

4 Testes com o Serviço de Matchmaking

O presente capítulo apresenta os testes que foram realizados com o serviço de *matchmaking* os quais tiveram o objetivo de verificar a viabilidade técnica do serviço. Foram realizados três testes básicos: análise dos resultados de *matching* que visou analisar o comportamento do algoritmo *Pearson R* dentro do escopo deste problema; análise do tempo de resposta que verificou a latência do MMS e, finalmente, análise de escalabilidade que, por meio de simulações, objetivou determinar um limite máximo de usuários que o serviço pode atender.

4.1. Ferramenta para Simulação

A arquitetura MoCA disponibiliza um simulador que permite emular Monitores para dispositivos móveis (com interface de rede 802.11) em computadores de mesa, de forma a permitir a simulação da movimentação e detecção de localização de vários usuários hipotéticos, em uma área de interesse. Da mesma forma como o Monitor real, o monitor simulado (Monitor/Sim) envia as informações de contexto (incluindo a intensidade de sinal de rádio de supostos Access Points 802.11) para o servidor de contexto CIS. O serviço de localização LIS utiliza esta informação coletada no CIS para inferir a posição do dispositivo de cada usuário em termos das regiões simbólicas, cadastradas e mapeadas pelo LIS. Finalmente, o MMS interage com o LIS para obter diretamente as informações de localização de um usuário, o conjunto de usuários presentes em uma região simbólica, ou para ser notificado de qualquer uma mudança de localização de algum usuário. A Figura 13 ilustra este processo.

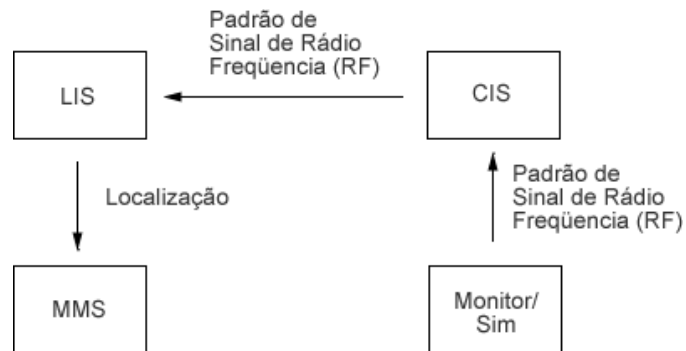


Figura 13 - Processo de simulação

Para realizar os ensaios de teste, é necessário primeiro configurar o Monitor/Sim. Este disponibiliza alguns padrões de sinais de RF que seriam enviados por monitores reais ao CIS em diferentes áreas da PUC-Rio, como por exemplo: *Espaço Aberto*, *Restaurante*, etc. Para simular que um dispositivo está localizado em uma destas áreas do campus da PUC, precisa-se copiar o padrão de frequência disponibilizado para um arquivo e, posteriormente, editá-lo a fim de incluir o endereço MAC do dispositivo móvel simulado. Depois de criados um ou mais arquivos deste tipo deve-se ajustar o Monitor/Sim, por meio do seu arquivo de configuração, para que esse(s) arquivo(s) contendo os padrões de RF seja(m) lido(s). Podem-se simular vários dispositivos de usuários localizados em diversas regiões. Para isto basta criar um arquivo que contenha vários destes padrões.

Como ilustração, o Quadro 3 apresenta um padrão do sinal de rádio frequência correspondente à área no campus da PUC-Rio situada entre os prédios da Fundação Padre Leonel Franca e do Rio Data Centro. Em destaque, o MAC do dispositivo simulado.

```

7#12888#63#0#0#0#10.10.10.24#255.255.255.0#00:02:2D:A5:06:45#00:
02:2D:A5:06:12&00:02:2D:A5:06:12#-77#LAC&00:0E:84:80:78:5E#-
78#Wi-Fi PUC&08:00:46:09:BE:67#-90#tecfplf&08:00:46:09:BE:67#-
90#tecfplf
  
