

Sistemas de localização dinâmica de serviços em ambientes de computação móvel

Leonardo M. A. Pinho¹, Alfredo Goldman¹

¹Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo, Brasil

Abstract. Um dos desafios atuais, com o aumento do número de serviços em uma rede, é a localização dinâmica dos mesmos. Várias tecnologias de localização de serviços foram desenvolvidas para simplificar o uso dos mesmos, permitindo que eles sejam encontrados, configurados e usados com um mínimo de esforço. Entretanto, não existe nenhum sistema que selecione o servidor mais conveniente ao cliente, como o mais próximo a este. Apresentamos uma solução para estender os sistemas existentes, integrando-os com sistemas de localização física de unidades móveis, a fim de localizar o serviço mais próximo ao cliente.

1 Introdução

Uma rede de computadores é um conjunto de subredes e computadores interligados. Ela oferece aos seus usuários um enorme potencial para compartilhamento de recursos, tais como documentos, programas, dados e principalmente serviços.

Definimos serviço como uma coleção de componentes distribuídos colaborando entre si, a fim de disponibilizar um recurso ao cliente, seja ele físico, como uma impressora, ou lógico, como um envio de e-mail.

Nesse ambiente, mecanismos para descoberta de serviços tornam-se extremamente importantes. Sem eles, os usuários utilizam uma fração muito pequena dos benefícios de uma rede, por falta de conhecimento dos seus recursos.

Outro fator importante para a utilização dos serviços de uma rede é a forma de acesso a estes. Idealmente, os usuários gostariam de acessar os recursos automaticamente, sem a necessidade de reconfigurar o seu sistema, como instalar *drivers* manualmente, ou procurar o endereço IP do servidor.

1.1 Ambiente Móvel

O cenário se torna mais problemático quando analisamos as redes sem fio e suas respectivas unidades móveis¹ (UMs). Nesse tipo de rede, os usuários mudam frequentemente e por isso não tem conhecimento dos serviços disponíveis e nem de como acessá-los.

Imagine uma pessoa que deseja imprimir um arquivo do seu dispositivo móvel (Palm, IPaq, etc) conectado a uma rede sem fio de um prédio comercial. Na maioria das vezes, o usuário precisa contactar o administrador da rede para descobrir o endereço da impressora mais próxima, suas características e a respectiva forma de acesso.

¹Chamaremos de *unidade móvel* um usuário acompanhado de um dispositivo conectado a uma rede sem fio no padrão IEEE 802.11b.

Seria interessante, se existisse um sistema em que o usuário simplesmente informasse o interesse pelo serviço de impressão, e o sistema automaticamente encontrasse a impressora mais próxima a ele de acordo com a sua localização, e ainda, reconfigurasse o seu dispositivo para que fosse possível a utilização da mesma.

1.2 Objetivos

Apresentaremos um sistema sensível à localização (*location-aware*) em que o usuário simplesmente informa o interesse por um determinado serviço da rede e a aplicação automaticamente encontra o servidor mais próximo a ele e reconfigura a unidade móvel para que esta acesse o serviço. Utilizaremos como exemplo o serviço de impressão, mas a solução se adequa a qualquer tipo de serviço.

Para desenvolver o sistema, foi necessário solucionar alguns problemas: a-) como prover um mecanismo de busca e acesso transparente aos serviços; b-) como obter a localização das unidades móveis em uma rede sem fio.

Atualmente existem no mercado várias tecnologias que visam resolver o problema a-), são os chamados sistemas de descoberta de serviços (SDS). Na seção 2 explicaremos em detalhes o funcionamento desses arcabouços e as respectivas modificações realizadas.

Na seção 3 explicamos a solução encontrada para o problema b-). Na seção 4, por sua vez, descrevemos o protótipo construído, que integra as soluções apresentadas. E finalmente, na última seção, apresentamos algumas conclusões.

2 SDS

As tecnologias para descoberta de serviços foram desenvolvidas para reduzir a complexidade da utilização dos serviços e simplificar o uso de dispositivos móveis em uma rede.

Para isso, essas tecnologias provêem uma localização dinâmica (*on the fly*) dos serviços disponíveis. Alguns sistemas também oferecem acesso transparente aos mesmos, atuando como uma “ponte” (*middleware*) entre o usuário e o provedor do serviço. Sistemas mais avançados fornecem ainda um “esqueleto” para a implementação de um serviço, servindo como um arcabouço na criação dos mesmos.

Infelizmente, os SDSs estudados não resolvem totalmente o problema abordado, pois nenhum sistema possui a funcionalidade de escolher o servidor mais próximo ao cliente. Não há nenhuma integração com sistemas de localização de unidades móveis.

Isso ocorre pois os mesmos, em sua maioria, foram desenvolvidos para redes estruturadas, onde os seus elementos são fixos. Assim, foi necessário adaptar os SDSs para o ambiente de computação móvel, em que a localização é muito importante, integrando-os com sistemas de localização de unidades móveis.

