



Bem vindo ao 7 Firebird Developers Day

“Definindo a abordagem ideal para sistemas com usuários em múltiplos sites”

Wagner Corrêa
Ramos
OBJECT Sistemas

object.com.br





- Apresentação
- Introdução
 - Introdução ao problema e quais soluções a comunidade de TI vem adotando para sistemas com usuários em mais de um local físico
- Análise das soluções
 - Análise comparativa das opções conhecidas para tratar este tipo de problema
- Exemplos
 - Alguns projetos exemplificando cada tipo de arquitetura
- Conclusões
 - Tentando descobrir a melhor abordagem para cada problema
- Questões

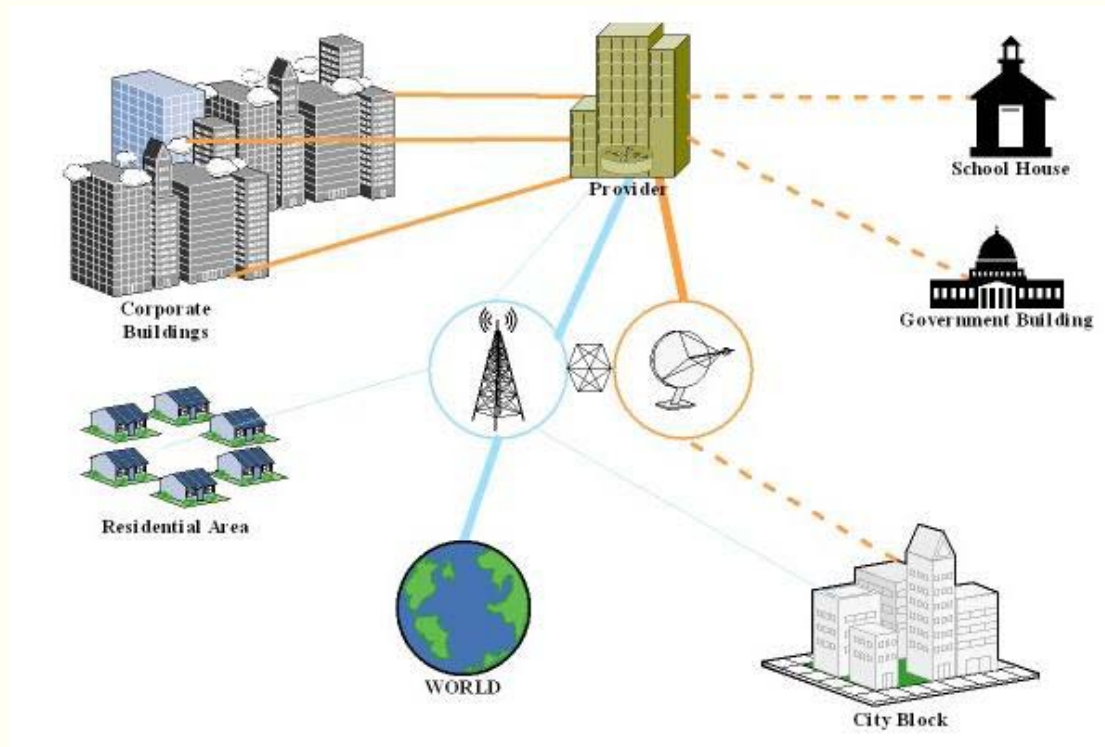


Wagner Corrêa Ramos

- Diretor da OBJECT Sistemas
- Desenvolvedor de software desde 1985
- Formação: Ciência da Computação, Unicamp 1998, MBA TIGEN FGV 2007
- BASIC, COBOL, DATAFLEX, SQLWINDOWS, CENTURA, PERL, PHP, JAVA J2EE.
- SQL SERVER, INFORMIX, ORACLE, SYBASE, MYSQL, POSTGRESQL, FIREBIRD, DB2.
- Experiência em arquitetura de aplicações com usuários em múltiplos sites desde 1997
- Idealizador e desenvolvedor do software **ObjectMMRS** (Object Multi-Master Replication Suite) 2010-2002
- Atualmente focado em implantação de projetos de integração de sistemas e replicação de banco de dados



- Definindo a abordagem ideal para sistemas com usuários em múltiplos sites



Problemas ?

