

Front End para Bancos de Dados

22/01/2021

Autor: Ozir F.Andrade Zotto

No novo e ainda não padronizado ambiente de processamento distribuído, os riscos de desenvolver aplicações que falhem nos requerimentos de performance, integridade ou segurança são muito reais. Uma forma de reduzir estes riscos, é a escolha apropriada de ferramentas e ambientes que reduzam a complexidade para os desenvolvedores.

Este artigo - resumo de uma pesquisa apresentada na disciplina Sistemas Distribuídos, como parte do Curso de Pós-graduação em Ciências da Computação na Universidade Estadual de Londrina - analisará algumas ferramentas que podem ser utilizadas para o desenvolvimento de aplicações em computação distribuída.

Este curso, que está sendo ministrado por professores da Universidade Federal do Rio Grande do Sul, é fruto de um convênio de cooperação técnica e financeira realizado entre a Celepar - Cia. de Informática do Paraná, a UEL - Universidade Estadual de Londrina e a Sercomtel - Serviço de Comunicações Telefônicas de Londrina, e conta com a participação de 4 funcionários da Celepar.

1. Introdução

A computação cliente/servidor é um irresistível movimento que está remodelando a forma como os computadores estão sendo utilizados. As grandes transformações na indústria de computadores são usualmente direcionadas pela combinação de novas tecnologias e necessidades. As principais tecnologias e necessidades que estão estimulando o desenvolvimento da computação cliente/servidor vêm de três direções *downsizing*, *upsizing* e *rightsizing*.

O movimento de pequenos grupos de trabalho, concentrados em aplicações cliente/servidor para uma segunda geração de tecnologia em escalas de corporações unidas em multicamadas de gerenciamento, irá forçar os Sistemas de Informação para maiores mudanças. Os talentos e disciplina necessários para integrar os numerosos e diferentes serviços são difíceis, essenciais e caros. Uma arquitetura será essencial para minimizar custos e maximizar os benefícios (13).

A arquitetura dos sistemas gerenciadores de banco de dados (SGBD) em

mainframes costuma ser radicalmente diferente das arquiteturas de sistemas distribuídos baseadas em Unix ou DOS. Estas diferenças podem causar problemas quando é necessário unir os dois ambientes para trabalharem juntos (4).

A evolução das arquiteturas cliente/servidor ocorre em ondas que atentam para as necessidades emergentes (18).

A primeira onda iniciou com experimentos acadêmicos de sistemas operacionais servidores de arquivos (3) e culminou com as redes locais e remotas, que tornam fácil para as aplicações o compartilhamento de arquivos, impressoras e outros equipamentos.

Atualmente, estamos no meio da segunda onda, com aplicações centralizadas em banco de dados. A tecnologia predominante é o "servidor de banco de dados SQL". Entretanto, também estamos vivenciando o encrespamento de duas tecnologias, osaxionadas por *Groupware* e *Tp Monitors*.

A terceira onda de tecnologia cliente/servidor é a dos objetos distribuídos. Objetos abrangem todas as tecnologias da primeira e da segunda onda e adicionam consideráveis novos valores. Os objetos têm o potencial único de distribuir inteligência entre os clintes e os servidores onde á mais necessária (18).

Embora a Internet não seja uma tecnologia emergente, o exponencial aumento de popularidade e crescimentos da "rede das redes" é o mais recente frenesi que extrapola aos circunstanciados ambientes dos negócios e dos computadores, apontando para mudanças ainda inimagináveis no tratamento das informações (6, 20) e na própria relação entre os seres humanos (19).

Esta introdução mostra que as transformações na indústria de computadores e direcionadas por novas tecnologias e necessidades, e que a evoluçõa da tecnologia cliente/servidor procura atender às mudanças nos ambientes de negócios e de computação. A partir dest pequeno levantamento bibliográfico, este trabalho analisará algumas ferramentas que podem ser utilizadas para o desenvolvimento de aplicações em computação distribuídas.

A seção 2 aborda as mudanças ocorridas nos ambientes de negócio e de computação; a seção 3 resume uma classificação proposta pela IBM no documento "Application Development Tools for Distributed Computing" (2), onde uma análise mais detalhada destas ferramentas pode ser encontrada; a seção 4 analisa algumas das ferramentas front-end do modelo centrado em recursos; a seção 5 conclui introduzindo o que vemos como sendo a próxima etapa das

tecnologias cliente/servidor: objetos distribuídos e intranet.

