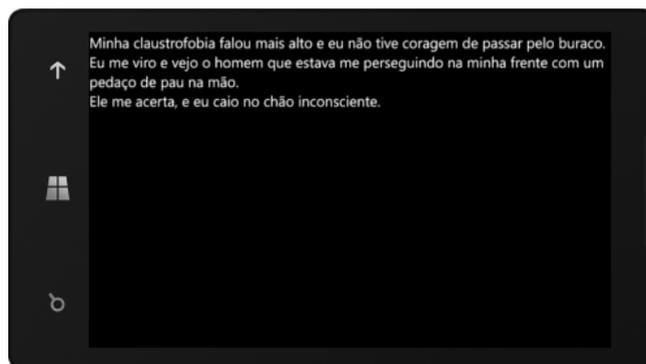


4.2 FLUXO NÃO LINEAR

Ao contrário de fluxos lineares, fluxos não lineares não podem ser previstos com exatidão, pois depende de uma escolha do usuário. A imagem a seguir ilustra a necessidade dessa decisão.

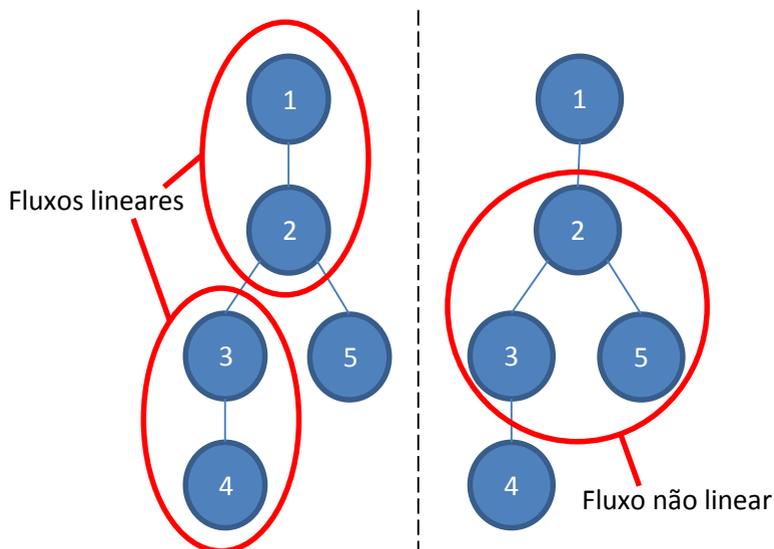


Ao tocar em um botão, o usuário então escolhe o novo fluxo para o livro.

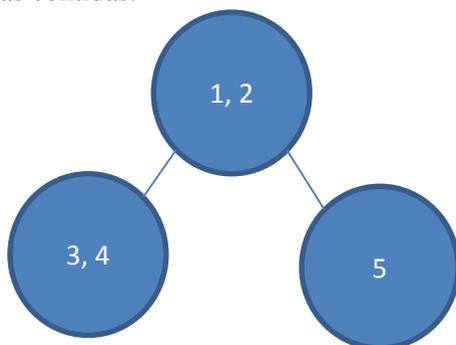


4.3 GRAFO DE FLUXO

Todo o fluxo da história apresentada pode ser representado em um grafo. A imagem a seguir ilustra um grafo para uma história fictícia.

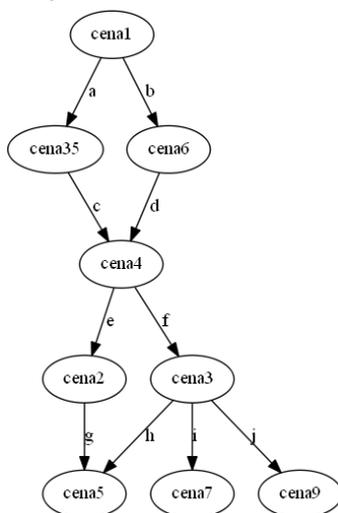


Para a transferência de páginas pode-se utilizar uma versão simplificada do mesmo grafo de fluxo, pois trechos onde ocorre a linearidade podem ser abstraídos para um único nó, com todas as páginas contidas.



4.4 TRANSFERÊNCIA SOB DEMANDA

Como vimos anteriormente, a decisão do usuário em seguir um determinado fluxo faz com que páginas que seriam utilizadas no fluxo não escolhido não sejam utilizadas, porém, para reduzir a quantidade de dados transmitidos pela rede, podemos utilizar algum tipo de mecanismo para a priorização da ordem de páginas a serem transferidas para o dispositivo do usuário. Para essa priorização, foi utilizado uma contagem de usuários que a partir de um nó seguem para um nó filho, a figura abaixo mostra essa contagem.