```

Quadro 3 - Exemplo de padrão de sinal de Rádio Frequência disponível pelo simulador do monitor

Para simular, por exemplo, a movimentação de um dispositivo por entre três regiões simbólicas da PUC, podem-se criar três arquivos, onde cada um apresente um padrão de sinal de RF correspondente a uma região simbólica, e ajustar o Monitor/Sim para ler estes arquivos em instantes de tempo diferente. O exemplo do Quadro 4 apresenta um arquivo de configuração do Monitor/Sim onde é ajustada a leitura de três arquivos em instantes de tempo diferentes (através do atributo *interval*). Desta maneira, é possível simular a movimentação dos usuários entre as regiões simbólicas caracterizadas pelos padrões de frequência contidos nos arquivos.

```
monitor.scanInterval=1000
monitor.repeating=true
cis.server.host=localhost
cis.monitor.port=55010
file1=C:\\Moca\\monitor-rc1c\\conf\\Instante1Area1.txt
interval1=50000
file2=C:\\Moca\\monitor-rc1c\\conf\\Instante2Area2.txt
interval2=50000
file3=C:\\Moca\\monitor-rc1c\\conf\\Instante3Area3.txt
interval3=50000
```

Quadro 4 - Exemplo de propriedades do simulador do monitor

A próxima seção apresenta uma análise dos resultados de *matching* gerados pelo MMS. Esta análise teve como objetivo verificar o quanto o algoritmo *Pearson R* é preciso na identificação de usuários com perfis similares. Neste primeiro teste, não foi levada em consideração a mobilidade de usuários.

4.2. Análise dos Resultados de Matching

Para realizar a análise dos resultados de *matching*, construiu-se uma base de conhecimento com onze usuários e dez instâncias genéricas de interesses (A-J). Atribuiu-se um nível de interesse para cada instância de interesse de um usuário. Neste teste, supôs-se que estes interesses têm níveis iguais para todas as regiões simbólicas, ou seja, todos os interesses foram configurados como *allLocation*. Desta forma, garante-se que todos os níveis de interesse sejam levados em conta, independentemente da localização dos respectivos usuários. A tabela abaixo apresenta a distribuição de valores dos níveis de interesse para os onze usuários, onde valores entre 1 e 4 devem ser interpretados como indicando baixa relevância do interesse, valores entre 3 e 7 como relevância

média, e valores acima de 6 como alta relevância da instância de interesse. A sobreposição dos intervalos de valores foi proposital, e tentou modelar o grau de subjetividade sobre o que cada usuário considera uma relevância alta, média ou baixa. A Tabela 5 apresenta a distribuição de valores para os onze usuários.

	A	B	C	D	E	F	G	H	I	J
1	10	9	9	8	3	6	5	2	2	2
2	9	9	9	9	6	7	4	3	4	1
3	8	8	9	10	10	4	6	3	2	4
4	9	10	8	7	4	4	5	6	6	7
5	3	4	5	6	10	10	10	3	2	4
6	4	4	6	5	9	9	8	2	9	3
7	3	10	1	2	9	9	10	2	4	4
8	1	3	4	2	4	3	1	9	10	9
9	5	7	2	4	2	3	4	10	8	9
10	1	2	4	2	1	2	3	8	8	8
11	1	1	1	1	1	9	1	10	10	10

Tabela 5 – Distribuição dos valores de teste

A distribuição dos valores foi feita de modo a criar grupos de usuários com afinidades de interesses. Os usuários foram agrupados em três conjuntos. O primeiro conjunto é formado pelos usuários {1,2,3,4}, o segundo grupo foi formado pelos usuários {5,6,7}, e o terceiro grupo pelos usuários {8,9,10,11}.