2. As mudanças nos ambientes

2.1 O ambiente do negócio

A razão econômica para a computação distribuída torna-se cada vez mais atrativa. Na proporção que os preços dos equipamentos diminuem, a performance aumenta e o conhecimento para executar trabalhos com auxílio do computador também aumenta. As principais teorias de administração hoje privilegiam um maior grau de responsabilidade e autonomia ao trabalhador, serviço receptivo ao cliente, conhecimento organizacional, reengenharia de processos e sensibilidade temporal com o mercado. Conseguir informações relevantes no exato momento em que são necessárias, direto na e para as estações de trabalho e servidores heterogêneos, são os desafios que os desenvolvedores de aplicações enfrentam neste novo e complexo ambiente de computação (2).

2.2 O ambiente de computação

Cliente/servidor, computação distribuída, processamento cooperativo, independente da terminologia desta emergente arquitetura de computação, o movimento de abandono aos monolíticos sistemas baseados em mainframes para sistemas distribuídos está forjando uma imprevista complexidade no desenvolvimento das aplicações. As redes e as infra-estruturas de comunicações estão, simultaneamente, ficando mais importantes e complicadas. Aplicações espalhadas em múltiplas plataformas estão substituindo monolíticas aplicações, processando em uma única plataforma. Os técnicos agora encaram o desafio de construir aplicações que unem sistemas de informação heterogêneos, com múltiplas plataformas, banco de dados, protocolos e localizações.

3. Classificação das ferramentas

A classificação proposta pela IBM (2) distingue seis diferentes modelos para computação distribuída:

1. "Front-ending" com interfaces gráficas para sistemas herdados residentes em mainframes.
2. Aplicações com lógica distribuída através de coleções de clientes e servidores.
3. "Data-staging" com os dados distribuídos para os clientes no front-end ou nos servidores departamentais.

4. Modelo centrado em recursos que tornam disponíveis informações de apoio à decisão aos usuários finais.
5. Modelo direcionado aos processos para automatização de tarefas pelo computador.
6. Modelo com múltiplas aplicações que encapsulam incompatíveis sistema para permitir a sua interoperabilidade.

Estes modelos de referências são agrupados em duas categorias distintas: Os quatro primeiros são chamados de Modelos de Referência Fundamentais, por se caracterizarem por um mapeamento um-por-um com as ferramentas disponíveis. Os demais são denominados de Modelos Compostos, pois precisam de uma composição e combinação de diversas das ferramentas existente, para conseguir as características técnicas necessárias ao desenvolvimento destes tipos de aplicações.

Há muitas maneiras de implementar sistemas de computação distribuída que sirvam aos requerimentos dos negócios. Nenhuma solução isolada cobre todas as necessidades. Frequentemente as organizações implementam diversos destes modelos simultaneamente, às vezes até mesmo numa mesma aplicação.

Estes modelos podem servir apenas como pontos de referência para determinar estratégias de computação distribuída e para auxiliar na seleção de apropriadas ferramentas de desenvolvimento.

3.1 Modelos de referência fundamentais

3.1.1 Front-endings

Em sua maior parte, estes modelos de referência abrangem "raspadores de telas" que providenciam interfaces gráficos de acesso aos sistemas baseados em mainframe para as estações de trabalho baseadas em Unix, PCs ou Macs. As aplicações no mainframe continuam processando sem modificações. Do ponto de vista do mainframe, ele está sendo acessado por um "terminal burro".

Os principais benefícios deste modelo são as habilidades para combinar aplicações das estações de trabalho com aplicações no mainframe, juntar os recursos de múltiplos sistemas herdados baseados no mainframe para o usuário final e adicionar valor às aplicações do mainframe, incluindo lógicas adicionais e/ou operações matemáticas aos dados recebidos do mainframe.

Basicamente, este é um primeiro passo na transição de sistemas herdados para

outros modelos de referência mais distribuídos.

Os principais benefícios dos modelos "front-end" são a habilidade de entregar rapidamente a funcionalidade das aplicações nas estações de trabalho, sem modificações nas aplicações do mainframe; a integração dos recursos do mainframe com as estações de trabalho; e, finalmente, as aplicações no mainframe permanecem inalteradas.