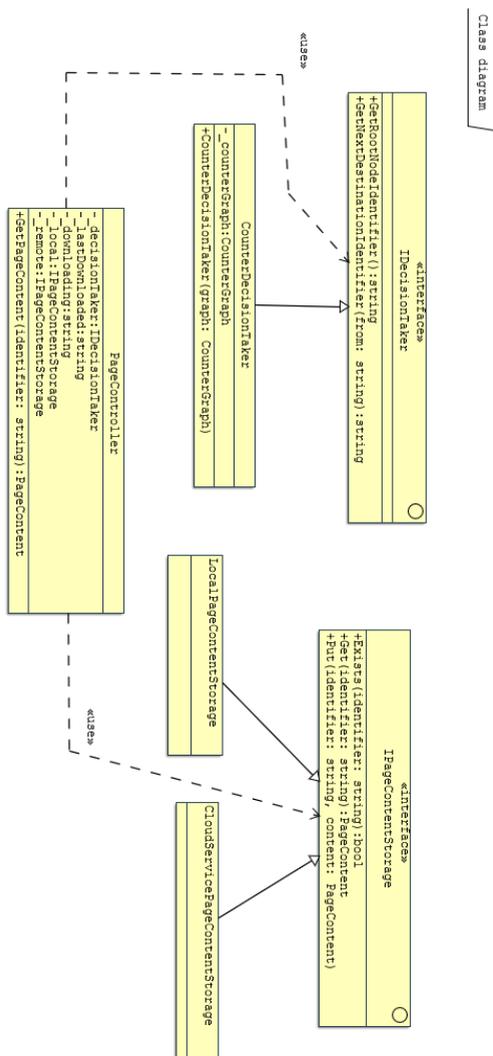


O grafo acima tem em seus vértices, representações de páginas de um determinado livro-jogo interativo, o peso de suas arestas é a quantidade de pessoas que tomaram a decisão de visitar o nó apontada por ela, por exemplo, a partir da página chamada de *cena1* um número *a* de usuários seguiram imediatamente para a *cena35* e *b* usuários seguiram para a *cena6*.

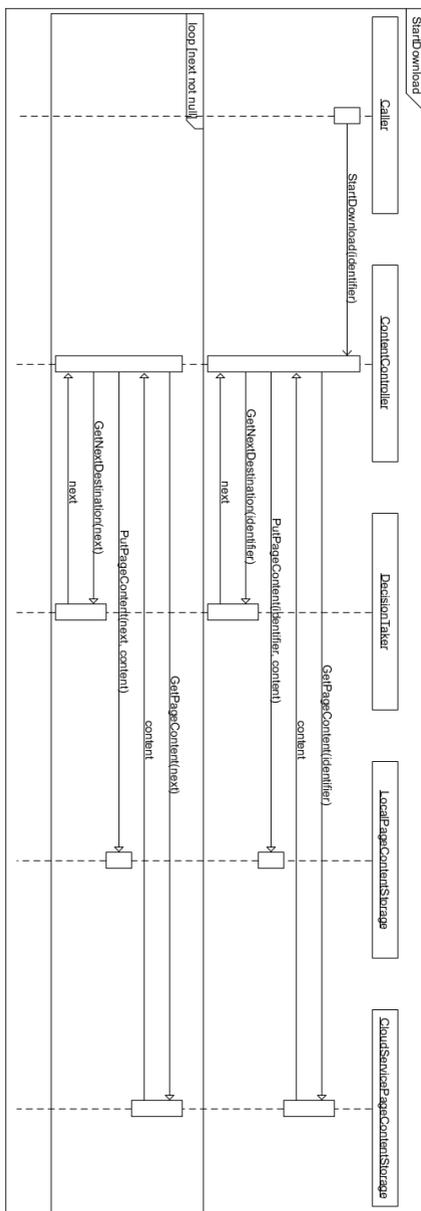
O método de priorização de páginas a serem transferidas, recebe esse grafo atualizado a partir de um serviço de armazenamento de uma nuvem computacional e começa a efetuar a transferência do conteúdo priorizando o caminho apontada pelas arestas de maior peso. Ao completar a transferência de todo o

conteúdo do nodo, ele começa então a transferir o conteúdo do próximo nodo filho do nodo recém transferido que possui a aresta com maior peso apontando para ele.

Para detalhar o funcionamento do controlador de conteúdo o diagrama de classes abaixo representa as classes utilizadas para que a transferência seja realizada.