Assume-se que, para o primeiro grupo de usuários, os interesses {A, B, C, D} possuam alta relevância. Já os interesses {E, F, G} foram considerados de relevância média e os {H, I, J} foram considerados de baixa relevância. Para o segundo grupo de usuários, consideram-se os interesses {A, B, C, D} de relevância média, os interesses {E, F, G} de relevância alta e os interesses {H, I, J} de relevância baixa. Finalmente, para o terceiro grupo de usuários, consideram-se os interesses {A, B, C, D} de relevância baixa, os interesses {E, F, G} também de relevância baixa e os interesses {H, I, J} de relevância alta.

As posições destacadas em cinza, na tabela acima, representam comportamentos aleatórios que visam simular situações onde um usuário possui outros interesses não compartilhados com os demais membros do seu grupo tradicional. Não foram utilizados valores aleatórios nos conjuntos de instâncias de interesse que caracterizam cada grupo, pois alterariam os resultados de forma significativa.

A fim de verificar se os resultados gerados pelo algoritmo implementado realmente estavam corretos, realizaram-se cálculos manuais. Os resultados obtidos desta simulação são apresentados na seção a seguir.

4.2.1. Resultados

A primeira conclusão, ou melhor, verificação, foi que o algoritmo de correlação *Pearson R*, implementado no MMS, estava correto. O algoritmo implementado acertou todos os testes em comparação aos cálculos que foram realizados manualmente.

Outra comprovação encontrada confirma que o *Pearson R* realmente é adequado para identificar a similaridade entre perfis de usuários. A seguir, serão apresentados alguns resultados que justificam esta afirmativa. A Tabela 6 apresenta os resultados de *matching* obtidos para o usuário 1, isto é, seus principais parceiros com afinidade de interesses:

Usuários	Resultados
2	90.84%
3	64.93%
4	63.80%

Tabela 6 – Resultados obtidos para o usuário 1

O maior valor de correlação foi obtido para os usuários 1 e 2, pois, como pode ser visto na Tabela 5, entre os valores atribuídos para 2 não foi modelado nenhum comportamento divergente. Os usuários 3 e 4 apresentam comportamentos divergentes e, por isso, obtiveram valores de correlação menores, embora ainda altos. Assumindo que um valor de correlação acima de 60% pode ser interpretado como caracterizando uma afinidade de perfis, nota-se portanto que o MMS apresentou um resultado satisfatório para o usuário 1, pois identificou os usuários com perfis de interesse mais similares.

Apesar do *resultado de matching* do MMS apresentar o interesse que obteve a maior similaridade, por exemplo, para os usuários 1 e 2 seriam os interesses *B* ou *C*, o MMS não é apropriado para identificar uma alta afinidade com relação a um único interesse. Isto acontece, pois o *Pearson R* calcula a correlação linear entre todos os valores de níveis de interesses indicados pelos usuários. Entretanto, suponha-se que exista mais um usuário na Tabela 5 (de

número 12) que tenha indicado o valor de interesse 10 para o tópico A e valor de 1 para todos os demais interesses. Mesmo neste caso onde o usuário 12 indicou somente um alto grau de interesse por A, o MMS encontraria valores de correlação positivos entre o usuário 12 e os usuários {1, 2, 3, 4}, ou seja, os usuários que também indicaram alto grau de interesses por A. Portanto, este exemplo demonstra a capacidade do *Pearson R* de identificar usuários com perfis de interesse similares.

Os resultados obtidos para o usuário 2 são apresentados na Tabela 7. Novamente, o MMS encontrou usuários com perfis similares a 2, ou seja, {1,3,4}. Como pode ser observado nas tabelas, o valor de *matching* do *Pearson R* é simétrico, para quaisquer dois usuários.