Disponibilidade,
Desempenho,
Segurança,
etc



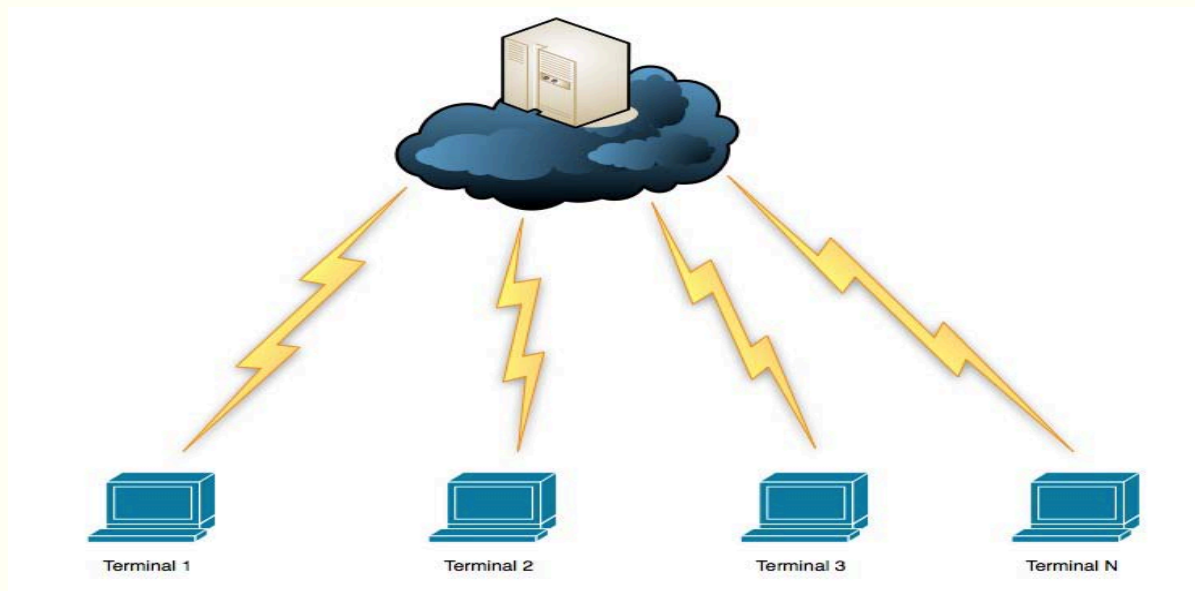
- Qual o maior problema quando temos usuários espalhados geograficamente ?
 - Disponibilidade ? (Venda pela Internet)
 - Desempenho ? (Checkout de supermercado)
 - Segurança ? (Segurança pública, Inteligência)
 - Consistência temporal ? (Operações bancárias)
 - (Para cada tipo de negócio temos um grau de importância diferente para estes problemas).
- Qual a solução ideal ?
 - Uma super infra-estrutura de rede ? Por exemplo, ligar todos os sites com fibra óptica ou rádio ? (se o custo permitisse...)
 - Estudar a necessidade do cliente e apresentar 2 ou 3 alternativas para que ele escolha de acordo com o quanto ele pode investir no momento



Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

1) Aplicação centralizada e acesso via terminal burro

- ♦ Terminal Burro (Dumb) é um tipo de equipamento para conexão em computadores através de portas seriais e tem habilidade única de envio e apresentação de informações em modo texto.
- ♦ Aplicação centralizada rodando em mainframe, unix, etc.



object.com.br

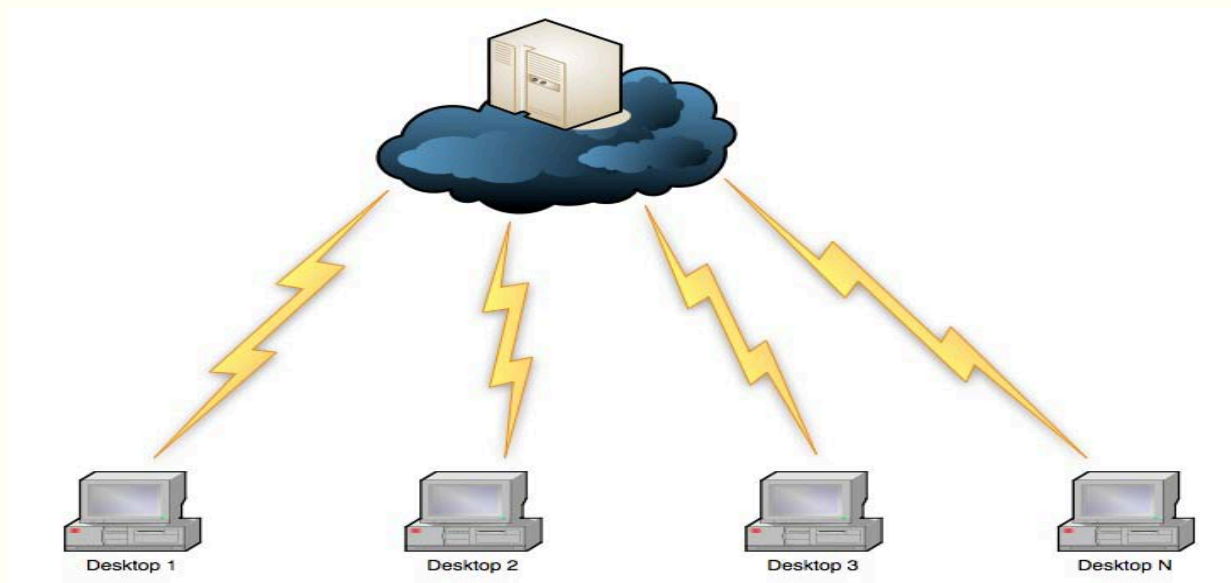




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

2) Aplicação centralizada e acesso via emuladores de terminal texto

- Programa rodando em microcomputadores com capacidade de emular um terminal burro. (VT100, VT220, etc)
- Microcomputador rodando Windows, Linux, etc, mas a aplicação continua centralizada em ambiente mainframe, unix, etc.

