Explicando as Falácias (ou falsas verdades) da Computação Distribuída

(Quanto mais as coisas mudam, mais elas permanecem as mesmas)

Arnon Rotem-Gal-Oz

[Este artigo é baseado em uma série de posts que apareceram pela primeira vez no Portal do Dr. Dobb www.ddj.com/dept/architect]

A indústria de software tem desenvolvido sistemas distribuídos há várias décadas. Dois exemplos incluem a ARPANET (que eventualmente evoluiu para a Internet), criado em 1969 pelo Departamento de Defesa dos EUA, e o protocolo SWIFT (usado para transferências de dinheiro), estabelecida no mesmo período de tempo [Britton 2001].

No entanto, Peter Deutsch, um companheiro da SUN, escreveu em 1994 sobre 7 pressupostos que arquitetos e designers de sistemas distribuídos podem adotar, que a longo prazo se mostraram errados, resultando em todos os tipos de problemas e penúrias para aqueles que resolverão os problemas decorrentes de tais suposições. Em 1997, James Gosling acrescentou outra falácia às apontadas por Peter Deutsch [JDJ2004]. Os pressupostos são agora conhecidos como "As 8 falácias da computação distribuída" [Gosling]:

1. A rede é de confiança.
2. A latência é zero.
3. Largura de banda é infinita.
4. A rede é segura.
5. A topologia não muda.
6. Há somente um administrador.
7. Custo de transporte é zero.
8. A rede é homogêneo.

Este texto descreve, explica e verifica a relevância de cada uma dessas falácias nos sistemas distribuídos nos dias de hoje.

1 - A rede é confiável

A primeira falácia é "*A rede é confiável*." Por que isso é uma falácia? Bem, quando foi a última vez que você viu um switch falhar? Afinal, até mesmo os switches atuais mais básicos têm um MTBFs (tempo médio entre falhas) a cada 50.000 horas de funcionamento (ou mais).

Por um lado, se a sua aplicação é do tipo crítico, rodando 365 dias no ano, você poderá até superar esse índice e a lei de Murphy vai se certificar de que isso aconteça no momento mais inadequado. No entanto, a maioria dos aplicativos não são assim. Então, qual é o problema?

Bem, há uma abundância de problemas: falhas de energia, alguém tropeça no cabo de rede, de repente clientes wireless se conectam, e assim por diante. Se o hardware não é suficiente, o software também pode falhar, e irá falhar em algum momento.

A situação é mais complicada se houver colaboração de um parceiro externo, como um aplicativo de *e-commerce* trabalhando com um serviço de processamento de cartão de crédito externo. O *lado deles* da conexão não está sob controle direto do *nosso lado* da conexão. Por último, existem ameaças de segurança, como ataques de DDOS e afins.

O que isso significa para seu projeto?

No lado da infra-estrutura, é preciso pensar sobre a redundância de hardware e de software para pesar entre os riscos de falha e o investimento necessário. No lado do software, é preciso pensar sobre mensagens/chamadas se perdendo sempre que uma mensagem for enviada ou uma chamada no sistema fixo for efetuada. Por outro lado, pode-se usar um meio de comunicação que forneça um pacote de mensagens completo, confiável, como WebSphereMQ ou MSMQ, por exemplo. Se não houver disponibilidade, faça novas tentativas; confirme mensagens importantes, identifique/ignore mensagens duplicatas (ou use mensagens idempotentes), reordene as mensagens (ou não dependa da ordem das mensagens), verifique a integridade da mensagem, e assim por diante.

Para resumir, a rede não é confiável e nós, como arquitetos/designers de software, precisamos corrigir isso.

2 - A latência é zero

Segunda falácia da Computação Distribuída: supor que "A latência é Zero".

Latência é quanto tempo leva para os dados se deslocarem de um lugar para outro (oposto da largura de banda, que é a quantidade de dados que pode ser transferida durante um determinado tempo). A latência pode ser relativamente boa em uma LAN, mas se deteriora rapidamente quando estamos analisando cenários em WAN ou internet.

A latência é mais problemática do que a largura de banda. Aqui está uma citação de um post por Ingo Rammer sobre latência vs. Largura de banda [Ingo] que ilustra isso:

> *"Mas eu considero realmente interessante observar que a largura de banda ponta-a-ponta aumentou em 1.468 vezes nos últimos 11 anos, enquanto a latência (o tempo de uma única toma de ping) só foi melhorado dez vezes Se isso ainda não é suficiente, há ainda um limite natural da latência. O mínimo round -.. tempo de viagem entre dois pontos da terra - é determinado pela velocidade máxima de transmissão de informação: a velocidade da luz. A uma velocidade próxima dos 300.000 quilômetros por segundo, sempre serão necessários pelo menos 30 milissegundos para um ping percorrer Europa - EUA - Europa, mesmo que o processamento seja feito em tempo real. "* É uma questão física que ainda não foi superada.