Usuário	Resultados
1	90.84%
3	72.31%
4	45.53%
6	7.12%
5	6.28%

Tabela 7 - Resultados obtidos para o usuário 2

Note-se que o MMS também encontrou correlação baixa entre o usuário 2 e os usuários 6 e 5. Provavelmente, isto se deve ao fato de o usuário 2 indicar relevância média dos tópicos {E, F, G}, enquanto que usuários 5 e 6 apresentam relevância alta para este conjunto. Analisando os resultados da Tabela 7 fica claro que o *Pearson R* indica quais são os usuários no grupo de interesse de usuário 2.

Resumindo, nota-se que para todos os usuários, o MMS identifica corretamente os seus respectivos grupos de interesse. Entretanto, o usuário 11 apresenta um caso interessante, pois possui interesses com alta relevância para os interesses {H, I, J}, mas também apresentou um comportamento fora do padrão para o interesse F. Mesmo assim, o MMS encontrou os usuários {8,9,10} do grupo 11, com valores de correlação significativos. A Tabela 8 apresenta os resultados para o usuário 11.

Usuário	Resultados
8	82,07
10	79,15
9	65,69
6	-0.07
7	-0.18
5	-0.29
4	-0.34
2	-0.70
1	-0.71
3	-0.91

Tabela 8 – Resultados obtidos para o usuário 11

Foge ao escopo desta dissertação apresentar uma análise detalhada sobre o desempenho do *Pearson R*, uma vez que tal análise já foi feita em outros trabalhos (Shardanand, 1994). No entanto, com base nos resultados dos testes feitos com o MMS, acredita-se que o algoritmo escolhido apresente resultados satisfatórios no que diz respeito à detecção de similaridades entre perfis de usuários.

4.3. Análise do Tempo de Resposta

Estes testes tiveram como objetivo medir o tempo de processamento necessário para atender às requisições dos usuários e avaliar o quanto o mecanismo de *caching* melhorou o tempo de resposta. Para realizar estes testes, instrumentou-se o código com chamadas de sistema, para obtenção do horário corrente, no início e no fim dos métodos relacionados com o processo de *matching* (*matchingOneToMany*).

Novamente, o MMS foi configurado com os onze usuários da Tabela 5 posicionados somente em uma região simbólica. Então, utilizando o cliente MMS, realizaram-se consultas e mediu-se o tempo de resposta do sistema. Foram criados três cenários de teste. No primeiro, desabilitou-se a *cache* do servidor MMS. No segundo, foi carregada apenas metade dos *objetos pessoa*¹¹ (objetos que representam os perfis dos usuários) na *cache*. No terceiro, carregaram-se todos os *objetos pessoa* na *cache* do sistema. Para cada cenário executou-se o teste dez vezes. As médias dos resultados são apresentados a seguir¹².

¹¹ Uma explicação mais detalhada sobre os *objetos pessoa* pode ser observada na seção 3.1

¹² Os testes foram executados em um Athlon 1.6 com 512M de memória randômica.

4.3.1. Resultados

Os resultados obtidos são apresentados na Tabela 9.

Número de Objetos Máximo da Cache	Tempo Médio (em segundos)
0	43.42
6	21.60
11	0.12

Tabela 9 – Resultados do tempo de Resposta

Como se pode observar, a ausência de um sistema de *cache* tornaria a utilização do MMS inviável. Para comparar o perfil de onze usuários, o MMS gastaria aproximadamente 44 segundos, enquanto que, para cem usuários, seriam gastos aproximadamente 6.6 minutos.

No entanto, carregando-se na *cache* metade do número total de objetos, ou seja, para onze usuários a *cache* conteria 6 objetos; o servidor MMS em média gasta aproximadamente 22 segundos para atender a cada requisição. Em um cenário com cem usuários, o MMS gastaria aproximadamente 3.3 minutos. Como mencionado na seção 3.1 do capítulo 3, a *cache* foi implementada usando uma tabela *hash* com uma remoção aleatória de objetos quando a tabela está cheia. A ausência, no MMS, de uma política otimizada de gerenciamento de *cache* deve ter sido a causa do mau resultado obtido.

