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,a 365x7, você poderá até superar esse índice - e a lei de Murphy vai se certificar de que isso acontece 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 pode falhar também, e ele irá.

A situação é mais complicada se você colaborar com 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 seu controle direto. Por último, existem ameaças de segurança, como ataques de DDOS e afins.

O que isso significa para seu projeto?

No lado da infra-estrutura, você precisa pensar sobre a redundância de hardware e software e pesar os riscos de falha versus o investimento necessário. No lado do software, você precisa pensar sobre mensagens/chamadas se perdendo sempre que você enviar uma mensagem ou fizer uma chamada com fio. Por outro lado, você pode usar um meio de comunicação que fornece um pacote de mensagens completo, confiável, como WebSphereMQ ou MSMQ, por exemplo. Se você não pode usar um, prepare-se para tentar novamente, confirmar mensagens importantes, identificar / ignorar duplicatas (ou use idempotent mensagens), reordenar as mensagens (ou não depender de ordem de mensagem), verificar a integridade da mensagem, e assim por diante. Uma nota sobre WS-Reliable Messaging: A especificação suporta vários níveis de garantia mensagem - mais uma vez, pelo menos uma vez, uma única vez e encomendas. Você deve lembrar que embora ele só cuida de entregar a mensagem, enquanto os nós da rede estão funcionando, ele doesnâ € ™ t alça persistência e você ainda precisa cuidar disso (ou use uma solução de fornecedor que faz isso para você) para uma solução completa. Para resumir, a rede não seja confiável e nós, como arquiteto de software / designers, precisa resolver isso.

2 - A latência é zero

A segunda falácia da Computação Distribuída é a suposição de que "*A latência é Zero*". Latência é quanto tempo que leva para os dados para se deslocar de um lugar para outro (versus largura de banda que é a quantidade de dados que pode transferir durante esse tempo). Latência pode ser relativamente bom em uma LAN - mas a latência se deteriora rapidamente quando você se move para WAN cenários ou cenários de internet. A latência é mais problemática do que a largura de banda. Aqui está uma citação de um post por Ingo Rammer na latência vs. Bandwidth [Ingo] que ilustra isso: "Mas eu acho que itâ € ™ s realmente interessante ver que a largura de banda end-to-end 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 este não for suficiente, há ainda uma tampa natural da latência O mínimo round -.. viagem tempo entre dois pontos da terra é determinado pela velocidade máxima de transmissão de informação:. a velocidade da luz Em cerca de 300.000 quilômetros por segundo (3,6 * 10E12 teraangstrom por quinzena), ele sempre vai demorar pelo menos 30 milissegundos para enviar um ping da Europa para os EUA e para trás, mesmo se o processamento seria feito em tempo real. " Você pode pensar que está tudo bem se você só implantar seu aplicativo em LANs. No entanto, mesmo quando você trabalha em uma rede local com Gigabit Ethernet você ainda deve ter em mente que a latência é muito maior, em seguida, acessar a memória local Assumindo que a latência é zero pode ser facilmente tentados a assumir fazer uma chamada sobre o fio é quase como fazer um chamada local - este é um dos problemas com abordagens como objetos distribuídos, que fornecem "a transparência da rede" - sedutor que você faça um monte de chamadas de ajuste fino para objetos que são realmente remoto e caro (relativamente) para chamar a. Levando em consideração a latência significa que você deve se esforçar para fazer o menor número possível de chamadas e supondo que você tem largura de banda suficiente (que vai falar sobre a próxima vez) que você gostaria de passar o máximo de dados em cada um esta chama. Não é um bom exemplo que ilustra o problema de latência eo 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 AJAX permite utilizar o tempo morto os usuários gastam digerindo dados para recuperar mais dados - no entanto, você ainda precisa considerar a latência. Vamos dizer que você está trabalhando em uma nova frente brilhante AJAX - final - tudo parece muito bem em seu ambiente de teste. Ele também brilha em seu ambiente de preparação passar os testes de carga com cores de vôo. O aplicativo ainda pode falhar miseravelmente no ambiente de produção se você deixar de testar para problemas de latência - recuperação de dados em segundo plano é bom, mas se você não pode fazer isso rápido o suficiente a aplicação ainda cambalear e ficará sem resposta .... (Você pode ler mais sobre AJAX e latência aqui.) [RichUI] Você pode (e deve) usar ferramentas como Shunra Virtual