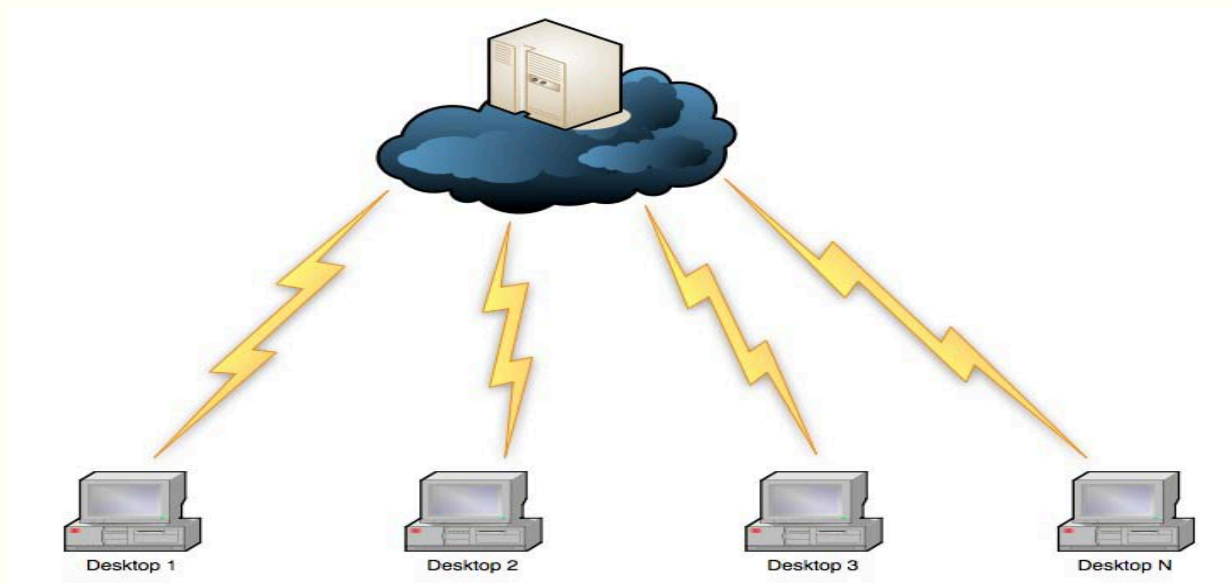




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

3) Aplicação centralizada e acesso via emuladores de terminal gráfico

- Programa rodando em microcomputadores para emular a área de trabalho de um microcomputador remoto. (VNC client, Conexão de área de trabalho remota do Windows, etc).
- Aplicação rodando em “plataforma baixa”

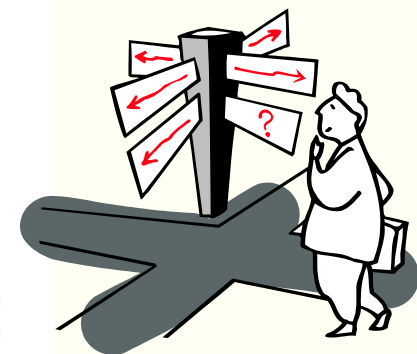
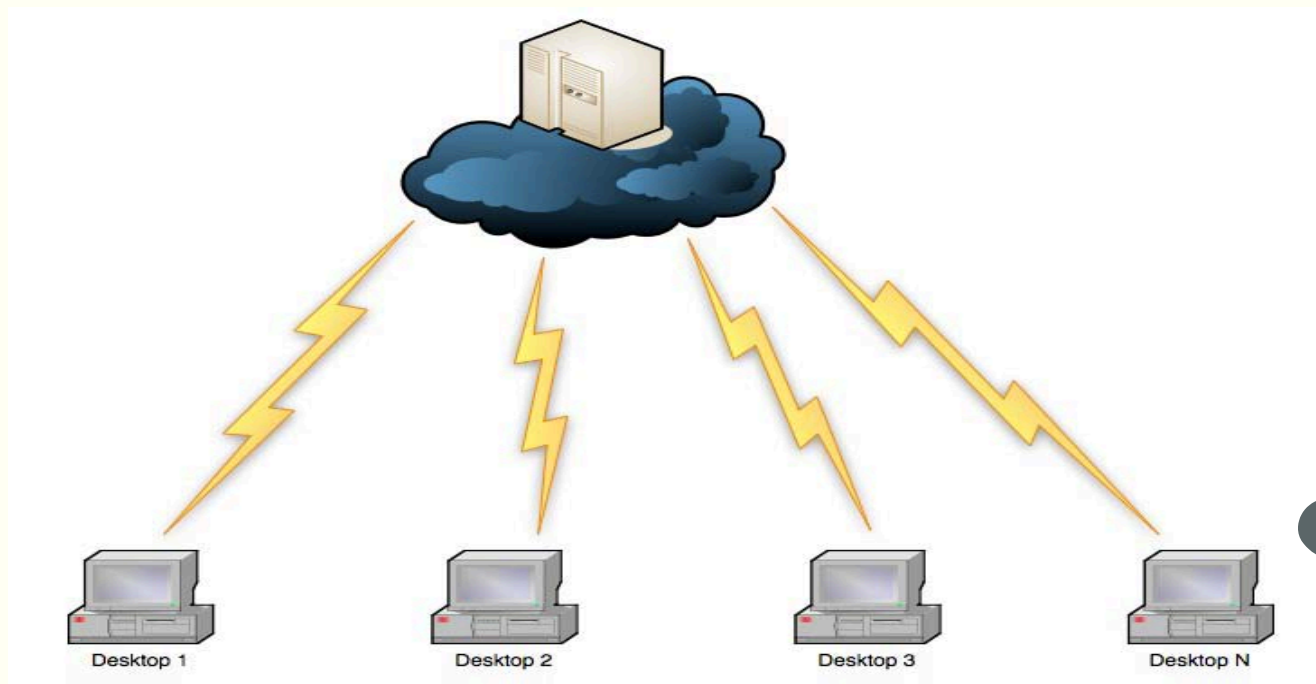