Os principais problemas incluem o fato de que modificações na aplicação no mainframe podem romper o conteúdo das telas no "front-end". Também os sistemas herdados frequentemente sofrem de falta de adequada documentação tornando difícil desenvolver e usar aplicações que pressupõem um certo layout de tela. Finalmente, "front-end" não corresponde muito no caminho de flexibilidade e poder prometidos pelo processamento distribuído. Esta não é uma abordagem flexível e escalável para a computação distribuída.

Easel, KnowledgeWare, Systema Strategies Express, Moart e IBM Cliente/Server for Windows são as principais ferramentas que suportam o rápido desenvolvimento de modelos de referência front-ending.

3.1.2 Lógica distribuída

Com a lógica distribuída, partes da aplicação residem em computadores da rede. Tipicamente uma aplicação executada em mais de uma plataforma, com os serviços de comunicação assegurando a sincronização dos processos distribuídos que são chamados. Todas as ferramentas classificadas como de Lógica Distribuída no documento (2) - Seer HPS, Dynsty Development Environment, Texas Instruments Information Engineering Facility, e Andersen Foundation for Cooperative Processing - provêem um alto nível de controle sobre o processo de desenvolvimento. Isto inclui o projeto da aplicação em um ambiente de desenvolvimento, alguma forma de participar a aplicação para distribuir a lógica entre o cliente e o servidor, e suporte para múltiplos desenvolvedores.

Lógica distribuída é um modelo altamente escalável e sofisticados que pressupõe substanciais recursos, e um compromisso organizacional sério para implementar o projeto. Há muito poder no particionamento de lógica e na geração automática de código. Entretanto, uma vez que o projeto está iniciado, o sucesso depende da habilidade da ferramenta para entregar 100% da funcionalidade necessária para construir e disponibilizar a aplicação. recorrer à codificação manual e modificações de baixo nível são dispendiosas, consomem tempo e um risco para o desenvolvimento de aplicações de lógica distribuída.

3.1.3 Data Staging

O modelo de referência "data staging" torna dados transacionais disponíveis em um formato que os tomadores de decisão e os analistas podem entender e analisar. "Data staging" contrasta com dados transacionais, que são difíceis para os profissionais do negócio manipular. "Data staging" significa fazer os dados serem acessíveis para os usuários finais, em termos de localização (no local e servidores de alta performance), conteúdo (campos calculados podem ser necessários) e formato (dados transacionais não normalizados).

O benefício deste modelo é a disponibilização dos dados para a tomada de decisões. O inconveniente primário é que a maior parte das tecnologias disponíveis é limitada a operações "read only". Outro problema potencial é que o tempo entre as atualizações determina a validade atual dos dados. Muitas destas soluções são dispendiosas para implementar.

As seguintes ferramentas "Data Staging" são citadas no documento da IBM (2): Red Brick Warehouse, Trinzic InfoPump, Logical Data Integrator da Mentor Communications e Prism Warehouse Manager.

3.1.4 Modelo centrado em recursos

O modelo centrado em recursos distribui os recursos especializados, que estão espalhados por toda a organização, para a estação de trabalho. O compartilhamento de recursos entre múltiplos clientes é o direcionamento fundamental para este tipo de modelo. As chaves para este modelo são facilidades de comunicação confiáveis para endereçar pedidos de recursos para o apropriado provedor e direcioná-los de volta para o clientes, e tecnologias de servidor com suporte para múltiplos usuários. Tipicamente, um ambiente de desenvolvimento gráfico que simplifica a construção de consultas SQL, ad-hoc para os usuários finais também é uma característica.

Muitas das ferramentas hoje disponíveis no mercado são fortemente direcionadas para providenciar intuitivo acesso gráfico para servidores de banco de dados relacionais. Dispositivos para acessar múltiplos banco de dados e bando de dados heterogêneos também são incluídos em muitas destas ferramentas.

A principal inconveniência é a falta de controle sobre os recursos consumidos.

A popularidade e a grande quantidade de ferramentas de desenvolvimento centradas em recursos disponíveis no mercado reflete o atual estado da arte das

arquiteturas cliente/servidor. Os usuários têm estações de trabalho conectadas às redes e precisam acessar bando de dados realiocanis da corporação, mas eles não entendem SQL e nem estão interessados em conhecer layouts de daos arcanos. Muitas destas ferramentas realizam o acesso aos bancos de dados relacionais para esta população, isolando-os dos detalhes técnicos necessários para escrever comandos SQL.

