

Implementing a spontaneous social network for managing ubiquitous interactions

Dante Zaupa^{1,3}, Cristiano Costa¹, Jéssica Silva^{1,4}, Jorge Barbosa¹, Adenauer Yamin²

¹*Programa de Pós-Graduação em Computação Aplicada (PIPICA)*

Universidade do Vale do Rio dos Sinos (Unisinos)

²*Programa de Pós-Graduação em Computação (PPGC)*

Universidade Federal de Pelotas (UFPEL)

³*Bolsista de Iniciação em Desenvolvimento Tecnológico e Inovação do CNPq*

⁴*Bolsista de Iniciação em Desenvolvimento Tecnológico e Inovação da FAPERGS*

{dante.zaupa, jessicacristine0}@gmail.com, {cac, jbarbosa}@unisinos.br,
adenauer@inf.ufpel.edu.br

Abstract

The widespread of mobile devices has allowed people to have access to social networks and the variety of services wherever they are and whenever they want. In this scope, the idea of creating spontaneous social networks allow the interaction among users through the creation of mechanisms of communication and sharing services in specific places. In this sense, this article presents a model of a spontaneous social network based in the user's current location. The proposal, which is called Mingle, allows users to interact, access and offer services related to their current location, preferably through their mobile devices. A prototype was developed, composed of two components, an Android client and a server to manage the spontaneous social network. Based on the prototype, two evaluations were conducted. The first one was related to the mobile client and is based on a scenario. The second evaluation regards the server, and assesses its performance. The results are encouraging and show good potential for Mingle use and application.

1. Introdução

A evolução tecnológica e os avanços científicos têm aproximado a humanidade cada vez mais do conceito de computação ubíqua ou *ubicomp* [1]. Nesse cenário o computador integra-se a vida cotidiana, tornando-se parte indistinguível dela. Além disso, o acesso ao ambiente computacional ocorre de todo lugar e a todo o momento por meio de vários dispositivos [2].

As redes sem fio propiciam acesso à Internet nas áreas urbanas e a melhoria nas interfaces dos dispositivos móveis permite uma experiência cada vez mais próxima e integrada ao cotidiano. O próximo passo, já presente em algumas aplicações, é a percepção dos atributos do usuário e do ambiente, tais como a localização física, estado emocional e história pessoal. Essa ação é denominada sensibilidade ao contexto (em inglês *context awareness*), e é a propriedade que as aplicações devem incorporar para fazer uso das informações que caracterizam a situação das pessoas, lugares ou objetos [3].

A popularização dos dispositivos móveis aliado ao uso das informações de contexto permitem o desenvolvimento de aplicações que propiciem uma interação ubíqua: de todo o lugar, a todo o tempo e relacionada com os interesses do usuário e com a sua localização. Essa última característica possibilita o que tem sido denominado *Location Based Services - LBS* (Serviços Baseados em Localização), em que não somente a localização é informada, mas um conjunto de serviços contextualizados à mesma são oferecidos [4].

Para estimular essa interação, pode-se fazer uso das redes sociais. Diferentemente de grande parte das que existem hoje, este artigo propõe o desenvolvimento de uma rede social espontânea [5], formada pelas pessoas presentes em um local específico e que se conectam a ela através de seus dispositivos móveis.

Para tirar maior proveito dos LBS, pode-se utilizar uma conexão a uma rede de computadores para integrar as pessoas fisicamente presentes naquele local em uma rede social [6]. Existem alguns estudos que já demonstram a convergência dessas duas redes [7].

Além da construção de espaços virtuais e do estímulo a auto-expressão, as redes sociais têm alterado nosso estilo de comunicação, apesar de manter os princípios da interação social humana [7]. Tem-se como premissa neste trabalho que nada mais natural, e portanto mais próximo do conceito de *ubicomp*, do que aliar a oferta de serviços baseados em localização com a formação de uma rede social espontânea, ou seja, uma rede social formada por pessoas presentes em locais específicos utilizando dispositivos móveis [8].