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

4) Aplicação centralizada e acesso via navegador Web

- Interface web



object.com.br

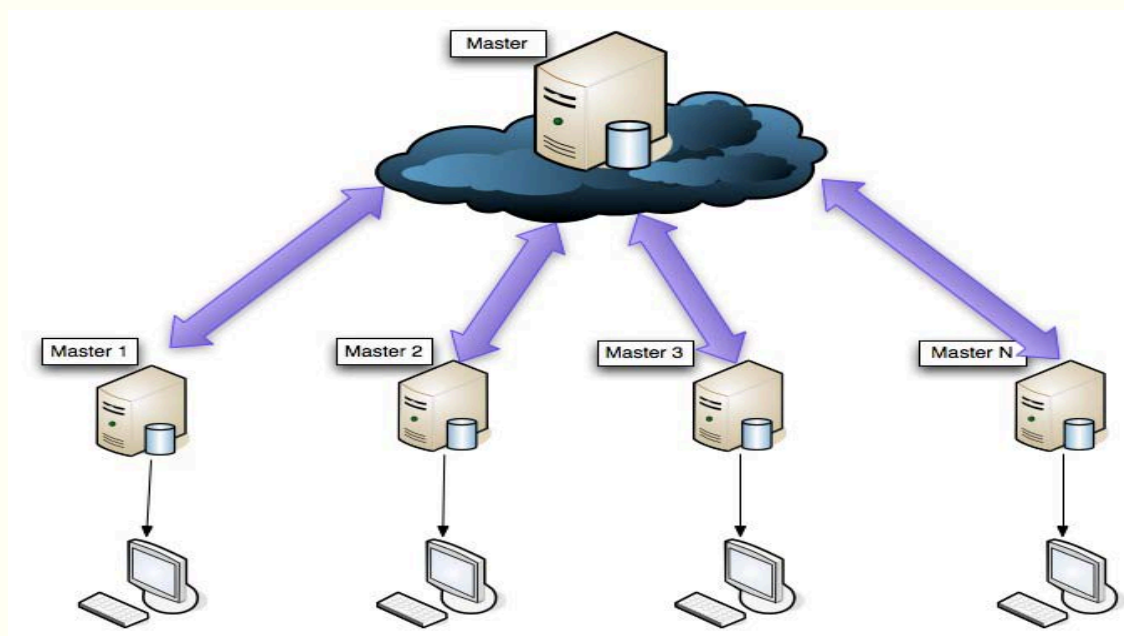




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

5) Base de dados distribuída

- A idéia é que uma aplicação possa se conectar a um único local e o próprio gerenciador de banco de dados escolha o local físico onde ele vai buscar os dados, dependendo de cada query. Os dados não ficam replicados mas sim “distribuídos”, ficando fisicamente próximos de onde mais se usa cada dado.



object.com.br

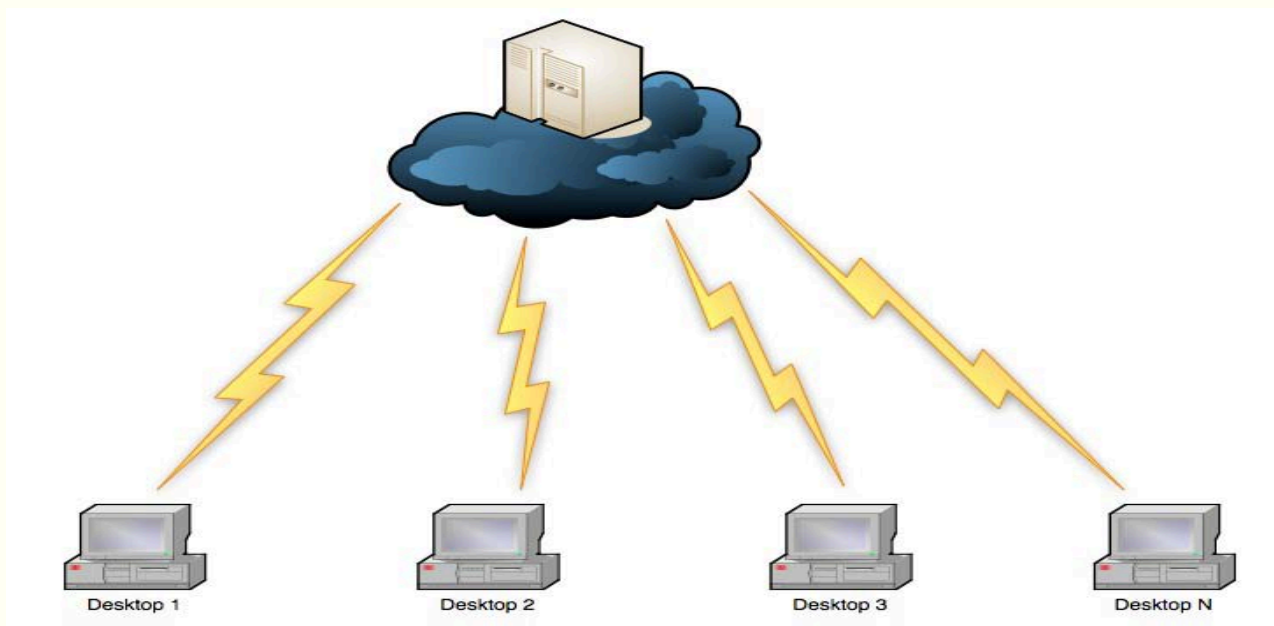




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

6) Aplicação distribuída em arquitetura SOA

- Servidores e clientes se comunicando via troca de mensagens (XML, WS).
- Não precisa ser um único servidor central, pode-se ter N servidores de aplicação conectados a um único banco de dados.