Pode-se pensar que está tudo bem se os aplicativos forem implantados somente em LANs. No entanto, mesmo quando se trabalha em uma rede local com Ethernet Gigabit, deve-se ter em mente que a latência é muito maior do que o tempo de acesso à memória local. Assumir que a latência é zero pode levar à conclusão de que fazer uma chamada no sistema fixo é quase como fazer um chamada local. Este é um dos problemas com abordagens do tipo 'objetos distribuídos fornecem "transparência da rede'" induzindo o mantenedor a realizar diversas chamadas a objetos que são realmente remotos e dispendiosos (relativamente).

Levar a latência em consideração significa que você deve se esforçar para fazer o menor número possível de chamadas, supondo que você tem largura de banda suficiente para passar o máximo de dados em cada chamada. Existe um bom exemplo que ilustra o problema de latência e o que foi feito para resolvê-lo no Windows Explorer em

http://blogs.msdn.com/oldnewthing/archive/2006/04/07/570801.aspx

Outro exemplo é AJAX. A abordagem do AJAX permite utilizar o tempo ocioso no sistema que os usuários gastam interpretando os dados recebidos para identificar quais serão os novos dados - no entanto, deve-se ainda considerar a latência. Vamos dizer que uma nova interface AJAX está sendo testada e tudo parece muito bem no ambiente de teste. Mesmo que o aplicativo passe no testes com louvor, ainda pode falhar seriamente no ambiente de produção se a latência não for verificada. A recuperação de dados em segundo plano é bom, mas se isso não acontecer rapidamente, a aplicação

poderá falhar e travar... Outras ferramentas como Shunra Virtual Enterprise, Opnet Modeler e muitos outros podem ser utilizadas para simular as condições da rede e entender o comportamento do sistema, evitando o fracasso na versão de produção.

3 - Largura de banda é infinita

A próxima falácia da Computação Distribuída é "largura de banda é infinita".

Na opinião do autor desse artigo, essa falácia não é tão importante quanto as outras. Se há uma coisa que está constantemente ficando melhor em relação às redes é largura de banda. No entanto, existem dois fortes motivos para manter essa suposição uma falácia.

Uma delas é que, enquanto a largura de banda cresce, o mesmo acontece com a quantidade de informação que tentamos forçar através dela. VoIP, vídeos e IPTV são algumas das aplicações mais recentes que ocupam largura de banda adicional. Downloads, UIs com mais conteúdo, e aplicativos dependentes de formato (ex. XML) estão também no local de trabalho, especialmente se você estiver usando T1 ou linhas de rede inferiores. No entanto, mesmo quando você acha que uma linha Ethernet de 10 Gbit seja mais do que suficiente, o sistema pode ser solicitado com algumas dezenas de Terabytes de dados novos por dia (informação baseada em um sistema real).

A outra força em ação para menor largura de banda é a perda de pacotes (juntamente com tamanho de quadro). Esta citação ressalta este ponto muito bem: "No ambiente de rede de área ou campus local, RTT e perda de pacotes são ambos geralmente pequenas o suficiente para que fatores além dos previstos na equação acima definam o seu limite de desempenho (por exemplo, largura de banda em links fracamente disponíveis, velocidades de encaminhamento de pacotes , as limitações da CPU do host, etc.). No entanto WAN, RTT e perda de pacotes são muitas vezes bastante grande e algo que os sistemas finais não pode controlar. Assim, sua única esperança para melhorar o desempenho na área ampla é a utilização de tamanhos de pacotes maiores. Vamos dar um exemplo:.. Nova York para Los Angeles Round Trip Time (RTT) é cerca de 40 ms, e vamos dizer que a perda de pacotes é de 0,1% (0.001) Com uma MTU de 1500 bytes (MSS de 1460), o débito do TCP terá um limite superior de cerca de 6,5 Mbps! E não, isso não é uma janela de limitação de tamanho, mas sim uma baseada na capacidade do TCP para detectar e recuperar de congestionamento (perda). Com 9000 bytes frames, o débito do TCP poderia chegar a cerca de 40 Mbps. Ou vamos olhar para esse exemplo em termos de taxas de perda de pacotes. Mesmo tempo ida e volta, mas vamos dizer que queremos alcançar uma taxa de transferência de 500 Mbps (metade de um "gigabit"). Para fazer isso com 9000 bytes frames, precisaríamos de uma taxa de perda de pacotes não superior a 1x10 ^ (- 5). Com 1500 bytes frames, a taxa de perda de pacotes necessários é baixo para 2.8x10 ^ -7! Enquanto o quadro jumbo é de apenas 6 vezes maior, que nos permite o mesmo rendimento em face de 36 vezes mais perda de pacotes. "[WareOnEarth] Reconhecendo a largura de banda não é infinita tem um efeito de equilíbrio sobre as implicações da" falácia Latência é zero ", isto é, se agir sobre a realização da latência não é zero nós modelamos algumas mensagens grandes limitações de largura de banda nos orientar para. esforçar-se para limitar o tamanho das informações que enviamos ao longo do fio. A principal implicação é, então, a considerar que, no ambiente de produção da nossa aplicação pode haver problemas de largura de banda que estão além do nosso controle. E devemos ter em mente o quanto é dados Espera-se que t ravel sobre o sábio A recomendação que fiz no meu post anterior -. para tentar simular o ambiente de produção -. é válido aqui também