Nesse âmbito, o objetivo desse artigo é apresentar um modelo, denominado *Mingle*, que tem por objetivo gerenciar a interação ubíqua em uma rede social dinâmica, com o mínimo de infraestrutura, formada pelas pessoas que estão fisicamente presentes em determinados locais. O principal resultado esperado consiste no desenvolvimento de um sistema de interação ubíqua, funcional, e que possa ser aplicado a diversos cenários reais, tanto com exploração comercial como puramente social. Para avaliação do modelo, foram implementados um cliente móvel para dispositivos Android e um servidor da rede social espontânea, que permite todas as interações relacionadas com uma célula. O primeiro foi avaliado através de um cenário de uso, enquanto o segundo foi avaliado quanto ao desempenho.

O artigo está organizado em 5 seções. A seção 2 descreve trabalhos relacionados. O modelo proposto para o *Mingle* é descrito na seção 3. A seção 4 apresenta detalhes de implementação e as duas avaliações realizadas, no cliente e no servidor. Por fim, a seção 5 apresenta algumas conclusões do trabalho e encaminhamentos para trabalhos futuros.

2. Trabalhos Relacionados

Atualmente, existem diversos trabalhos que exploram os conceitos de redes sociais aplicados à área de computação móvel e ubíqua. Dentre esses trabalhos, foi buscado analisar aqueles que são os mais representativos e também os que possuem alguma característica de espontaneidade, ou seja, formam redes sociais relacionadas com contextos específicos.

O *MobiClique* [9] define uma rede social espontânea criada a partir do encontro oportunístico de dispositivos móveis. Através do uso de Bluetooth, o sistema permite que a rede social seja formada, a partir do encontro de dispositivos móveis, eliminando a necessidade de um servidor central.

Outro trabalho relacionado é o *SCOPE* [10] que propõe uma rede social espontânea P2P. Assim como o *MobiClique*, o *SCOPE* não necessita de uma infraestrutura pré-estabelecida, porém baseia-se no uso do protocolo ad hoc 802.110 foco do projeto é em

compartilhar informações e conteúdos, não prevendo o suporte a serviços.

O *MobiSoC* [11], por sua vez, é um middleware que permite capturar, gerenciar e compartilhar estados sociais de comunidades. O projeto incorpora algoritmos de descoberta que detectam padrões geo-sociais emergentes para caracterizar os estados sociais representados. O projeto busca realizar associação entre pessoas presentes no mesmo local e colaboração espontânea baseada nesse lugar.

Por fim, foi considerado o trabalho *SenseWorld* [12]. O projeto foca na aplicação de redes sociais para construir aplicações de sensoriamento. Além de conectar pessoas a componentes de software, sensores são envolvidos para detectar informações contextuais. O *SenseWorld* explora diversas informações contextuais sem focar na interatividade espontânea dos indivíduos nem no compartilhamento de serviços e conteúdos.

3. O Modelo Mingle

A proposta central do projeto *Mingle* é usar como elemento integrador de uma rede social a própria localização física do indivíduo. As pessoas presentes em um determinado lugar, aqui denominado de célula, passam a fazer parte de uma rede social espontânea, criada especificamente naquela localização e que irá integrar as pessoas presentes naquele momento.

A partir da rede social, usuários podem acessar um conjunto de conteúdos e serviços virtuais relacionados com o local específico onde se encontram. Além disso, podem interagir entre si. A forma de acesso e uso da rede social é realizada preferencialmente através do próprio telefone celular da pessoa. Outras formas de acesso também são possíveis, através de algum outro dispositivo móvel (como *tablets* ou *notebook*) ou estacionário (como um computador desktop ou terminais de autoatendimento com tela sensível ao toque).

As pessoas presentes na célula não precisam inicialmente se identificar para os demais participantes. Posteriormente, elas podem decidir por informar a sua localização ou identidade para pessoas específicas da célula. Cada célula possui também um nodo *base* que oferece serviços e conteúdos específicos relacionados com aquele local. Além disso, as próprias pessoas que conectam-se na rede social também podem oferecer serviços e conteúdos gerados pelo próprio usuário. A disponibilização dos conteúdos gerados pelo usuário é considerada hoje um fator que impulsiona a revolução da computação móvel [13].