object.com.br

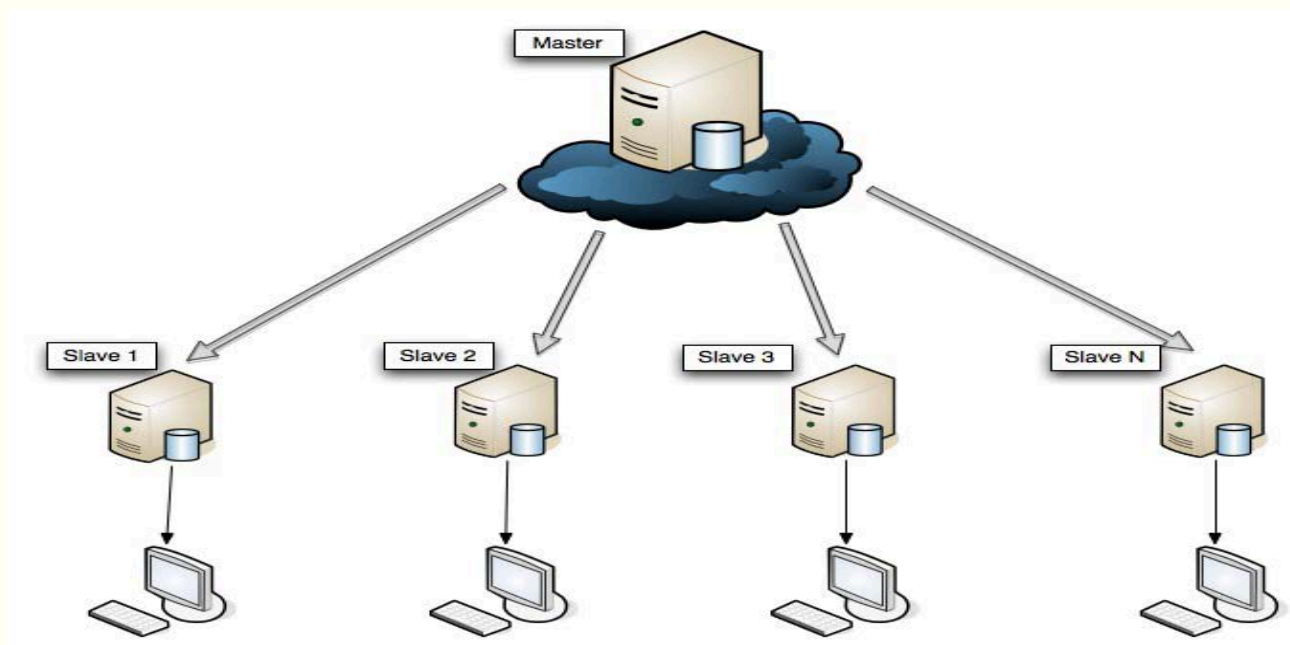




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

7) Bases de dados replicadas Master / Slave

- Aplicação cliente/servidor tradicional
- Aplicação web descentralizada



object.com.br

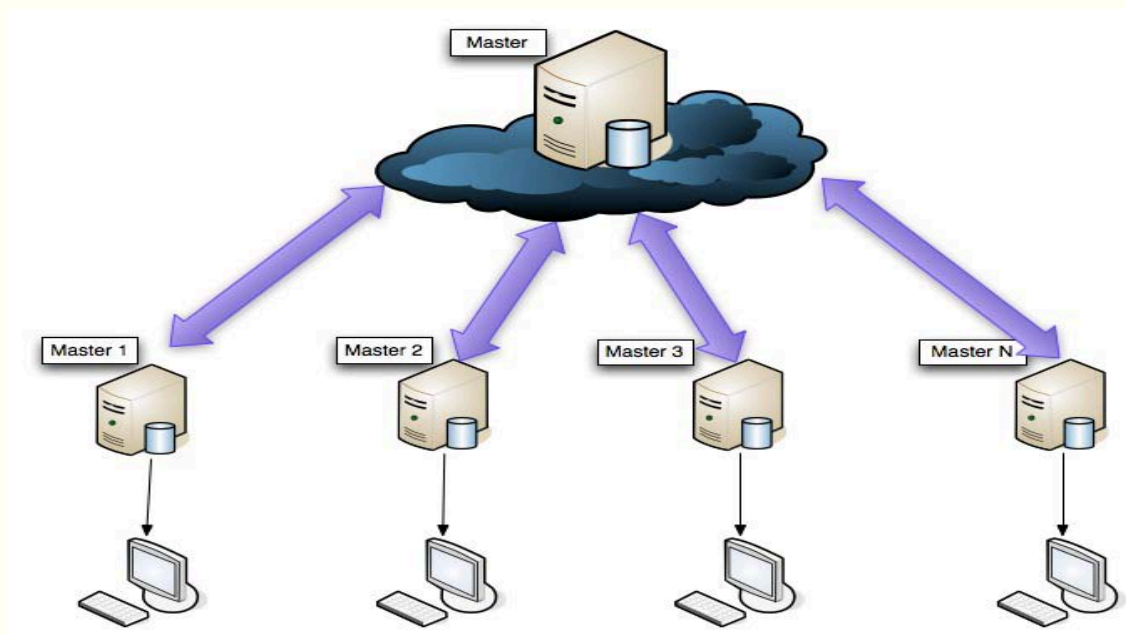




Como a comunidade de TI vem resolvendo o problema de usuários espalhados geograficamente com o decorrer do tempo ?

8) Bases de dados replicadas Multi-Master

- Aplicação cliente/servidor tradicional
- Aplicação web descentralizada



object.com.br





- Analisaremos os prós e contras de cada uma das soluções
 - 1) Aplicação centralizada e acesso via terminal burro
 - 2) Aplicação centralizada e acesso via emuladores de terminal texto
 - 3) Aplicação centralizada e acesso via emuladores de terminal gráfico
 - 4) Aplicação centralizada e acesso via navegador Web
 - 5) Base de dados distribuída
 - 6) Aplicação distribuída em arquitetura SOA
 - 7) Bases de dados replicadas Master / Slave
 - 8) Bases de dados replicadas Multi-Master