Apresentaremos a seguir uma solução que estende o *Service Location Protocol* (SLP) [Perkins and Guttman, 1999], desenvolvido pelo *IETF*, para se tornar um sistema sensível à localização. Na próxima seção descreveremos as modificações realizadas no Jini [Inc., 2001], tecnologia desenvolvida pela Sun Microsystems.

2.1 Service Location Protocol (SLP)

Service Location Protocol foi desenvolvido pelo *IETF* (*Internet Engineering Task Force*) para possibilitar que uma aplicação descubra automaticamente a localização de um serviço desejado, incluindo o endereço (ou domínio) e outras informações de acesso.

Sua arquitetura é composta por três agentes, que interagem entre si: *User Agent* (UA), o usuário do serviço, que procura pela localização de um serviço; *Service Agent* (SA), o provedor do serviço, responsável por oferecer um serviço à rede; e o *Directory Agent* (DA), o repositório de serviços, que atua como uma central de informação sobre a localização dos serviços existentes na rede.

A comunicação entre os agentes é feita utilizando pares de mensagens enviadas através do protocolo de comunicação TCP/IP ou UDP.

2.1.1 Modificações

Para solucionar o problema abordado, foi necessário realizar algumas modificações no SLP, para que o mesmo contemplasse a seleção do servidor mais próximo ao cliente de acordo com a sua localização.

Alteramos ligeiramente a especificação do SLP, incluindo um novo tipo de mensagem `SrvRqstWithLocation`, que contém os mesmos campos da mensagem `SrvRqst` e um campo adicional *User Location*, que informa a localização atual da unidade móvel. Portanto, no momento da requisição de um serviço, o DA terá conhecimento da localização do usuário e poderá encontrar o servidor mais próximo ao UA.

Implementamos um *service agent*, que localiza o DA da rede e publica o seu serviço, informando a sua localização, geralmente fixa, através do atributo *SrvLocation*.

Após a publicação da URL do serviço, é tarefa do SA iniciá-lo propriamente. Para isso, implementamos uma classe `PrintService`, que é responsável por disponibilizar o canal de comunicação final com a impressora.

O aplicativo é inicializado fornecendo uma porta e um nome para a impressora. Com esses dados, o sistema abre um canal de comunicação na porta especificada. A partir desse momento, o cliente pode enviar, via *socket*, o arquivo serializado a ser impresso, e o aplicativo reenvia o mesmo para a impressora.

Além disso, foi necessário implementar um DA que recebesse mensagens do tipo `SrvRqstWithLocation` e, ao recebê-la, realizar uma busca na sua lista de serviços considerando a localização informada pelo usuário.

2.2 Jini Network Technology

O *Jini* [Inc., 2001] foi desenvolvido pela Sun Microsystems Inc. com o objetivo de possibilitar a criação de um centro de serviços em uma rede. Através desse elemento, os usuários podem descobrir facilmente os serviços existentes na rede.

O Jini foi desenvolvido sobre a linguagem Java, utilizando sua tecnologia para objetos distribuídos *RMI* na comunicação entre os elementos do sistema.

O Jini é composto por três elementos: *Lookup Service* (LS), *Service Provider* (SP), *Client* (C). Os dois últimos elementos são os usuários finais do sistema. O *Service Provider* é o elemento que disponibiliza o serviço e o *Client* é o elemento que deseja utilizá-lo. O *Lookup Service* é um elemento intermediário entre as requisições dos clientes e dos provedores, geralmente chamado de localizador (gerenciador) de serviços.

2.2.1 Modificações

Assim como o SLP, o Jini não fornece um método para seleção de serviços depois de encontrado o conjunto de servidores que atendem a requisição. Para disponibilizar essa

nova funcionalidade foi necessário modificar alguns procedimentos do Jini.

Criamos uma nova classe `ServiceRegistrarWithLocation`, que estende a classe `ServiceRegistrar` (representa o LS no Jini), sobrecarregando o método `lookup` com um parâmetro adicional que representa a localização atual do cliente.

Além disso, implementamos o *service provider*, que localiza o LS da rede e publica o seu serviço juntamente com um atributo que representa a sua localização.

Após a publicação da URL do serviço, o SP inicia o serviço propriamente. De forma análoga ao sistema SLP, isto é feito através da instanciação da classe `PrintService` recebendo como parâmetro a porta e nome do serviço (definidos no `ServiceProviderProxy`). Depois disso, o servidor está pronto para receber requisições de algum cliente através do seu *proxy*.

Posteriormente, o cliente pode consultar o serviço registrado, instanciando a classe `ServiceRegistrarWithLocation`, que é responsável por encontrar o serviço mais próximo ao cliente, examinando o atributo de localização de cada servidor registrado.

O método `lookup` é chamado informando a localização atual do cliente. O método retorna um objeto serializado `ServiceProviderProxy`, que é usado pelo cliente para utilizar efetivamente o serviço.