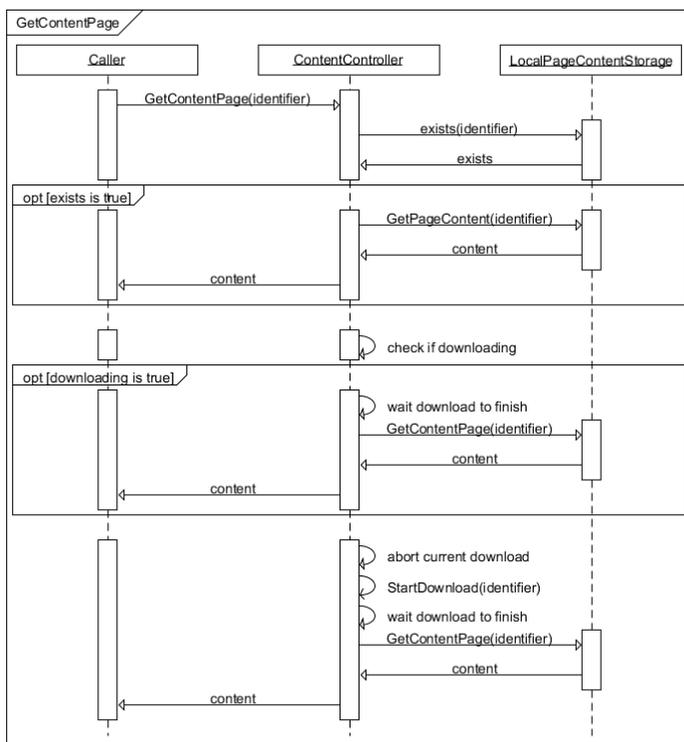
O diagrama de sequência que ilustra a tarefa de realizar a transferência é exibido abaixo.



Note que *CloudServicePageContentStorage* realiza a tarefa de obter do serviço de armazenamento na nuvem o conteúdo da página solicitada.

A tarefa acima deve ser executada durante toda a leitura do usuário e em plano de fundo.

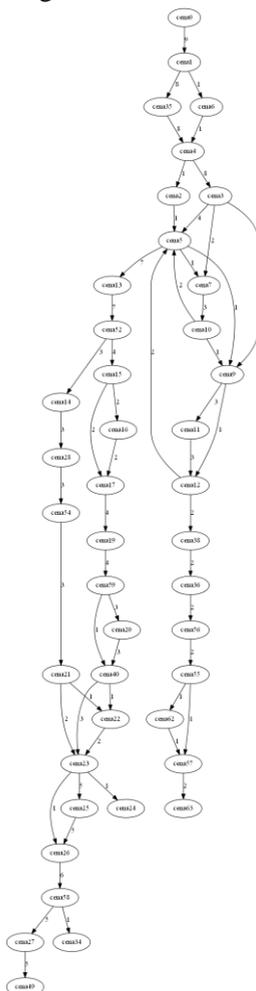
Ao usuário tomar uma decisão, primeiro deve ser verificado se o conteúdo a ser exibido para o usuário está disponível localmente, caso positivo, o sistema retorna a cópia local dessas informações com grande velocidade. Caso o conteúdo não esteja disponível localmente e não estiver atualmente sendo transferido, deve-se abortar a transferência do conteúdo atual e iniciar a transferência do conteúdo solicitado. Nesse caso existe um atraso da aplicação a realizar a ação solicitada pelo usuário. O diagrama de sequência abaixo ilustra essa tarefa.



5 RESULTADOS

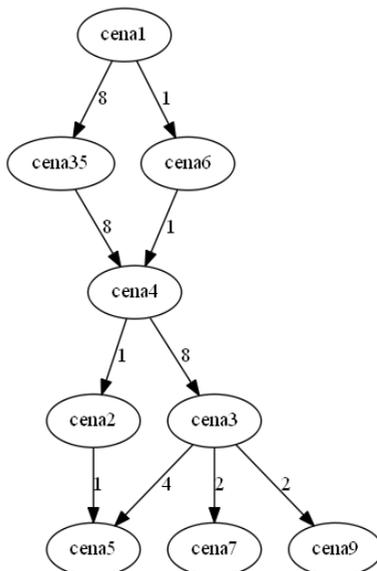
5.1 DESENVOLVIMENTO DOS TESTES

Foi realizado um teste com nove participantes utilizando um livro-jogo com o intuito de ensinar biologia de forma interativa e leve, o caminho realizado por cada participante foi capturado e exibida conforme o grafo a seguir:

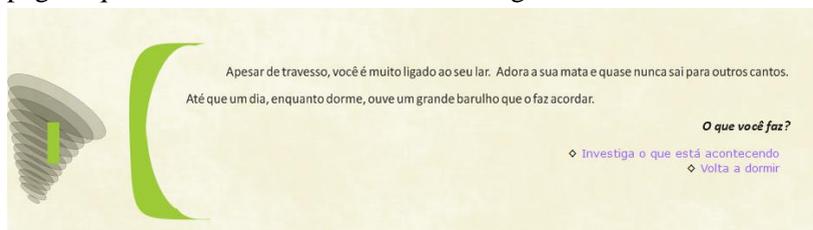


5.2 ANÁLISE SOBRE OS DADOS OBTIDOS

A fim de facilitar o entendimento, este capítulo irá focar apenas na parte inicial do grafo.



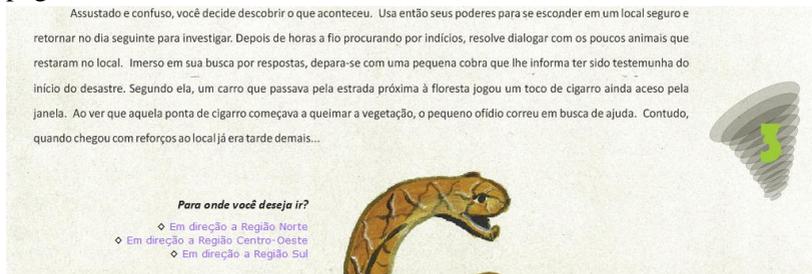
Como pode ser observado no grafo acima, a partir da *cena1* oito a cada nove pessoas tendem a ir para a *cena35*, ao observar a página que a *cena1* contém obtivemos o seguinte:



Um possível motivo para que os usuários convergissem a tomar a decisão de ir para a *cena35* (*investiga o que está acontecendo*) pode ser devido ao fato de que a outra opção não é atrativa o bastante para o usuário.

Para esse caso, a decisão de priorizar o download da *cena35* em comparação com a *cena6* se mostra bastante eficiente.

Ao realizar a mesma análise sobre a *cena3*, temos a seguinte página do livro:



Como pode ser observado pelo grafo e a imagem da página exibida acima, uma convergência menor dos usuários é obtida, porém, ainda com certa tendência de seguir para a *cena5*. Para esse caso, a possibilidade de erro na priorização do download é maior.

6 CONCLUSÕES E TRABALHOS FUTUROS

O Trabalho desenvolvido forneceu uma visão sobre algumas limitações encontradas em dispositivos móveis e o quanto é importante não ignorar tais limitações. Apenas por reduzir a quantidade de informações não utilizadas que são trafegadas através da rede sem fio, foi possível trazer benefícios como o prolongamento da vida útil da bateria e redução na espera de iniciar a leitura. Mídias não lineares têm uma grande quantidade de informações que o usuário não iria acessar, que em caso de download da mídia toda seriam desperdiçados.

6.1 LIMITAÇÕES

A abordagem proposta demonstrou-se menos eficiente quando o usuário toma decisão diferente da maioria, nesse caso, várias páginas que já foram transferidas seriam descartadas, causando problemas encontrado em dispositivos móveis devido ao uso desnecessário da rede.

6.2 TRABALHOS FUTUROS

A eficiência do aplicativo poderia ser melhorada com as seguintes otimizações:

- Reduzir o impacto causado pela falha ao prever decisão do usuário pode melhorar a experiência do usuário, reduzindo o tempo de espera para que a página seja exibida.
- Melhorar o sistema de priorização do download utilizando mais informações sobre o usuário pode reduzir a quantidade de falhas no sistema de previsão de decisões.

REFERÊNCIAS

ABAWAJY, J. H.; DANDAMUDI, S. P. *Parallel job scheduling on multicluster computing system*. In: IEEE. Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on. [S.l.], 2003. p. 11–18.

BORGES, V. C. M.; DANTAS, M. A. R. *Uma abordagem de submissão e monitoração de múltiplas tarefas para ambientes de grade computacional utilizando dispositivos móveis*. XXXIII SEMISH, 2006, pages 403–418.

BORGES, V.C.M. *Uma Abordagem de Submissão e Monitoração de Múltiplas Tarefas para Ambientes de Grade Computacional Utilizando Dispositivos Móveis*. Dissertação (Mestrado) – Universidade Federal de Santa Catarina (UFSC), 2006.