No entanto, no cenário em que todos os objetos foram mantidos na *cache*, o tempo de resposta caiu drasticamente. Para atender a uma requisição de um usuário, o MMS gastou aproximadamente 0.12 segundos e, para comparar o perfil de cem usuários, o MMS gasta aproximadamente 1.09 segundos por requisição¹³.

O teste evidenciou que carregar no MMS coleções de níveis de interesses para cada usuário, com o Jena, é uma operação muito custosa. Desta forma, é interessante que os sistemas que manipulam dados usando o Jena utilizem algum sistema de *cache*. Contudo, o sistema de *cache* criado, apesar de não apresentar um desempenho ideal (possivelmente por ser simples), ainda

¹³ Estes tempos de resposta são válidos para casos onde o serviço não se encontra sobrecarregado.

assim mostrou-se útil e necessário para tornar os tempos de resposta do MMS aceitáveis.

4.4. Análise de Escalabilidade

Esta análise teve como objetivo identificar o limite máximo de usuários que a implementação do MMS, realizada nesta dissertação, seria capaz de suportar em um cenário de mobilidade de usuários. Os objetivos concretos do teste foram:

- Verificar se o sistema atenderia a todas as requisições dos usuários;
- Verificar o tempo de resposta para cada requisição;
- Medir o tempo entre as respostas da primeira e da última requisição, avaliando assim, o grau de assíncronia que isto causaria aos usuários.

Para a realização deste teste foi necessário criar programas auxiliares: um para criar os perfis dos usuários, de acordo com a ontologia proposta, e outro para realizar requisições simultâneas ao servidor MMS.

Para criar os perfis dos usuários, inicialmente, foram estipulados dez interesses, identificados com as letras de *A* até *J*. Usando uma função randômica, foi atribuída uma valoração aleatória para cada nível de interesse. Todos os níveis de interesses foram configurados como válidos em todas localizações, isto garantiu que sempre seriam calculados dez níveis de interesses entre os usuários. O identificador do dispositivo (endereço MAC) e o endereço de e-mail de cada usuário foram indexados pelo número do usuário que estava sendo criado. Por exemplo, para o primeiro usuário o seu endereço MAC foi o número *1* e seu e-mail *1@exemplo.com*¹⁴. Uma vez criados os dados, estes foram convertidos para o formato de ontologias OWL, o formato de entrada de perfis do MMS. Enviando a descrição OWL para o servidor MMS por meio de um cliente *Web Service*, foi possível criar uma base de conhecimento com cem usuários.

¹⁴ Com o objetivo de tornar o acompanhamento do teste mais fácil de ser feito, propositalmente criou-se endereços MAC não realistas. Normalmente um endereço MAC segue o formato de seis pares de dígitos em base hexadecimal, por exemplo, 00:74:4D:8A:88:01.

Para realizar as requisições criou-se um segundo programa cliente que inicia várias *threads*, cada uma representando um usuário do teste e efetuando uma requisição assíncrona para o servidor MMS.

Para realizar os testes, definiram-se duas áreas simbólicas, *A* e *B*. Configurou-se o Monitor/Sim para que simulasse que todos os cem usuários estivessem, inicialmente, na área *A*, e, posteriormente, migrassem todos para a área *B*. As requisições assíncronas ao MMS foram feitas ainda na área *A*, imediatamente antes da migração para a região simbólica *B*. Os testes foram repetidos dez vezes, para se obter a média dos tempos de notificação gerados pelo MMS e que são apresentados a seguir¹⁵.