Enterprise, Opnet Modeler e muitos outros para simular as condições da rede e entender o comportamento do sistema, assim, evitar o fracasso no sistema 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 forte quanto as outras. Se há uma coisa que está constantemente ficando melhor em relação às redes é largura de banda. No entanto, existem duas causa no trabalho 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 dependência de formatos (ex. XML) estão também no local de trabalho, especialmente se você estiver usando linhas de rede T1 ou inferiores. No entanto, mesmo quando você acha que uma linha Ethernet de 10 Gbit seja mais do que suficiente, o sistema pode ser atingido com mais de 3 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, que 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 f outros do que a equação acima atores definir o seu limite de desempenho (por exemplo, larguras de banda de link disponível crus, 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 do perímetro é a única defesa!**

De acordo com Preventsys e Qualys, 52% dos diretores de segurança da informação principais reconheceu ter um "Moat & Castle" abordagem ao seu 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-ativa" 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 a Fort Knox (que seria necessário um pequeno exército para atravessar), enquanto que 10% em relação a segurança da rede de queijo suíço (falhas de segurança dentro e por fora). Os restantes 14% dos entrevistados descreveram a sua segurança de rede atual como sendo bloqueado no interior, mas ainda não completamente garantido para o exterior. Preventsys e Qualys também descobriu que 46% dos agentes de segurança gastar mais de um terço do seu dia, e em alguns casos, tanto quanto 7 horas, analisando relatórios gerados a partir de suas várias soluções pontuais de segurança. "No caso de você acaba de desembarcar de um outro planeta, a rede está longe de ser garantido Aqui estão algumas estatísticas que ilustram esta:. Através do monitoramento 24x7 contínuo de centenas de empresas Fortune 1000, RipTech descobriu várias tendências extremamente relevantes em segurança da informação Dentre. eles: 1 Geral Internet tendências de ataques estão mostrando uma taxa anual de crescimento de 64% 2 A empresa média experimentada 32 ataques por semana ao longo dos últimos 6 meses 3 ataques durante a semana aumentou nos últimos 6 meses "[RipTech]. Quando eu tentei encontrar algumas estatísticas de incidentes atualizados, eu vim com o seguinte [CERT]: Nota: 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 fornecer pouca informação no que respeita à apreciação do âmbito e impacto dos ataques. Portanto, a partir de 2004, deixará de publicar o número de incidentes relatados. Em vez disso, vamos 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 Aladdin afirma que os custos de Malware para 2004 (vírus, worms, cavalos de Tróia etc.) são estimados entre 169 bilião dólares e 204.000 milhões dólares [Aladdin] As implicações da rede (in) segurança são óbvias -. você precisa para construir a segurança em suas soluções do dia 1. Eu mencionei em um post anterior que a segurança é um sistema de atributo de qualidade que precisa ser levado em consideração a partir do nível arquitectónico. Há dezenas de livros que falam sobre segurança e eu não posso começar a mergulhar em todos os detalhes em um curto post. Em essência, você precisa executar a modelagem de ameaças à avaliar os riscos de segurança. Em seguida, na sequência de novas análises decidir quais riscos são deve ser mitigado por que medidas (um tradeoff entre custos, riscos e sua probabilidade). A segurança é geralmente uma solução multi-camadas que é tratada na rede, infra-estrutura e aplicação níveis. Como arquiteto você pode não ser um especialista em segurança - mas você ainda precisa estar ciente de que é necessário de segurança e as implicações que pode ter (por exemplo, você pode não ser capaz de usar multicast, contas de usuário com privilégios limitados podem não ser capaz de acessar alguns recursos de rede etc.)

5 - A topologia não muda

A quinta Falácia da Computação Distribuída é sobre a topologia da rede. "Topologia não muda." É isso mesmo, ela não muda desde que permaneça no laboratório de teste. Quando você implanta um aplicativo em estado selvagem (isto é, a uma organização), a topologia da rede é geralmente fora do seu controle. A equipe de operações (IT) é provável que adicionar e remover servidores de vez em quando e / ou fazer outras alterações para a rede ("este é o novo Active Directory, vamos utilizar para SSO, estamos substituindo RIP com OSPF e este servidores do aplicativo estão se movendo para a área 51 "e assim por diante). Por último, existem falhas de servidor e rede que podem causar alterações de