Cada pessoa constrói o seu perfil definindo os serviços e conteúdos oferecidos. Além disso, informa as preferências pessoais e interesses. Tais informações são armazenadas em uma ontologia [14][15]. Essa ontologia é empregada tanto para a criação de uma base de conhecimento das pessoas e das células, quanto para a inferência de informações que facilitem a interação entre essas pessoas.

No Mingle a sensibilidade ao contexto é uma característica-chave para transformar a relação entre as pessoas em uma interação ubíqua. Além da localização física e das preferências pessoais, outras informações de contexto são consideradas pelo modelo, tais como informações temporais, contexto social, contexto histórico e condições ambientais.

Existem dois tipos de Células previstas no Mingle. *Células públicas* são aquelas que permitem liberdade total na interação entre pessoas, não realizando nenhum tipo de intermediação ou de arbitragem entre os usuários da rede. Por outro lado, *células privadas* realizam a intermediação entre as pessoas. Com isso, é possível controlar os tipos de serviços e de conteúdos oferecidos pelos usuários e também gerenciar tal operação.

A Figura 1 mostra um exemplo de uma célula privada do Mingle em um Shopping Center, com pessoas ligadas à ela através de seus dispositivos computacionais. Dentro da célula, estão listadas algumas perguntas que com o Mingle se busca responder.

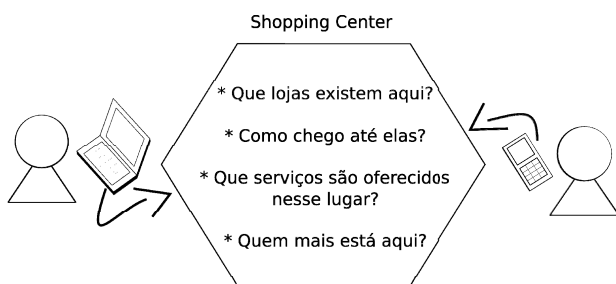


Figura 1. Exemplo de célula privada do Mingle para um Shopping Center e possíveis perguntas que o modelo pode ajudar a responder.

A Figura 2 ilustra a arquitetura do Mingle com o fluxo de trabalho típico do modelo. O cliente se conecta ao servidor e envia um script contendo um conjunto de operações, tipicamente de escrita ou consulta à base de dados. O servidor instancia um *manager*, que é escolhido pelo conjunto de dados sendo acessado, para processar esse script. Ao final, caso tenha havido uma consulta, o *manager* retorna um novo script, que contém os dados solicitados pelo cliente.

Na Figura 2 está indicado o fluxo de trabalho do Mingle, subdividido em dez passos. O processo começa quando um cliente solicita conexão a uma célula do Mingle (Passo 1). Quando essa solicitação é aprovada (Passo 2), o cliente envia então um roteiro, no formato JSON, criado automaticamente a partir das solicitações do usuário, e que descreve o conjunto de dados que ele está solicitando, assim como operações de filtragem sobre esses dados (Passo 3). A filtragem do conjunto de dados ocorre no servidor, servindo a um duplo propósito: reduzir a quantidade de processamento no cliente, tipicamente um smartphone, e reduzir o impacto da transmissão do conjunto de dados solicitado na rede, uma vez que esse conjunto pode ser consideravelmente grande. Após receber o script um Gateway instancia um novo MingleManager (Passo 4), que vai selecionar o gerenciador adequado para processar o script (Passos 5 e 6). O gerenciador é tipicamente uma classe que implementa a interface MingleManager. Esse gerenciador interpreta as operações contidas no script e executa as consultas e/ou alterações no banco de dados (Passos 7, 8 e 9). Os dados recebidos são então guardados em um novo arquivo, também no formato JSON, que é enviado pelo gerenciador de volta para o Cliente (passo 10).