1) Aplicação centralizada e acesso via terminal burro

Prós	Contras
Tecnologia completamente dominada, robusta.	Tecnologia descontinuada, sem novos equipamentos
Pouca necessidade de banda de rede, trabalha com modems de 9600 bps (aprox.8kb)	O terminal trabalha com uma interface RS232/RS422, que requer um modem com este tipo de interface, hoje de alto custo.
Queda de rede não causa inconsistência, apenas interrompe o trabalho. Ao reconectar pode-se continuar exatamente do ponto em que parou, a tela vai estar lá do jeito que estava antes.	O processamento é todo central, com isso exige-se um computador de alto desempenho para poder comportar uma quantidade grande de usuários.
	Inúmeros fatores contra, como falta de recursos humanos para a programação, tecnologia estagnada, hardware de elevado preço, etc



2) Aplicação centralizada e acesso via emuladores de terminal texto

Prós	Contras
Sobrevida às aplicações centralizadas com terminal burro	Software não consegue evoluir e acompanhar as novas necessidades da empresa
Pouca necessidade de banda de rede, substituição dos antigos hardwares de terminal burro por um emulador em um micro.	Alguma dificuldade de configuração dos terminais, e dificuldade do usuário ter de se adaptar a 2 ambientes completamente diferentes
Queda de rede não causa inconsistência, apenas interrompe o trabalho. Ao reconectar pode-se continuar exatamente do ponto em que parou, a tela vai estar lá do jeito que estava antes.	O processamento é todo central, com isso exige-se um computador de alto desempenho para poder comportar uma quantidade grande de usuários.
Permite a integração mesmo que com copy/paste entre a aplicação antiga e as novas	Inúmeros fatores contra, como falta de recursos humanos para a programação, tecnologia estagnada, hardware de elevado preço, etc



3) Aplicação centralizada e acesso via emuladores de terminal gráfico

Prós	Contras
Alternativa ao uso de cliente-servidor (cliente remoto) conectando direto ao banco de dados via TCP/IP	Exige rede estável, pode ter pouca banda mas necessita estabilidade para o usuário ter um desempenho satisfatório. No lado server precisa de boa infra de rede com banda suficiente para suportar todos os clientes.
Queda de rede não causa inconsistência, apenas interrompe o trabalho. Ao reconectar pode-se continuar exatamente do ponto em que parou, a tela vai estar lá do jeito que estava antes.	O processamento é todo central, com isso exige-se um ou mais servidores de aplicação de alto desempenho para poder comportar uma quantidade grande de usuários.
Permite a integração mesmo que com copy/paste entre a aplicação emulada e uma aplicação local	Problemas para impressão de documentos, relatórios, interoperabilidade com aplicações locais.



4) Aplicação centralizada e acesso via navegador Web

Prós	Contras
Excelente para aplicações com muitos usuários remotos, muitas consultas rápidas, e uso moderado de teclado (pouca entrada de dados)	Problemas da plataforma HTML criada com a finalidade de exibição de documentos estáticos, e que hoje tentamos fazer com que ela seja uma plataforma rica para entrada de dados e exibição dinâmica de dados.
	O processamento é todo central, com isso exige-se um ou mais servidores Web de alto desempenho para poder comportar uma quantidade grande de usuários.
Queda de rede pode não comprometer a consistência mas depende da programação realizada.	Ao reconectar a maioria das aplicações web não permite continuar no ponto em que parou, a tela NÃO vai estar lá do jeito que estava antes. Nem sempre é possível dar um back e tentar novamente. (Problemas de duplo submit, etc)
Permite alguma integração com aplicações locais	Complexidade para o desenvolvimento, pouca produtividade.



5) Base de dados distribuída

Prós	Contras
A preocupação com o acesso aos dados é do gerenciador de banco de dados e não da aplicação	Muito estudado, muito discutido, mas praticamente nenhuma solução comercial ou open-source disponível. Ainda um sonho apenas.
Toda a complexidade é transferida ao gerenciador de banco de dados distribuído	Boa produtividade no desenvolvimento da aplicação, caso pudessemos abstrair o problema do acesso.



6) Aplicação distribuída em arquitetura SOA

Prós	Contras
Bancos de dados locais	Integração entre os bancos de dados é feita via troca de mensagens (web-services), e uso de middlewares de message-queues (MQSeries por exemplos). Alto custo, alta complexidade.
Interoperabilidade total, os sistemas ficam “abertos” garantindo assim o investimento realizado e posterior evolução sem reengenharias.	Baixa produtividade no desenvolvimento da aplicação, ainda muito complexo.



7) Bases de dados replicadas Master / Slave

Prós	Contras
Grande parte do acesso a banco de dados em rede local	A atualização de dados só pode ser feita no servidor master central. Então resolve-se apenas parte do problema, ou seja, as consultas são agilizadas mas todas as transações ainda são centralizadas.
Boa parte da aplicação rodando de forma descentralizada, com acesso local ao banco de dados	Parte transacional da aplicação ainda necessitando rodar de forma centralizada, seja via aplicação web ou via emulador de terminal gráfico.
Administração mais simples que uma replicação multi-master	Com pouco mais esforço consegue-se uma replicação multi-master com praticamente a mesma infra-estrutura usada para a replicação master-slave, dependendo apenas de um software replicador multi-master.



8) Bases de dados replicadas Multi-Master

Prós	Contras
Banco de dados sempre local	A aplicação precisa em algumas situações ser adaptada para poder trabalhar de forma multi-master.
Resolve-se além do problema de desempenho e de disponibilidade, dependendo da configuração o problema de backup também	Administração do ambiente, mais complexo do que administrar uma base única central. Precisa de um produto de replicação que seja de fácil entendimento, transparente e não caixa-preta.
Como o processamento pode ficar completamente descentralizado, os servidores usados podem ser de baixo custo	Maior número de servidores a administrar, precisa de uma política de administração de servidores bem elaborada, e preferência para sistemas operacionais facilmente administrados remotamente.
Economia em banda de rede.	