roteamento. Quando você está falando sobre os clientes, a situação é ainda pior. Há laptops indo e vindo, redes ad-hoc sem fio, novos dispositivos móveis. Em suma, a topologia está mudando constantemente. O que isso significa para as aplicações que escrevemos? Simples. Tente não depender de endpoints ou rotas específicas, se você não pode estar preparado para renegociar endpoints. Outra implicação é que você iria querer ou fornecer transparência de localização (por exemplo, usando um ESB, multicast) ou fornecer serviços de pesquisa (por exemplo, um diretório / JNDI / LDAP Ativo). Outra estratégia consiste em resumo a estrutura física da rede. O exemplo mais óbvio para isso é DNS nomes em vez de endereços IP. Recentemente me mudei minha (outros) blog a partir de um serviço de hospedagem para outro. A transferência correu sem contratempos como eu tive ambos os sites em funcionamento. Então, quando as tabelas de roteamento DNS foram atualizados (leva um ou dois dias para a mudança de ondulação) leitores só veio para o novo site, sem saber o encaminhamento (topologia) mudou sob seus pés. Um exemplo interessante é que se deslocam de WS-Routing para WS-Addressing. No WS-Routing uma mensagem pode descrever o seu próprio caminho de roteamento - isso pressupõe que uma mensagem pode saber o caminho que necessita para viajar com antecedência. A topologia não muda (isto também provoca uma vulnerabilidade de segurança - mas isso é outra história), onde os WS-Addressing mais recentes depende de roteamento "Next Hop" (o caminho TCP / IP funciona), que é mais robusto. Outro exemplo é o roteamento no SQL Server Service Broker. A parte problemática é que as rotas precisam ser definidas dentro corretor de serviço. Isso é problemático, uma vez que agora tem que se lembrar de ir para o Service Broker e atualizar a tabela de roteamento é quando mudanças na topologia. No entanto, para atenuar este problema, o roteamento baseia-se na semântica de próxima hop e permite especificar o endereço pelo nome DNS.

Há somente um administrador

A sexta Falácia da Computação Distribuída é "Há somente um administrador".

É possível fugir dessa falácia se você instalar seu software em pequenas redes locais isolados (por exemplo, uma única pessoa de grupo TI sem WAN / Internet). No entanto, para a maioria dos sistemas corporativos a realidade é muito diferente. O grupo de TI geralmente tem diferentes administradores, atribuídos de acordo com a experiência - bancos de dados, servidores web, redes, Linux, Windows, estrutura principal e similares.

Esta é a situação simples.

O problema ocorre quando a empresa colabora com entidades externas (por exemplo, conectar-se com um parceiro de negócios), ou se seu aplicativo é implantado para o e-commerce e é hospedado por algum serviço de hospedagem e a aplicação consome serviços externos (pense Mashups). Nessas situações, os outros administradores não estão ainda sob seu controle, e eles podem ter suas próprias agendas/regras. Neste ponto você pode dizer "Ok, há mais de um administrador. Mas por que eu deveria me importar?"

Bem, contanto que tudo funcione, talvez você não se importe. Você se importaria, no entanto, quando as coisas se extraviassem e houvesse a necessidade de identificar o problema (e resolvê-lo).

Por exemplo, recentemente tive um problema com uma aplicação ASP.NET que exigia plena confiança em um serviço de hospedagem, que só permitia confiança média. O pedido teve de ser reformulado (desde a mudança de serviço de acolhimento não era uma opção), a fim de trabalhar. Além disso, você precisa entender que os administradores provavelmente não vai ser parte de sua equipe de desenvolvimento por isso precisamos fornecer-lhes ferramentas para diagnosticar e encontrar problemas. Isto é essencial quando a aplicação envolve mais de uma empresa ("Será que é o seu problema ou o nosso?"). Uma abordagem proativa é incluir também ferramentas para a monitorização em curso operações de bem; por exemplo, para permitir que os administradores a identificar os

problemas quando eles são pequenos - antes que se tornem uma falha do sistema. Outra razão para pensar sobre vários administradores é evoluções. Como você vai lidar com eles? Como você está indo para certificar-se que as diferentes partes do nosso aplicativo são sincronizados e pode realmente trabalhar em conjunto (distribuído, lembra?); por exemplo, se o esquema DB atual coincidir com o modelo de mapeamento O / R e objeto atual? Mais uma vez este problema agrava quando terceiros estão envolvidos. O seu parceiro pode continuar a interoperar com o nosso sistema quando fizemos alterações ao contrato público (em uma SOA) para que, por exemplo, você precisa pensar sobre a compatibilidade com versões anteriores (ou compatibilidade talvez até mesmo para a frente) ao conceber contratos de interoperabilidade. Em suma, quando há mais de um administrador (a menos que nós estamos falando sobre um sistema simples e até mesmo que pode evoluir mais tarde se for bem sucedido), é preciso lembrar que os administradores podem restringir suas opções (administradores que estabelecem cotas de disco, limitada privilégios, portas e protocolos limitados e assim por diante), e que você precisa para ajudá-los a gerenciar suas aplicações.

