

JXTA e a Revolução P2P - Tirando Proveito do Edge Computing

Bruno Ferreira de Souza e Fabiane Bizinella Nardon

A informática tem repetido um ciclo de centralização/descentralização. Estamos claramente passando pela fase de centralização com os ambientes web, J2EE e Web Services. Nessa fase, grandes servidores concentram a maior parte do tráfego na rede, e os equipamentos periféricos, das pontas, são meros coadjuvantes na troca e na distribuição de informações.

Olhando para o futuro, para onde vai o mercado mundial e os próximos anos da informática, estamos no início de uma onda de descentralização, que provavelmente irá se ampliar no decorrer da próxima década. Considerando somente o aumento de celulares inteligentes e PDAs, a quantidade de "computadores" começa a criar uma massa crítica fundamental para essa nova onda.

Ao invés de tentar prever quando essas coisas acontecerão (por mais que os sinais indiquem que a onda já começou), como desenvolvedores, podemos entender e nos preparar conhecendo as nossas alternativas, e a melhor forma de utilizá-las.

É interessante observar que devido ao longo tempo de maturação, existem diversas arquiteturas que já evoluíram, foram implementadas e discutidas, e que possuem implementações robustas e confiáveis, mas que são ainda desconhecidas ou pouco utilizadas. Além disso, por desconhecimento, é comum se considerar apenas uma única solução para um problema, quando muitas vezes uma combinação de soluções seria a mais vantajosa.

Já que estamos entrando de cabeça na onda de um mundo de aplicações mais distribuídas, precisamos entender o que esse mundo pode nos trazer. É preciso considerar um mundo no qual o poder de processamento é cada vez mais distribuído. Temos hoje computadores pessoais cada vez mais poderosos, aliados a equipamentos como telefones celulares e agendas eletrônicas e que dentro em breve estarão em contínuo contato com outros mais futuristas como televisores inteligentes, controles-remotos universais e capazes de processamento, carros e telefones de mesa com processadores embutidos, rádios, relógios, MP3 players, e eventualmente pequenos servidores domésticos capazes de interagir com nossos equipamentos de home-theater, de multimídia, de video games.

Dentro de uma empresa, temos a presença de sistemas multimídia de comunicação, equipamentos de colaboração remota, sensores de segurança e de presença, sistemas de auxílio a decisão em realtime, o bom e velho banco eletrônico. Isso é claro, sem falar nos servidores, provavelmente maiores, mas com certeza, mais distribuídos.

Junte a tudo isso a conexão de rede. Primeiro, conectando PCs a redes remotas. Depois disponibilizando aplicações para nossos celulares. Amanhã, conectando tudo o que temos a redes locais, e eventualmente a redes remotas de alta velocidade. Tudo isso, e muito mais, está disponível hoje, e estará em nossas casas dentro da próxima década.

Parece muito tempo? Esse é o tempo que temos para tentar resolver a principal questão que essa visão futurista nos traz: como construir software capaz de transformar o que parece um futuro distante, em realidade. Como garantir que seremos capazes de criar os sistemas que não apenas interligarão os grandes servidores, mas suportarão o crescente número de aparelhos e clientes, nos limites da rede, independentes e distribuídos.

Se nossas aplicações não forem criadas para esse mundo distribuído, deverão ser repensadas. E como o tempo de maturação das tecnologias distribuídas é normalmente longo, serão as arquiteturas que hoje parecem futuristas demais, pouco usadas e conhecidas, que construirão os sistemas de amanhã.

O grande desafio desse mundo distribuído é realmente utilizá-lo. Seria fácil demais simplesmente continuar a criar Web Services em grandes servidores J2EE, e centralizar cada vez mais. Mas será preciso aproveitar os novos recursos distribuídos, e centralizar não será uma opção...

Estamos caminhando para um mundo cada vez mais conectado, onde um número crescente de equipamentos, sistemas, aparelhos e pessoas precisarão se comunicar e trocar informações e comportamento. É esse mundo e mercado que queremos atingir. São as aplicações que farão essa visão uma realidade que precisamos construir.