3 Sistemas de localização de UMs

Para que o cliente informe a sua localização ao SDS é necessário algum mecanismo que calcule automaticamente a localização atual da unidade móvel.

Com o aumento do número de redes sem fio e de dispositivos móveis, surgiu a necessidade de desenvolver aplicativos que reagissem com a localização do usuário, comportando-se de forma diferente conforme o contexto em que a unidade móvel se encontra. Por exemplo, aplicativos que fornecem informações baseadas na localização atual da UM, como os restaurantes próximos, hospitais na vizinhança, etc.

Esse tipo de aplicação também pode ser utilizada em supermercados cujo carrinho contém uma tela e na medida que o cliente se locomove, o display do carrinho fornece informações específicas sobre os produtos do corredor atual. O sistema também pode ser instalado em outros estabelecimentos, como shoppings, convenções, entre outros.

Entretanto, para que seja possível o desenvolvimento de aplicações sensíveis à localização é necessário um sistema que consiga rastrear e fornecer a localização de uma unidade móvel. A partir da década de 90, surgiram sistemas com essa finalidade.

Para solucionar o problema proposto, gostaríamos que a tecnologia de localização não utilizasse nenhum *hardware* adicional acoplado a UM, apenas a rede sem fio para localizar a UM. Por esse motivo, as opções se restringiram ao Ekahau [Inc., 2002] ou o Radar [Pinho and Goldman, 2003]. O sistema foi desenvolvido utilizando o Ekahau.

3.1 Ekahau Positioning Engine

A principal razão para o sucesso desse sistema é o seu processo de localização, que é feito totalmente via *software*. Diferente de outros sistemas, ele não requer nenhum *hardware* adicional, apenas utiliza a infra-estrutura da rede sem fio padrão IEEE 802.11b, para efetuar a localização de um dispositivo conectado à este tipo de rede.

Sua heurística principal se baseia na diferença da intensidade de sinal (*Received Signal Strength Intensity* - RSSI) entre locais físicos diferentes. No processo de instalação do Ekahau é realizado um processo de calibragem do sistema em que é armazenada a intensidade do sinal em alguns pontos de amostra do ambiente coberto pelas ERBs. Baseado nos pontos de amostra armazenados, o Ekahau consegue comparar a intensidade de sinal da UM com os mesmos e inferir a localização atual do dispositivo.

A arquitetura do sistema é composta por outros quatro módulos: *Positioning Engine*, disponibiliza a localização e rastreamento de UMs aos clientes; *Manager*, analisa as áreas cobertas e gerencia áreas lógicas; *Calibrator*, responsável por gravar os dados da calibragem; *Device Service*, serviço executado nas UMs a serem rastreadas.

4 Protótipo

Para testar efetivamente o sistema implementamos um cliente para dispositivos móveis, integrando as soluções já apresentadas. A implementação do protótipo pode ser encontrada em <http://www.ime.usp.br/~lmap/prototipo/index.html>.

O protótipo foi implementado totalmente em Java e utilizando a arquitetura Waba [Pinho and Goldman, 2003], a fim de manter a compatibilidade entre os módulos e aumentar o número de dispositivos móveis compatíveis ao sistema. Utilizamos para testes um Palm m505 e um Ipaq contendo uma placa sem fio. Na versão para o Palm a localização é fornecida pelo usuário. No Ipaq, a localização é obtida através do Ekahau.

5 Conclusão

Os sistemas de descoberta de serviço serão muito importantes nos futuros ambientes de rede, como em redes *ad hoc* e redes sem fio, que demandam uma reorganização automática devido a grande mobilidade dos seus elementos.

Eles provêm os serviços da rede aos usuários móveis sem qualquer intervenção administrativa, permitindo um acesso rápido e transparente aos recursos.

Nesses tipos de ambientes, em que os usuários são móveis, é muito importante que as aplicações sejam construídas levando-se em consideração a localização, de forma que ela se adapte ao local em que a unidade móvel se encontra, comportando-se de maneira diferenciada de acordo com a localidade do usuário.

Este trabalho mostra como adaptar os SDSs ao ambiente de computação móvel, transformando-os em sistemas sensíveis à localização [Pinho and Goldman, 2003], que reagem de forma diferente conforme a localização da unidade móvel. Acreditamos que isso incentivará a criação de serviços *location-aware*, como o de impressão, ou o de um guia turístico virtual, entre outros.

Referências

- Inc., E. (2002). Ekahau Positioning Engine. Technical report.
- Inc., S. M. (2001). Jini Architectural Overview. Technical report.
- Perkins, C. and Guttman, E. (1999). Service Location Protocol, Version 2. *RFC 2608*.
- Pinho, L. M. A. and Goldman, A. (2003). Sistemas de Localização Dinâmica de Serviços em Ambientes de Computação Móvel. *WCSF2003 - V Workshop de Computação sem Fio e Computação Móvel*. URL: <<http://www.inatel.br/wcsf/>>.