Arquitetura	Disponibilidade	Desempenho	Segurança
1/2) Centralizada terminal burro / emulador texto	C	C	C
3) Centralizada emulador gráfico	B	C	B
4) Centralizada Web	C	C	C
5) Base distribuida	B	B	C
6) Distribuida SOA	B	B	B
7) Master Slave	B	B	B
8) Multi Master	A	A	A

	A	B	C
Disponibilidade	N servidores, rede local	N servidores, rede WAN	1 servidor, rede WAN
Desempenho	Todas operações em servidor local	Algumas operações em servidor local	Todas operações em servidor remoto
Segurança	Sem risco de parada total	Risco de parada parcial	Risco de parada total





- **Telefonica - Aplicação de Auxilio à Lista (102) - 1999**
 - 1999, Aplicação centralizada emulador de terminal texto.
 - Aplicação em arquitetura CORBA, com interface padrão Windows para usuário final e daemon rodando em N servidores fazendo “raspa de tela” de emuladores de terminal texto.
 - Aprox. 1500 usuários em cada site / call center.
 - Cada site com aprox. 6 servidores CORBA acessando o mainframe central
 - Tecnologias: C++, Servidores com Sun Solaris, Estações cliente com Windows NT Workstation, BD Oracle para tarifação.
- **Secretaria de Segurança Pública do Estado de SP – Gestão Prisional - 2005**
 - Aplicação inicialmente centralizada e acesso via emulador de terminal texto
 - 2007, New GEPEN, Aplicação distribuída com uso de replicação Multi-Master (ObjectMMRS), e integração com o sistema legado via web-services.
 - Aprox. 2000 usuários no total, sendo aprox. 10 usuarios remotos em cada uma das 142 unidades prisionais.
 - Tecnologias: Mainframe Unisys, Java/J2EE, PHP, Oracle, PostgreSQL, ObjectMMRS, Fedora 4, Windows



- **Penha S/A – Sistema de Vendas – 1996**
 - Aplicação cliente-servidor
 - Tentativa de conexão direta com o banco via TCP/IP dos usuários de vendas na cidade de SP e servidor de banco de dados em Itapira.
 - Solução Master / Slave
 - Tecnologias: Digital Unix, Windows, SQLWindows, Informix.
- **Allisson – Software de gestão para Transportadoras - 2010**
 - Atualmente Aplicação centralizada usando o recurso de terminal server do Windows para atender usuários remotos
 - Migrando para solução Multi-Master
 - Tecnologias: Windows Server, Delphi, Firebird, ObjectMMRS
- **Locaweb – Website de venda de autos - 2008**
 - Aplicação centralizada Web
 - Inicialmente um servidor único de banco de dados
 - Atualmente 5 servidores de banco de dados trabalhando de forma Master / Slave para distribuir a carga e suportar aprox. 400 mil acessos diários.
 - Tecnologias: PHP, Postgresql, ObjectMMRS



- EMBRAPA – Sistema de Projetos Agropecuários – 2000
 - Uma aplicação em Delphi com dados locais em formato texto/componente do Delphi
 - Uma aplicação Web central em java/j2ee, e banco Oracle
 - Um middleware fazendo a “integração” da base local com a base central, transmitindo dados em XML.
 - Tecnologias: Windows, RedHat, Delphi, XML, Java/J2EE, Oracle
- Shibata – Sistema de gestão de supermercados – 2009
 - Inicialmente era uma Aplicação Web com uso de replicação master-slave, com 1 servidor slave em cada loja e 1 servidor master central
 - Migrado para multi-master, agora com 1 servidor master em cada loja e mais um central
 - Tecnologias: Slackware, PHP, Postgresql, Slony, ObjectMMRS



- **CIGS Sistemas – Software de gestão para indústrias – 2007**
 - Software desenvolvido para o uso descentralizado multi-master
 - 1 servidor com Firebird em cada unidade industrial.
 - Tecnologias: Windows Server, Firebird, Delphi, ObjectMMRS
- **SG Sistemas – Software de gestão para comércio - 2008**
 - Software desenvolvido para o uso descentralizado multi-master
 - 1 servidor com Firebird em cada loja / filial.
 - Tecnologias: Suse, Firebird, Delphi, ObjectMMRS
- **Marinha do Brasil – Software para gestão hospitalar - 2010**
 - Software desenvolvido para o uso descentralizado multi-master
 - 1 servidor em cada unidade de saúde da Marinha (42 no total) e 1 servidor central.
 - Tecnologias: Suse, Java/J2EE, Postgresql, ObjectMMRS