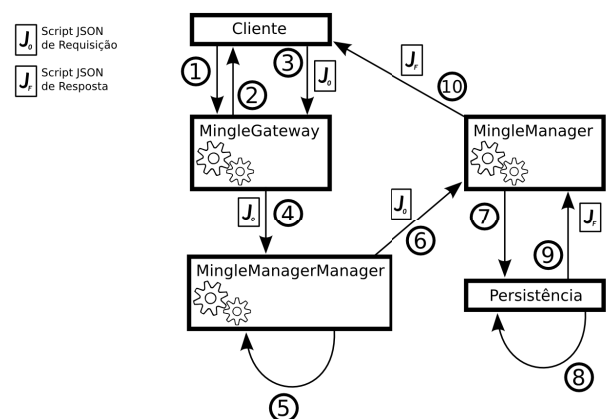


Figura 2. Arquitetura do Mingle, mostrando os principais componentes e também o fluxo de trabalho típico do modelo.

O Mingle foi projetado com uma ontologia para armazenar as informações da rede social. A ontologia do Mingle é apresentada na Figura 3. Na ontologia estão representadas as principais entidades relacionadas com o modelo. A classe *Person* representa os usuários do Mingle que estão conectados em uma Célula. Uma especialização da classe *Person* representa um usuário que é administrador do sistema (classe *Administrator*). Além disso, cada pessoa no

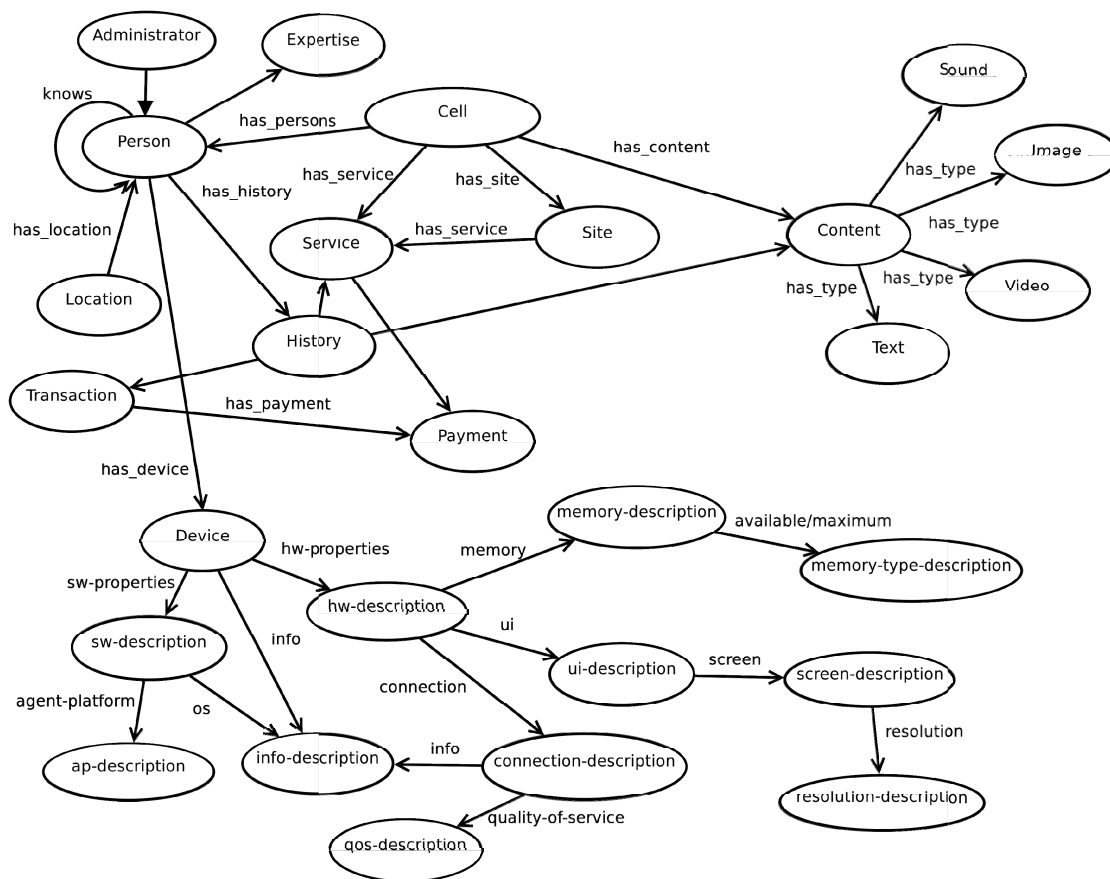


Figura 3. Ontologia do Mingle

Mingle tem associada a sua história (representada pela classe *History*), as suas especialidades (classe *Expertise*), a sua localização (classe *Location*) e o seu dispositivo (classe *Device*).