Custo de transporte é zero

Falácia número 7: "custo de transporte é zero". Há duas maneiras de se interpretar essa afirmação, sendo que ambos são falsas suposições. Uma indica que não há custo entre a camada de aplicação para a camada de transporte. Isso é uma falácia, uma vez que o empacotamento dos dados (paralelo para serial) para envio utiliza recursos de ambos os computadores e adiciona à latência. Interpretando a declaração desta forma enfatiza a falácia "A latência é Zero", lembrando-nos de que existem custos adicionais (tanto em tempo e recursos). A segunda maneira de interpretar a declaração é que os custos (como em dinheiro de caixa) para a criação eo funcionamento da rede são gratuitas. Isto também está longe de ser verdade. Existem custos - os custos para comprar os roteadores, os custos para proteção da rede, os custos com a locação financeira a largura de banda para acesso à internet, e os custos de operação e manutenção da rede em execução. Alguém, em algum lugar terá que escolher a guia e pagar esses custos. Imagine que você criou com êxito Google de Dilbert - motor de busca assassino [Adams] (talvez usando a mais recente Web 2.0 sinos-e assobia-na IU), mas você irá falhar se você deixar de ter em conta os custos que são necessários para manter seu serviço até , correndo, e responsivo (E3 Lines, datacenters com interruptores, SANs etc.). O takeaway é que, mesmo em situações você acha que os outros falácias não são relevantes para a sua situação, porque você confiam nas soluções existentes ("sim, nós vamos implantar protocolo HSRP da Cisco e se livrar do problema de confiabilidade de rede"), você ainda pode ser delimitada pelos custos da solução e que você precisa para resolver seus problemas usando mais custo - soluções eficazes.

A rede é homogênea.

A falácia Computação Distribuída oitavo e último é "A rede é homogêneo." Enquanto os primeiros sete falácias foram cunhado por Peter Deutsch, eu li [JDJ2004] que o oitavo falácia foi adicionado por James Gosling seis anos mais tarde (em 1997). A maioria dos arquitetos de hoje não são ingênuos o suficiente para assumir essa falácia. Qualquer rede, exceto talvez os muito triviais, não são homogêneos. Heck, mesmo a minha rede doméstica tem um HTPC baseado em Linux, um par de PCs baseados em Windows, uma (pequena) NAS, e um dispositivo WindowMobile 2005 - - todos ligados por uma rede sem fio. O que é verdade em uma rede doméstica é quase uma certeza em redes corporativas. Eu acredito que uma rede homogênea, hoje, é a exceção, não a regra. Mesmo se você conseguiu manter sua rede interna homogênea, você vai bater esse problema quando você iria tentar cooperar com um parceiro ou fornecedor. Assumindo que esta falácia não deve causar muito problema ao nível da rede menor como IP é praticamente onipresente (por exemplo, até mesmo um ônibus especializado como Infiniband tem uma implementação I P-Over-IB, embora possa resultar em sub-aproveitamento de recursos IP inativos. Vale a pena prestar atenção ao fato de a rede não é homogêneo no nível do aplicativo. A implicação disso é que você tem que assumir a interoperabilidade serão

necessários mais cedo ou mais tarde e estar pronto para apoiá-lo desde o primeiro dia (ou ., pelo menos, projeto onde você gostaria de adicioná-lo mais tarde) Não confie em protocolos proprietários - que seria mais difícil integrá-los mais tarde, fazer uso de tecnologias padrão que são amplamente aceitos; os exemplos mais notáveis sendo XML Web Services ou por o.. Assim, grande parte da popularidade de XML e Web Services pode ser atribuído ao fato de que ambas as tecnologias ajudar a aliviar os efeitos da heterogeneidade do ambiente corporativo. Em suma, a maioria dos arquitetos / designers de hoje estão conscientes desta falácia, que é por que tecnologias interoperáveis são populares. Ainda assim, é algo que você precisa ter em mente, especialmente se você estiver em uma situação que exige o uso de protocolo proprietário s ou transportes.

Resumo

Com quase 15 anos desde que as falácias foram elaborados e mais de 40 anos desde que começamos a construir sistemas distribuídos, as características e os problemas subjacentes de sistemas distribuídos permanecer praticamente o mesmo. O que é mais alarmante é que arquitetos, designers e desenvolvedores ainda são tentados a onda alguns destes problemas fora de pensar a tecnologia resolve tudo. Lembre-se que os aplicativos (bem sucedidos) evoluir e crescer assim mesmo se as coisas parecem OK por um tempo, se você não prestar atenção às questões abrangidas pelas falácias eles vão traseiro sua cabeça feia e mordê-lo. Espero que a leitura deste trabalho tanto ajudou a explicar o que significam as falácias, bem como fornecer algumas orientações sobre o que fazer para evitar as suas implicações.