Três abordagens básicas podem ser discernidas pelas empresas produtores destas ferramentas: Construtores de interfaces gráfios (GUI) pra acesso a dados relacionias, Construtores GUI para multiplataformas e "gateways" para banco de dados.

As principais ferramentas do modelo Centrado em Recursos serão analisadas com mais detalhe na seção 4.

3.2 Modelos compostos

3.2.1 Modelo comosto direcionado ao processo

Num modelo de referência direcioinado ao processo, um processo do negócio é executado como uma série de tarefas de processamento através de múltiplas estações. Por ser um modelo complexo para ser criado e mantido, os desenvolvedores podem ser melhor atendidos selecionando ferramentas com forte cpacidade de modelagem do negócio.

Este modelo é melhor indicado para processos rotineiros automáticos, que são complexos ou mororos demais paa agentes humanos.

É possível argumentar que cada negócio é uma série de processos, e que mutias tarefas administratias de rotina poderiam ser automatizadas com este modelo. Entretanto, de um ponto de vista prático, negócios são altamente dinâmicos, o que torna inapropriado este modelo, por requerer uma grande quantidade de trabalho para ser ajustado, e por ser difícil de ser modificado. O principal inconveniente é a falta de tecnologia disponível com poderes suficientes para prover todas as capacidades necessárias para facilmente desenhar, construir e modificar o projeto.

3.2.2 Modelo composto para múltiplas aplicações

De um ponto de vista conceitual, um modelo de referência de múltiplas aplicações "encapsula" incompatíveis aplicações, efetivamente escondendo os recursos de computação e os dados dos outros modelos. As incompatibilidades

são sobrepujadas pelo desenvolvimento de formatos comuns de transferência de dados para a passagem das informações requeridas entre os sistemas.

De um ponto de vista prático, as soluções correntes no mercado não chegam perto da elegância e simplicidade implicadas no termo "encapsulamento".

O benefício primário deste modelo é a integração de sistemas que têm fortes valores funcionais adicionados, mas que precisam ser coordenados horizontalmente para atingir os objetivos do negócio.

4. Análise das ferramentas do modelo centrado em recursos

4.1 Interfaces gráficas para dados relacionais

As ferramentas construtoras de interfaces gráficas para acesso a dados relacionais suportam o desenvolvimento de "front-ends" GUI para isolar os usuários finais do SQL e das considerações de layout dos dados. Estas são as ferramentas abanlisadas: KnowledgeWare Object View, Powersoft PowerBuilder, Gupta SQL Windows, IBM VisualAge, IBM Envy/400, Uniface e Progress. As primeiras cinco são ambientes de desenvolvimento gráfico que provêem conectividade com banco de dados. Aplicações desenvolvidas com esas ferramentas são ideais para suporte à decisão e baixo processamento transacional. Entretanto, para aplicações mais complexas e que exijam maior performance, os desenvolvedores estarão melhor servidos por Uniface e Progress Versão 7, que são ambientes de desenvolvimento 4GL (Fourth Generation Language) com capacidades para construções gráficas. Para organizações com grandes investimentos na plataforma IBM AS/400, a ferramenta IBM ENVY/400 provê aplicações com a habilidade de explorar os recursos desta plataforma.

4.2 Interfaces gráficas para multiplataformas

As ferramentas construtoras de interfaces gráficas (GUI) para multiplataformas atendem à necessidade de entregar as capacidades dos recursos organizacionais para heterogêneas estações clientes. As ferramentas desta categoria são primariamente orientadas para resolverem o problema de construir a mesma aplicação para clientes em múltiplas plataformas. Elas isolam os desenvolvedores das diferenças entre plataformas, provendo um real ambiente de desenvolvimento multiplataforma para interfaces cliente. As ferramentas desta categoria são: Visix Galaxy, Neuron Data Open Interface, Kasew Kase: VIP e NeXT NXTStep.

Visix Galaxy, Neuron Data Open e Kase: VIP suportam o desenvolvimento GUI para múltiplas plataformas. Galaxy e Open Interface têm um ambiente de

desenvolvimento gráfico que permite ao desenvolvedor uma larga flexibilidade no desenvolvimento interativo de interfaces. Kase: VIP faz uma abordagem ICASE mais tradicional, o que significa que mudanças na GUI precisam ser especificadas em prévias partes CASE do produto.