4.4.1. Resultados

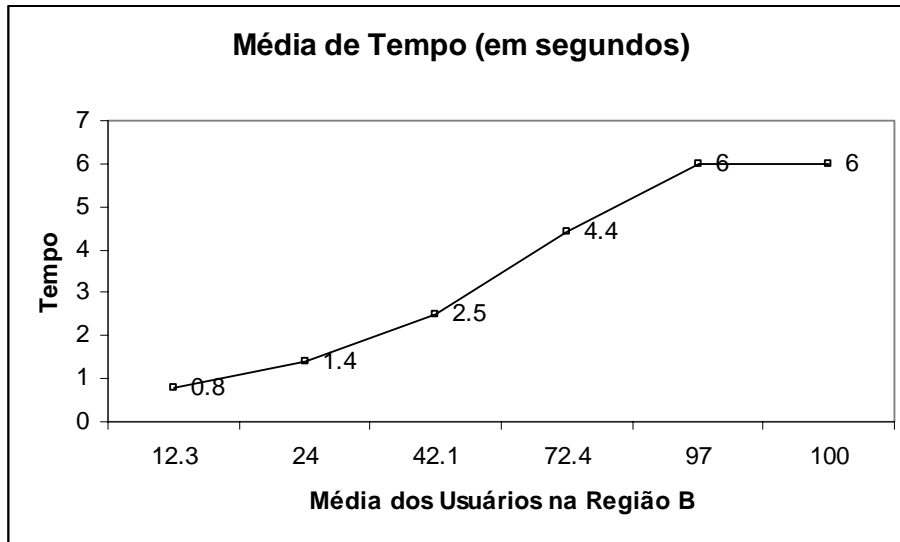
Os resultados obtidos com o teste de escalabilidade revelaram que o MMS foi capaz de atender a todos os usuários, ou seja, ele processou todas as requisições recebidas.

Quanto ao tempo de resposta para atender a cada usuário, à medida que mais usuários entravam na região simbólica, o tempo de resposta também aumentava. Isto se deve ao fato de que, na implementação atual do MMS, o tempo de resposta é diretamente proporcional a:

- Quantidade de usuários na região simbólica.
- Quantidade de níveis de interesses analisados.

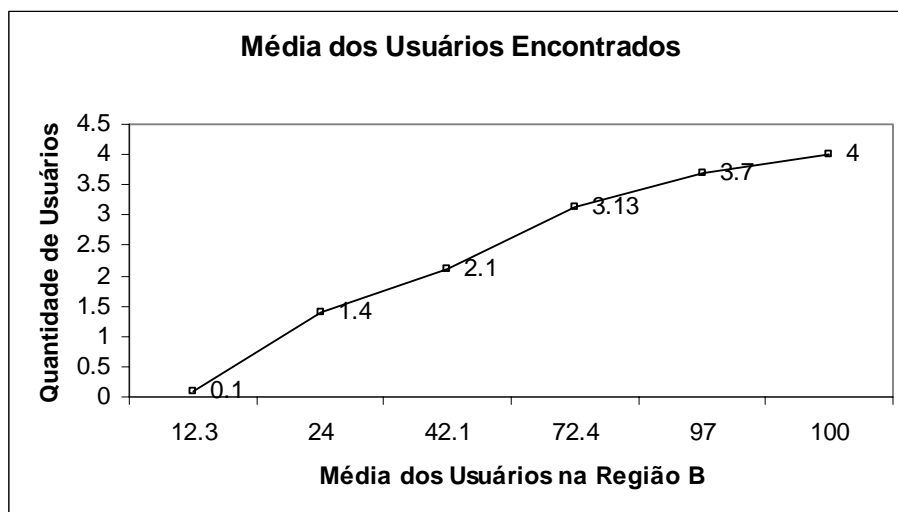
O Quadro 5 apresenta um gráfico que representa o desempenho do MMS em termos do tempo de resposta. A cada dez requisições, tratadas pelo MMS, foi feita uma média dos usuários que já haviam entrado na região *B* e dos tempos de resposta correspondentes. Desta forma, o gráfico demonstra, por exemplo, que durante as dez primeiras requisições, tratadas pelo MMS, existia uma média de 12.3 usuários em *B*, e que cada requisição levou em média 0.8 segundos para ser processada. Por volta da quinquagésima requisição, todos os cem usuários já haviam entrado na região *B*, o que estabilizou a média do tempo de resposta para cada requisição em aproximadamente 6 segundos.