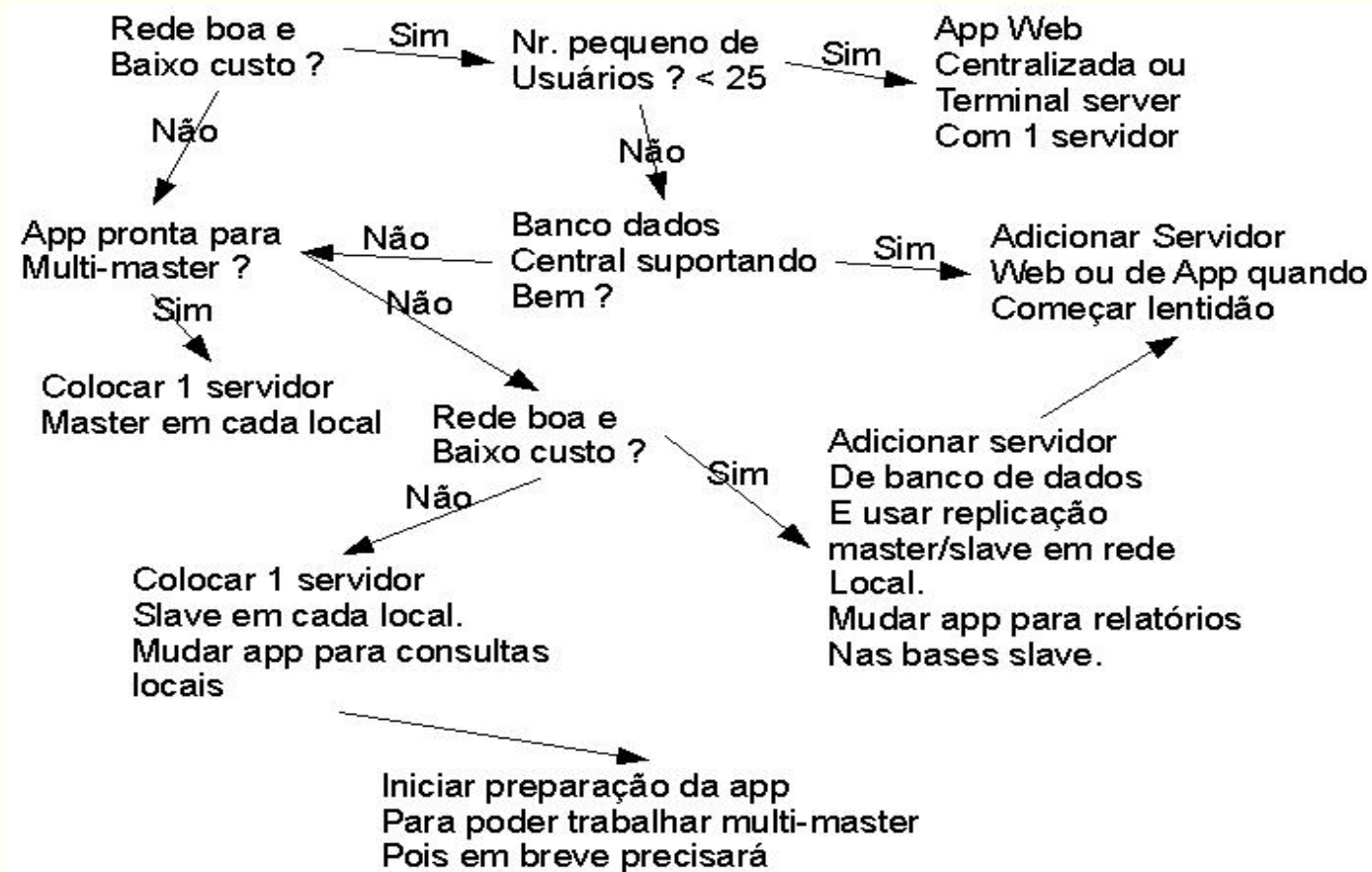
- Existe uma única solução ideal para todos os casos ?
 - Cada negócio exige uma análise e uma solução adequada
- Procurar estudar o negócio, avaliar questões como:
 - O que acontece se o sistema fica fora do ar ? (Consegue-se ou não prosseguir com o trabalho mesmo que lentamente ?) Exemplo: Padaria x Vendas pela Internet
 - É possível medir a perda quando o sistema está fora de operação ? Exemplo: Serviço 102
 - Qual o impacto no negócio de um sistema lento ? Perde uma venda ? Perde o cliente para sempre ?
 - Quanto vale o banco de dados para a empresa ? Se houver uma perda total de dados a empresa prossegue normalmente ?
 - Se o sistema está inacessível em um lugar é possível dar continuidade no negócio através de envio de FAX ou telefonema para outras unidades da empresa ?



- Exemplos de coisas que acontecem com a lentidão de sistemas centralizados
 - Investimentos enormes em sistemas de gestão estilo SAP acabam não trazendo resultados esperados porque a empresa se esquece, por exemplo, que a entrada de dados precisa ser feita em obras, onde não existe boa infra de rede para se rodar uma app web pesada ou usar um terminal-server. Com isso toda a velocidade na tomada de decisão que era esperada com o novo sistema perde-se porque a entrada de dados acaba sendo offline via planilhas, emails, etc e com posterior retrabalho.
 - Um sistema sofisticado de gestão prisional que precisa da entrada de dados frequente dos agentes penitenciários, acaba não ocorrendo pela lentidão do mesmo, então todas as informações úteis que poderiam ser usadas pela equipe de Inteligencia da Polícia é desperdiçada.
 - Um serviço de solicitação de passaporte que poderia atender por exemplo 10 pessoas em uma hora acaba atendendo 3 ou 4. Com isso a solução acaba sendo em abrir novos escritórios, com mais gente, mais aluguel, infra, etc.
 - Um sistema de controle de acesso demora irritantes 10 ou 15 segundos para liberar uma catraca.



- Sugestão de abordagem para sistemas com usuários em múltiplos sites





- Porque sempre procuro adotar o uso de replicação multi-master em meus clientes ?
 - Além de ser fabricante de um software replicador multi-master...
 - Mesmo em casos onde tem-se boa rede e de baixo custo, e/ou com quantidade pequena de usuários, ao usarmos replicação multi-master ganhamos “segurança”, ou seja, um hot-backup pronto que pode ser usado por todos enquanto o outro servidor estiver em manutenção. Além disso deixamos a banda de rede para outros usos da empresa ao invés de ocuparmos com emuladores.
 - É trabalhoso entender o funcionamento de um software de replicação, mas o retorno é muito favorável, pois você terá em mãos uma solução para atender pequenos e grandes clientes dando a ambos a melhor qualidade com o menor custo possível.
 - Mas lembre-se sempre, cada caso é um caso.



Questões ?

Agradeço à presença de todos !

Wagner Corrêa Ramos

wagner@object.com.br

object.com.br