A classe que representa o dispositivo traz todas as informações relacionadas, como o detalhamento do software (*sw-description* e suas instâncias) e do hardware (*hw-description* e suas instâncias). Essas informações são relevantes para a sensibilidade ao contexto e também para a adaptação dos conteúdos e serviços ao dispositivo que o usuário tem disponível.

As células são representadas pela classe *Cell*. Essa classe, além de ter associada todas as pessoas presentes, disponibiliza serviços (classe *Service*) e conteúdos (classe *Content*). Uma célula pode ser ainda subdivida em espaços menores, chamados no Mingle de sítios (classe *Site*). Por exemplo, em um shopping poderíamos ter sítios específicos para cada loja ou para representar a praça de alimentação. Os conteúdos podem ser de vários tipos: sons (classe *Sound*), imagens (classe *Image*), vídeos (classe *Video*) e textos (classe

Text). Por fim, o uso de um serviço pode envolver uma transação financeira (classe *Transaction*) e incluir um pagamento (classe *Payment*).

4. Implementação e Avaliação

Uma implementação do Mingle foi realizada, tanto do servidor quanto de um cliente Android. O servidor foi desenvolvido na linguagem Java Enterprise Edition (J2EE) e utiliza o servidor de aplicações JBoss¹. Para descobrimento de serviços, é usado o Avahi², um sistema de DNS multicast. Cada célula Mingle divulga-se como um serviço no Avahi, que é lido pelo cliente Android e usado para selecionar à qual célula o usuário está se conectando. A ontologia foi modelada no Protégé³ e é armazenada em OWL. Para ler os

¹ <http://www.jboss.org/>

² <http://www.avahi.org/>

³ <http://protege.stanford.edu/>



Figura 4. Telas da execução do cenário de avaliação do Mingle: (a) Escolha da Célula; (b) Opções disponíveis; (c) Cadastro de serviço; (d) Serviços disponíveis.

dados desse arquivo, a biblioteca Jena⁴ foi utilizada. Para a camada de persistência com banco de dados foi usado o PostgreSQL⁵. E por fim, para a interpretação de arquivos JSON foi usada a biblioteca de código aberto Jackson⁶.

Foram realizadas duas avaliações no protótipo desenvolvido. O cliente Android foi avaliado a partir de um cenário de uso. Já o servidor foi avaliado quanto a escalabilidade, avaliando a carga suportada.

4.1. Avaliação através de Cenário de Uso

A comunidade científica tem utilizado cenários para avaliar projetos de computação móvel, ubíqua e sensível ao contexto [3][16][17]. Para avaliação do protótipo foi empregado o seguinte cenário:

“Adriana é aluna de Computação de uma Universidade e usuária de um smartphone com Android. Ela tem instalado no seu celular o cliente do Mingle e ao chegar na Universidade conecta-se na célula local da rede social. Uma vez conectada, ela visualiza todas as opções disponíveis na célula, tais como conteúdos (dicas, eventos, mapa), serviços e pessoas presentes naquele local. Adriana então lembra que tem dois livros que adquiriu para usar em disciplinas já cursadas e que não deseja mais. Por causa disso, Adriana cria um serviço para vender esses

livros. Adriana opta então por visualizar os serviços da célula, para ver se o serviço que criou foi adicionado corretamente. Ela verifica que o serviço já está disponível. Além disso, Adriana descobre que existe um serviço gratuito para um grupo de estudos de Java. Ela obtém mais detalhes do serviço, pois está justamente tendo dificuldades nessa linguagem.”