NeXT NeXTStep não gera plataformas GUI por si mesma, mas abre acesso aos arquivos de sistemas da rede. Também provê um bom ambiente de desenvolvimento orientado a objeto, baseado no núcleo Mach. NeXT NeXTStep é única em sua habilidade para trazer recursos não relacionais para a estação de trabalho, incluindo voz e gráficos 3D. A abordagem orientada a objeto provê um ambiente de desenvolvimento rápido e flexível. Em conjunto com produtos de terceiros para acesso aos dados, NeXT e NeXTStep pode ser um poderoso ambiente de desenvolvimento. Entretanto, está limitado para entrega em clientes UNIX.

4.3 Gateways para banco de dados

Os "gateways" para banco de dados citados neste trabalho não são ferramentas para desenvolvimento de aplicações propriamente ditas. Poderiam ser melhor considerados como dispositivos centrados em recursos provedores de acesso a fonte de dados heterogêneas. Essencialmente são uma camada "middleware" com uma linguagem de manipulação de dados e de tradução de formato de dados. Alguns produtos também provêm serviços de endereçamento aos dados, que livram os desenvolvedores de ter que codificar informações de endereçamento nas aplicações.

Estes produtos resolvem o problema de ter diferentes tipos de sistemas de gerenciamento de dados nas aplicações. Os dados podem estar em sistemas de gerenciamento de banco de dados relacionais, hierárquicos, distribuídos em rede ou orientado a objetos. Mesmo no reino dos bancos de dados relacionais, cada vendedor pode implementar SQL de formas ligeiramente diferentes.

Isto cria a necessidade de ferramentas para isolar os desenvolvedores e os usuários finais de diferenças em linguagens de manipulação de dados, tipos e layouts de dados, e de terem que especificar tabela de endereço de banco de dados.

Os "gateways" para banco de dados são ferramentas de desenvolvimento que traduzem comandos SQL e escondem endereços de banco de dados dos desenvolvedores e das aplicações que os utilizam, provendo uma camada de isolamento que protege os investimentos e facilita a migração. Estas são as principais ferramentas classificadas como "Gateways" para Banco de Dados:

Sybase OmniSQL Gateway, Oracle, SQL*Connect, Information Builder's EDA/SQL e Micro Decison Ware.

5. Conclusão

A avaliação e escolha de ferramentas é um trabalho árduo, principalmente no mercado cliente/servidor onde mais de 200 fornecedores competem com promessas de entregar soluções quase instantâneas. A abordagem da classificação proposta pela IBM é uma das metodologias que pode facilitar esta escolha. O resultado do estudo indica que embora não haja um estilo predominante de computação distribuída emergindo como padrão, há alguns modelos distintos de estilos de implementação que são consistentes através de diferentes empresas e áreas funcionais.

Poucas das ferramentas avaliadas apresentam uma real orientação a objetos. Entretanto, muitas provêm os benefícios primários desta orientação, através de objetos reusáveis e abordagem de programação direcionada aos eventos.

Cliente/Servidor é, primeiramente, um relacionamento entre programas processando em máquinas separadas. Isto requer uma infra-estrutura para fazer coisas com que os PCs isolados nunca tiveram que se preocupar. Por exemplo, interprocessos robustos de comunicações entre as redes precisam ser incluídos no projeto. Interfaces gráficas usando GUIs e OOUIs (Interfaces de usuário orientado a objetos) precisam ser exploradas para fazer as aplicações parecerem mais como objetos do mundo real ao invés de processos de programas de computador. As interfaces de usuários estão se tornando ambientes complexos, com ênfase responsiva nas tarefas humanas. Com avançados clientes OOUI um novo tipo de *superclientes* será induzido, transformando as estações clientes em "locais de trabalho" multimídia onde muitos diálogos paralelos são conduzidos com uma variedade de servidores (18).

A Internet é um fenômeno mundial que também está forçando mudanças rápidas e fundamentais no desenvolvimento das ferramentas cliente/servidor.

Muitas grandes empresas têm o mesmo problema de distribuição e heterogeneidade de informação que caracteriza a Internet. Muitas também têm um backbone Internet em operação. O sucesso da World Wide Web com o arquivo padrão HTML (HyperText Markup Language) e servidores FTTP (HypertText Transfer Protocol) na solução deste problema em escala global despertou as empresas produtoras de software, que já começam a disponibilizar, ou prometem em suas próximas versões a facilidade de conexão com a Internet e a infra-estrutura para produzir e receber informações sob demanda (6, 8 15).