¹⁵ Os testes foram executados em um Athlon 1.6 com 512M de memória randômica.



Quadro 5 - Média dos tempos de resposta com relação ao número de usuários em B

Como mencionado anteriormente, nestes testes os níveis de interesses foram distribuídos de forma aleatória, de forma que apenas um subconjunto dos usuários co-localizados tem interesses similares. A título de curiosidade, o gráfico do Quadro 6 apresenta a média dos usuários encontrados pelo MMS com correlação acima de 60% (valor alto de correlação). Por exemplo, nota-se que entre a décima primeira e a vigésima requisição existia, em média, 24 usuários na região simbólica *B*, mas, em média, foram encontrados cerca 1.4 usuários relacionados para cada requisição.



Quadro 6 - Média dos usuários encontrados em B

Os resultados gerais dos experimentos foram os seguintes:

- O MMS gastou, na média, aproximadamente 4.3 segundos para atender a cada usuário;
- Verificando a quantidade de usuários na região *B*, no momento do tratamento de cada requisição, chegou-se à conclusão que, em média, o MMS comparou cerca 72 usuários na região *B* durante cada teste;
- Verificando a quantidade de usuários encontrados para cada requisição, chegou-se à conclusão que, em média, o MMS encontrou 2.9 possíveis parceiros com correlação acima de 60%.
- O tempo que o MMS gastou para tratar a requisição do último (centésimo) usuário a entrar em *B* é, em média, 5.5 minutos.

Os resultados obtidos revelaram que, na atual versão, o MMS não é escalável em termos de número de usuários, visto que já com cem usuários co-localizados apresenta grandes tempos de resposta para as requisições de alguns usuários (da ordem de minutos). Sobretudo, tais tempos de resposta tornam a utilização do serviço inviável para cenários com alta taxa de movimentação dos usuários ou com regiões simbólicas muito pequenas e próximas. Nestes casos poderia acontecer que MMS notificasse os usuários tardiamente sobre a presença de possíveis parceiros em regiões já visitadas. Da mesma forma, cenários com muitas regiões simbólicas (de tamanho pequeno) implicariam uma alta taxa de trocas de regiões; sendo portanto inapropriadas para uso do MMS. No entanto, assumindo-se que o tempo médio de permanência de usuários em cada região é da ordem de dezenas de minutos, e que cada usuário tem uma quantidade menor ou igual a dez interesses por região, acredita-se então que o MMS possa ser utilizado.

Para cada par de usuários, o valor de um *matching* entre os seus perfis é calculado, enviado ao usuário que fez a requisição (ou pedido de notificação) ao MMS e, em seguida, é descartado. Acredita-se que, para que o MMS seja capaz de atender a cenários com um número maior de usuários e com maior mobilidade, faz-se necessário otimizar o cálculo do valor de *matching* entre perfis. Uma possibilidade seria manter armazenados (em *cache*) estes valores para o conjunto de usuários co-localizados e, somente re-calcular, quando algum usuário atualiza o seu perfil em uma região.

Outro detalhe importante, a complexidade do algoritmo *Pearson R* implementado no MMS é da ordem $O(n^2)$, ou seja, não foram realizados estudos a fim de otimizar a implementação deste cálculo de correlação. Além disso, outras funções de *matching* (além do *Pearson R*) poderiam ser testadas dentro do escopo deste problema. Estas novas funções poderiam fornecer resultados de *matching* mais significativos ou (devido a uma baixa complexidade) melhorar desempenho do serviço.

Sendo assim, acredita-se que existam várias frentes para otimização do MMS e de seu *matching* que poderão ser feitas em trabalhos futuros, a fim atender ao requisito de escalabilidade do serviço.