Se não temos condição de ver muito no futuro, temos como saber o que existe hoje, que talvez nos leve até lá. As aplicações do futuro podem até aparecer do nada, mas serão construídas em cima das experiências inovadoras que temos hoje. Entre estas experiências estão tecnologias como Jini, Tuple-Spaces, JavaSpaces, Grid Computing e uma nova tecnologia criada especificamente para explorar o poder da computação peer-to-peer, o JXTA.

JXTA vem da palavra "juxtapose", ou colocar lado-a-lado, e é mais uma tecnologia pensada por Bill Joy - fundador da Sun Microsystems, um dos criadores do Unix, de Java e de Jini, entre outras tecnologias. JXTA é uma definição de um conjunto de protocolos criados para fornecer uma infra-estrutura para aplicações Peer-to-Peer (ou P2P).

O movimento de aplicações P2P começou a atingir a imprensa com o lançamento (e explosão) do software Napster, para a troca de músicas na internet. A idéia principal por trás do P2P é que todos os nós (peers) da rede podem conversar diretamente, de forma totalmente distribuída. Dessa forma, para disponibilizar informação, não é necessário um servidor central, como o mundo web exige.

P2P é um modelo de desenvolvimento de aplicações, e existem várias aplicações P2P em uso hoje, as mais populares sendo as aplicações de troca de mensagens (ou Instant Message - IM). Mas para desenvolver uma aplicação P2P, existem diversas questões e dificuldades que precisam ser resolvidas, entre elas: identificação dos peers, suporte a diferentes protocolos, capacidade de atravessar firewalls, suporte à localização de recursos remotos, comunicação. Por tudo isso que JXTA foi criado, para ser uma infra-estrutura para o desenvolvimento de sistemas P2P. Ao invés de você criar sua própria infra-estrutura, com seus próprios protocolos e soluções para esses problemas, você utiliza JXTA como plataforma básica para desenvolver sua aplicação.

Isso traz diversas vantagens, onde a principal é o acesso a um grande número de usuários. O número de usuários, como mostrou o Napster, é o grande problema e o grande benefício dos sistemas P2P. Em um ambiente P2P mais usuários significam mais conteúdo, que significa mais usuários, o que dá mais performance para o sistema, que traz mais usuários, e mais conteúdo e mais performance, e entramos em um ciclo virtuoso que faz o sistema P2P crescer. Por outro lado, como vimos no caso do Napster, menos conteúdo gera menos usuários, que faz diminuir a performance, que causa menos usuários, que traz menos conteúdo e menos performance e menos usuários, e esse ciclo vicioso acaba por matar o sistema.

Com uma plataforma como o JXTA, mesmo uma aplicação com poucos usuários se beneficia do fato de existir um grande número de usuários da infra-estrutura, e portanto, conseguimos com que o sistema como um todo se comporte muito melhor para todas as aplicações.

Nos próximos anos, estaremos acompanhando o processo de descentralização da rede e, dessa forma, o mundo como um todo irá tirar proveito de um maior número de sistemas em todas as áreas. Com esse processo de descentralização, não só os grandes servidores serão importantes, e não só empresas serão capazes de fornecer informações e serviços, mas a informação estará cada vez mais distribuída com o indivíduo e esse será cada vez mais capaz de disponibilizar serviços para o resto da rede.

Para os desenvolvedores e empresas, quem vai se sobressair nesse novo mundo serão aqueles capazes de criar aplicações inovadoras que tirem proveito do mundo distribuído. Utilizando tecnologias como Jini, JXTA e Tuple-spaces, e integrando com serviços disponíveis em servidores J2EE ou dentro de Grids. Estamos caminhando a passos largos para um mundo em rede, distribuído, e nesse mundo, ser capaz de interagir com as diversas tecnologias e arquiteturas é a resposta certa.