A execução do cenário foi realizada com um celular Samsung Galaxy S II executando o sistema operacional Android 2.3. A execução ocorreu conforme esperado, tendo o Mingle realizando as ações conforme previsto no Cenário. A Figura 4 mostra algumas telas capturadas do protótipo criado para executar o cenário: (a) tela que permite a escolha da célula (poderiam estar disponíveis mais de uma célula em um determinado local); (b) tela que apresenta as opções disponíveis na célula; (c) tela de criação de serviço, na qual o serviço de venda de livros usados é adicionado; (d) tela que lista os serviços disponíveis na célula, incluindo a “Venda de Livros Usados” recém adicionada e o “Grupo de Estudos Java” já disponibilizado anteriormente.

4.2. Avaliação de Desempenho do Servidor

Para efetuar a avaliação do servidor do Mingle foi criada uma máquina virtual gerada pelo software Oracle VirtualBox⁷. Essa máquina virtual teve um núcleo do processador do hospedeiro, um Intel Core 2 Duo com 2.93GHz, com 1296MB de memória RAM e

⁴ <http://jena.sourceforge.net/>

⁵ <http://www.postgresql.org>

⁶ <http://wiki.fasterxml.com/JacksonHome>

⁷ <http://www.virtualbox.org>

foi ligada ao hospedeiro através de uma placa Ethernet virtual configurada em modo ponte, ou seja, como uma máquina independente na rede. Nessa máquina virtual foi instanciado um servidor do Mingle representando a célula “Mingle_Test_Local”.

Para a avaliação do desempenho do servidor, foram gerados gráficos no *Gnome System Monitor* que representam o uso de CPU (em porcentagem de capacidade) e rede (em KB/s) no servidor de testes criado. Todas as medições foram repetidas 10 vezes e o gráfico apresentado é da média da avaliação. Não houve desvio padrão significativo em nenhum dos testes. Cada uma das repetições foi monitorada por 60 segundos.

Os gráficos da Figura 5 mostram a carga de CPU e rede da máquina de testes antes da execução do servidor Mingle. Analisando os dados observa-se que a sobrecarga do sistema é em torno de 40%, considerando apenas a execução do sistema operacional e do próprio VirtualBox. Além disso, há um tráfego de rede que oscila entre 0 e 3 kb/s.

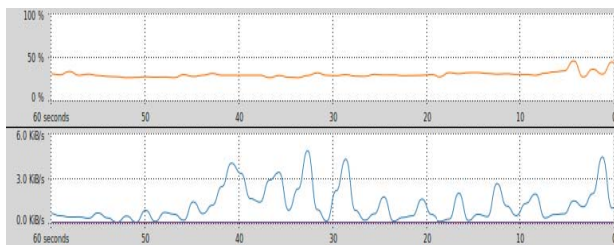


Figura 5. Gráfico mostrando a utilização de CPU e rede da máquina de testes ociosa, i.e., antes do Mingle ser executado.

A Figura 6 mostra a máquina virtual com o servidor esperando conexão por 50 segundos, e mostra que existe um pequeno intervalo de maior processamento no momento de execução, mas que logo após o sistema volta ao mesmo nível de processamento do estado anterior. Durante 10 segundos (entre o intervalo 50 e 40) a carga ultrapassa os 50%, chegando ao pico de 63%.

A Figura 7 mostra o servidor escrevendo 200 linhas de dados em uma tabela da base de conhecimento. Ao terminar, o Mingle extrai os dados da tabela, armazena em uma estrutura de dados temporária em memória RAM, e envia para o Cliente. O gráfico mostra que em termos de rede a operação não onerou a máquina virtual. Em termos de CPU, houve uma carga temporária de 100%, durante cerca de 5 segundos enquanto a tabela estava sendo montada na memória RAM. Fora esse intervalo, a carga manteve-se estável e próxima dos valores obtidos nas avaliações anteriores.

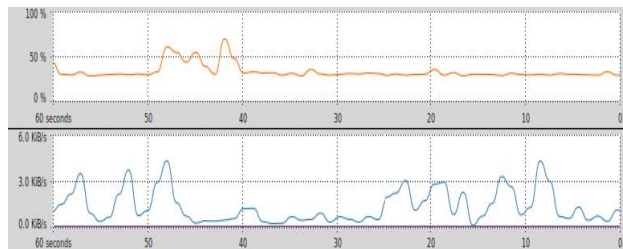


Figura 6. Gráfico mostrando a utilização de CPU e rede da máquina de testes, com o servidor esperando conexão. A máquina foi iniciada no segundo 50.