4 - A Rede é Segura

Peter Deutsch introduziu as falácias de computação distribuída em 1991. Pode-se pensar que nos últimos 15 anos a falácia "a rede é segura" já não exista mais.

Infelizmente, esse não é o caso. Mas não é porque a rede agora é segura, ninguém seria ingênuo o suficiente para assumir que é. As estatísticas publicadas pelo Aladdin.com [Aladdin] mostram que:

"**Para 52% das redes, o perímetro é a única defesa!**

De acordo com Preventsys e Qualys, 52% dos diretores de segurança da informação principais reconhecem ter uma abordagem do tipo "Moat & Castle" no seu sistema de segurança global da rede. Eles admitiram que, uma vez que o perímetro de segurança é penetrada, suas redes estão em risco. No entanto, 48% consideram-se "pró-ativos" quando se trata de segurança da rede e sentem que eles têm uma boa compreensão sobre a postura de segurança de sua empresa. 24% sentiram que sua segurança era semelhante ao Fort Knox (que seria necessário um pequeno exército para atravessar), enquanto que 10% comparam a segurança da sua rede a um queijo suíço (falhas de segurança dentro e fora). Os restantes 14% dos entrevistados descreveram a sua segurança de rede atual como sendo bloqueado no interior, mas ainda não completamente garantida para o exterior.

A Preventsys e Qualys também descobriu que 46% dos agentes de segurança gastam mais de um terço do seu dia, e em alguns casos, mais de sete horas, analisando relatórios gerados a partir de suas várias soluções pontuais de segurança. "A não ser que você tenhas acabado de desembarcar de um outro planeta, a rede está longe de ser garantida". Aqui estão algumas estatísticas que ilustram esta situação.

Através do monitoramento 24x7 contínuo de centenas de empresas do grupo Fortune 1000, a RipTech descobriu várias tendências extremamente relevantes em segurança da informação. Entre. Elas mestgão:

1 Tendências de ataques gerais na Internet estão mostrando uma taxa anual de crescimento de 64%

2 A empresa média sofre 32 ataques por semana a cada 6 meses (fator crescente)

3 ataques durante a semana aumentou nos últimos 6 meses "[RipTech].

Quando se tenta encontrar algumas estatísticas de incidentes atualizados, encontrou-se a seguinte nota [CERT]: Dado o amplo uso de ferramentas automatizadas de ataque, ataques contra Internet - sistemas conectados - tornaram-se tão comuns que a contagem do número de incidentes relatados fornece pouca informação no que diz respeito à apreciação do âmbito e impacto dos ataques. Portanto, a partir de 2004, não serão mais publicados os números de incidentes relatados. Em vez disso, as empresas irão trabalhar com outras pessoas da comunidade para desenvolver e apresentar um relatório sobre métricas mais significativas." (o número de incidentes de 2003 foi de 137.539 incidentes ...) Por fim, a Aladdin afirma que os custos de Malware para 2004 (vírus, worms, cavalos de Tróia etc.) são estimados entre 169 e 204 bilhões de dólares [Aladdin]. As implicações de segurança das redes (in- e extra-net) são óbvias: é preciso construir as soluções de segurança desde o 1º dia.

É constantemente mencionado que a segurança é um sistema de atributo de qualidade que precisa ser levado em consideração a partir do nível arquitetônico. Há dezenas de livros que falam sobre segurança e não se pode começar a mergulhar em todos os detalhes em um artigo pequenmo como esse. Em essência, é preciso executar a modelagem de ameaças para avaliar os riscos de segurança. Em seguida, na sequência de novas análises, decidir quais medidas devem ser tomadas para que os riscos sejam reduzidos (um tradeoff entre custos, riscos e sua probabilidade).

A segurança é geralmente uma solução multi-camadas que é tratada na rede, infra-estrutura e níveis de aplicação. Arquiteto de sistemas podem não ser especialistas em segurança, mas precisam estar cientes da necessidade da segurança e das suas implicações (por exemplo: bloqueio de serviço multicast, contas de usuário com privilégios limitados - sem acesso a alguns recursos de rede etc.) .