A sobrecarga de informações é uma doença de nosso tempo. Tecnologias que forma criadas para resolver o problema parecem tê-lo tornado pior (6).

A tecnologia WWW oferece nova abordagem para a autoria, gerenciamento e publicação de informação distribuídas, sem a complexidade dos velhos sistemas centralizados. A informação passa a ser criada e gerenciada por aqueles que a produzem, sem ter que depender de programadores para criar entradas de dados e extração de relatórios. Com novos browsers um usuário recuperar informações de sistemas e fontes distribuídas na Intranet da empresa, usando uma interface simples e uniforme, sem ter que conhecer nada sobre os servidores que está acessando. Estas simples mudanças na tecnologia irão revolucionar as infra-estruturas das informações e mudar as organizações.

A idéia de um mundo unido, que antes era apenas um sonho de poetas, ficcionistas e visionários místicos (19), começa a tomar forma no mundo concreto.

Referências Bibliográficas

01 CANTER, S. SQL: putting up a good front. *PC Magazine*, v. 12, n. 19, p. 237 - 320, nov. 1993.

02 _____, *Centura overview* Gupta Corporation.
http://www.gupta.com/centweb/ct_ovrw.htm, fe. 1996.

03 CARDOSO, A. S. Front-ends permite acesso a diversas bases de dados. *PC Mundo*, Rio de Janeiro, v. 7, n. 75,
p. 19-24, out. 1991.

04 DAMASIO, E. Compartilhar é a saída. *Connections Magazine*, v. 12, n. 19, p. 20-23, nov. 1993.

05 EFFENDI, S. *Call to the nations*, Bahá'í World Center. 1977. KROL, E. *The whole internet users' guide & catalog*.
O'Reilly & Associates, Inc. 2. ed. 1994.

06 GORMAN, M. M. *Enterprise database in a client/server environment*. New York: J. Wiley 1994.

07 IBM CORPORATION, *Application development tool for distributed computing*. Prepared for IBM Corporation by
Hurwitz Consulting Group, Inc.
<http://www.csc.ibm.com/advisor/library/ap2476.html>, nov. 1995.

08 ORFALI, R. et al. ~~Essencial client/server survival guide~~. New York: J. Wiley 1994.

09 RICHARDSON, M. F.; NEEDHAM, R. M. The TRIPOS Filling Machine, a front end to a file server. ~~ACM Operating Systems Review~~, v. 17, n. 5, p. 120-128, 1983.

10 SCHULTE, R. ~~Server are taking over client/server~~. Gartner Group, Inc. Service:LVCS, Note Number: SPA-401-336, 16 oct. 1995.

11 SHIMBERG, D. Following a client/server database methodology. ~~DBMS:database & client/server solutions~~, San Mateo, v. 8, n.5, p. 48-58, 70-71, maio 1995.

12 SHIPLEY T. ~~IS organization model for client/server, part 1~~. Gartner Group, Inc. Service: LVSC, Note Number: SPA-970-345, 23 ict, 1995.

13 _____. ~~IS organization Model for client/server, part 2~~. Gartner Group, Service: LVSC, Note Number: SPA-970-346, 23 oct. 1995.

14 - _____, ~~Stretching IS organization to the break point~~. Gartner Group, Inc. Service: LVSC, Note Numer: SPA-970-344, 23 oct. 1995.

15. SMITH, D. ~~WWW recebtrakuzatuib nyths to have minor effect on C/S~~. Gartner Group, Inc. Service: SSA, Note Numer: SPA-400-016, 16 oct. 1995.

16 TAYLOR, A. G. Peering into the mainframe database. ~~UNIX World Magazine~~, McGraw-Hill, Supplement; Special Report: MIS Mets Open Systems, p. 55-59, nov. 1991.

17 TELLEEN, S. L. ~~Intranet Concept~~. Amdahl Corporation. <http://www.amdahl.com/doc/products/bsg/intra/concepts.html>, fev. 1996.

18 _____. ~~What is Server/Cliente Computing?~~ Unisys Corporation. <http://www.unisys.com/sofstor/aseries/infomgm/hodbc6.html>, fev. 996.

19 ZISMAN, M. ~~Lotus Notes Web Strategy~~. Lotus Corporation.

<http://www.lotus.com/mediadv/zispart2.htm>, dez.
1995.