Por fim, a Figura 8 mostra o servidor atendendo a uma requisição de 5000 operações de escrita de dados na persistência, em uma tabela que não continha nenhum registro. Após a escrita, o servidor leu os mesmos dados, armazenou em uma estrutura de dados temporária em memória RAM, e enviou para o Cliente. O impacto dessas ações na rede não é grande, porém em termos de CPU novamente a carga aumentou durante o processo de construção da tabela na memória RAM. Esse processo durou cerca de 10 segundos, tendo a carga oscilando entre 85% e 100%.

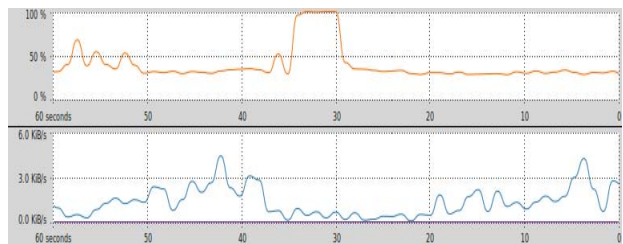


Figura 7. Gráfico mostrando a utilização de CPU e rede da máquina de testes, com o servidor processando uma escrita de 200 linhas de dados, seguida por uma operação de leitura e uma de envio dos dados solicitados. O processo começa na marca de 40 segundos.

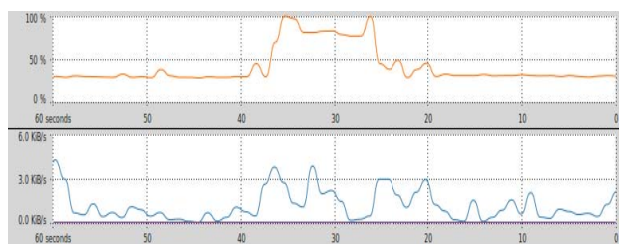


Figura 8. Gráfico mostrando a utilização de CPU e rede da máquina de testes, com o servidor processando um script de escrita de 5000 linhas de dados, seguida por uma operação de leitura e uma de envio dos dados solicitados. O processo começa na marca de 40s.

Os experimentos realizados mostram que, mesmo em uma máquina com pouco poder de processamento, sob grande carga, o servidor Mingle apresentou resultados satisfatórios. Apesar de ter alguns momentos de sobrecarga, em que o uso do processador chegou em 100%, é necessário ponderar o hardware limitado que foi empregado nos experimentos. Os gráficos também mostraram que o impacto na rede não é significativo. Os resultados são encorajadores e sinalizam que um hardware com maior capacidade de processamento e memória RAM deve ser capaz de servir a um grande número de clientes.

5. Conclusão

A recente popularização de dispositivos computacionais móveis, que vêm no formato de *smartphones*, *tablets*, *netbooks* e *notebooks*, abre possibilidades antes inexistentes. Hoje, com ampla disponibilidade de dispositivos e de conexão entre eles, é possível aproveitar-se das capacidades dos mesmos para criar aplicações *ad hoc*. O Mingle é um projeto surgido dessa possibilidade.

Com a ideia de permitir a interação entre as pessoas e o ambiente, foi modelado o Mingle baseado em quatro perguntas básicas: “Quem está aqui?”, “Onde posso ir?”, “O quê posso extrair desse ambiente?” e “Quando acontecerá um evento de meu interesse aqui?”. Essas perguntas permitem inspecionar possibilidades oferecidas pela célula.

Comparado com os trabalhos relacionados estudados, o Mingle permite amplas possibilidades de conexão, não ficando restrito a Bluetooth ou NFC, traz mais possibilidades que o simples compartilhamento de informações e permite que os usuários criem seus próprios conteúdos.

O Modelo do Mingle foi dividido em vários módulos, cada um responsável por um conjunto de dados diferente, que servem ao cliente como persistência de dados e como processador auxiliar, filtrando os dados e reduzindo a quantidade de processamento e armazenamento por parte do cliente. Essa característica é especialmente desejável quando o cliente é um *smartphone*, por causa da capacidade limitada de processamento e de bateria.