BRUNEO, Dario; SCARPA, Marco; ZAIA, Angelo; PULIAFITO, Antonio. *Communication paradigms for mobile grid users*. 3rd (CCGrid'03), p. 669–676, May 2003

BUY YA, R. (Ed.). *High Performance Cluster Computing: Architectures and Systems*. Prentice hall. [S.l.: s.n.], 1999.

CHEN, Guanling; KOTZ, David. *A Survey of Context-Aware Mobile Computing Research*. Dartmouth College, Hanover, NH, 2000.

CLARKE, B; HUMPHREY, M. *Beyond the 'device as portal': Meeting the requirements of wireless and mobile devices in the legion grid computing system*. 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and

Mobile Computing (associate with IPDPS 2002), p. 192–199, Abril 2002.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.
Sistemas Distribuídos: conceitos e projeto. [S.l.]: Bookman, 2007.
ISBN 978-85-60031-49-8.

DANTAS, M. *Computação distribuída de alto desempenho: redes, clusters e grids computacionais*. [S.l.]: Axcel Books, 2005.

FERREIRA, D. J.; DANTAS, M.A.R.; PINTO, A. R.;
MONTEZ, C.; RODRIGUEZ, Martius. *High Performance Computing Systems and Applications*. In HPCS 2007. 21st International Symposium, May 2007.

FOSTER, I. et al. *Cloud computing and grid computing 360-degree compared*. In: IEEE. Grid Computing Environments Workshop, 2008. GCE'08. [S.l.], 2008. p. 1–10.

FORMAN, George; ZAHORJAN, John. *The Challenges of Mobile Computing*. IEEE Computer, vol. 27, n. 4, p. 38-47, Apr., 1994.

GARFINKEL, S. L. *Architects of the Information Society: Thirty-Five Years of the Laboratory for Computer Science at MIT*. [S.l.]: TheMITPress, 1999.

IMIELINSKI, T.; BADRINATH, B. R. *Data Management for Mobile Computing*. In Sigmod Record, Vol. 22, No. 1, p. 34-39, 1993.

IMIELINSKI, T.; VISWANATHAN, S.; BADRINATH, B. R. *Energy efficient indexing on air*. In: *Proceedings of the 1994*

ACM Sigmod International Conference on Management of Data, 1994. ACM Press. p.25–36.

JING, Jin; Helal, ABDELSALAM S.; ELMAGARMID, Ahmed. *Client-server computing in mobile environments*. In *ACM Computing Surveys*, v.31, n.2, p.117–157, 1999.

JAVADI, B.; AKBARI, M. K.; ABAWAJY, J. H. A *performance model for analysis of heterogeneous multi-cluster systems*. *Parallel computing*, Elsevier, v. 32, n. 11, p. 831–851, 2006.

KRAUTER, K.; BUYYA, R.; MAHESWARAN, M. A *taxonomy and survey of grid resource management systems for distributed computing*. *Software: Practice and Experience*, Wiley Online Library, v. 32, n. 2, p. 135–164, 2002. (BRASILEIRO et al., 2008).

MATEUS, G. R.; LOUREIRO, A. A. F. *Introdução à Computação Móvel*. 11a Escola de Computação COPPE/Sistemas, 2008.

MELL, P.; GRANCE, T. *The NIST definition of cloud computing (draft)*. NIST special publication, v. 800, p. 145, 2011.

PERRY, Mark; O'HARA, Kenton; SELLEN, Abigail; BROWN, Barry; HARPER, Richard. *Dealing with mobility: understanding access anytime, anywhere*. *ACM Trans. Comput.-Hum. Interact.*, [S.l.], v.8, n.4, p.323–347, 2001.

PITOURA, E.; BHARGAVA, B. *Building information systems for mobile environments*. In: *Third International Conference on Information and Knowledge Management*, 1994. [s.n.].

PITOURA, E.; SAMARAS, G. *Data management for mobile computing*. In: Kluwer Academic Publishers, 1998. [s.n.]. v.10.

PONCIANO, L.; BRASILEIRO, F. *Assessing green strategies in peer-to-peer opportunistic grids*. Journal of Grid Computing, Springer, p. 1–20, 2012.

RIMAL, B.; CHOI, E.; LUMB, I. *A taxonomy and survey of cloud computing systems*. In: IEEE. INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. [S.l.], 2009. p. 44–51.

TANENBAUM, A. S.; STEEN, M. V. *Distributed systems*. [S.l.]: Prentice Hall, 2002. MEFFE; MUSSI; MELLO, 2006

ZHANG, Q.; CHENG, L.; BOUTABA, R. *Cloud computing: state-of-the-art and research challenges*. Journal of Internet Services and Applications, Springer, v. 1, n. 1, p. 7–18, 2010.