Os experimentos indicaram que essa transferência de responsabilidade não faz com que o servidor necessite de um processador poderoso ou de grandes quantidades de memória para atuar em um ambiente com poucos usuários. Além disso, o cenário desenvolvido permitiu avaliar a aplicabilidade do modelo e sua utilização através de um smartphone Android.

Como trabalhos futuros, pretende-se expandir o servidor para suportar diferentes células. Além disso, a possibilidade de troca de informações entre as células, compartilhando usuários e históricos deve ser estudada. Por fim, pretende-se expandir os testes realizados aplicando o Mingle em um ambiente real e testando o uso do servidor com cargas geradas a partir desse uso.

Agradecimentos

Agradecemos à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) por parcialmente financiar o presente projeto de pesquisa. Também agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por ter concedido bolsas ao projeto. Por fim, gostaríamos de agradecer a Universidade do Vale do Rio dos Sinos (Unisinos), e particularmente ao Programa de Pós-Graduação em Computação Aplicada (PIPCA) pelo suporte e apoio à execução da pesquisa.

Referências

- [1] Weiser, M. The computer for the 21st century. *Scientific American*, v. 265, n. 3, pp 94-104, 1991.
- [2] C.Costa, A.Yamin, and C.Geyer, Toward a general software infrastructure for ubiquitous computing, *IEEE Pervasive Computing*, vol. 7, pp. 64–73, 2008.
- [3] A. Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
- [4] Vaughan-Nichols, S. Will mobile computing’s future be location, location, location? *Computer*, 42(2):14–17, 2009.
- [5] Huang, J.; Liu, Q.; Lei, Z.; Chiu, D., Applications of Social Networks in Peer-to-Peer Networks, In: : Abraham, A and Hassanien, A (Eds.), *Computational Social Networks*, London, Springer, pp. 301-327, 2012
- [6] Watts, D. J., *Six degrees: The science of a connected age*. New York, W. W. Norton & Company, 2002
- [7] J. Kleinberg. The convergence of social and technological networks. *Communications of the ACM*, 51(11), Nov 2008.
- [8] B.G.D. Silva, C.A.D. Costa, D. Zaupa, and G. Freitas, An ontology-based repository for a spontaneous social network, *Proc. CASON, IEEE*, 2011, pp.227-232.
- [9] A.-K. Pietilainen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot. Mobiclique: middleware for mobile social networking. *WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks*, Aug 2009.
- [10] M. Mani, A.-M. Nguyen, and N. Crespi. Scope: A prototype for spontaneous p2p social networking. *Per- vasive*

- Computing and Communications Workshops (PER- COM Workshops), 2010 8th IEEE International Conference on DOI - 10.1109/PERCOMW.2010.5470664, pages 220– 225, 2010.
- [11] A. Gupta, A. Kalra, D. Boston, and C. Borcea. Mobisoc: a middleware for mobile social computing applications. *Mobile Networks and Applications*, 14(1), Feb 2009.
- [12] R. Ganti, Y.-E. Tsai, and T. Abdelzaher. Senseworld: Towards cyber-physical social networks. *Information Processing in Sensor Networks*, 2008. IPSN '08. International Conference on DOI - 10.1109/IPSIN.2008.48, pages 563–564, 2008.
- [13] Jensen, C., Vicente, C., and Wind, R. (2008). User-generated content: The case for mobile services. *Computer*, 41(12):116–118.
- [14] N. Noy and D. McGuinness. *Ontology development 101: A guide to creating your first ontology*. 2001.
- [15] I. Horrocks. Ontologies and the semantic web. *Commun. ACM*, 51:58–67, December 2008.
- [16] Satyanarayanan, M. Pervasive computing: vision and challenges. *Personal Communications*, IEEE, Pittsburgh, v. 8, n. 4, August 2001, pp. 10-17.
- [17] Satyanarayanan, M. Mobile Computing: the next decade. *ACM Workshop on Mobile Cloud Computing & Services*, Proceedings, San Francisco, July 2010. pp. 1-6.