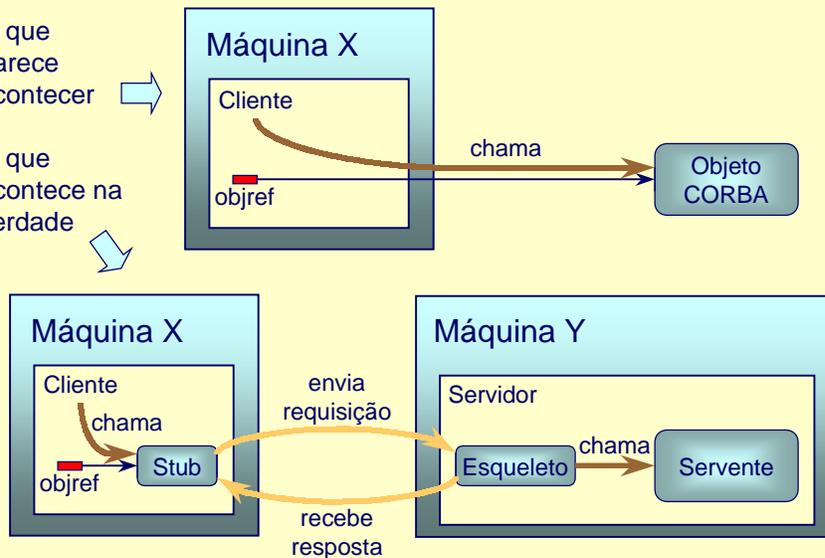


Repositórios de Implementações e *Binding*



Chamada Remota de Métodos

- O que parece acontecer
- O que acontece na verdade



GIOP/IIOP

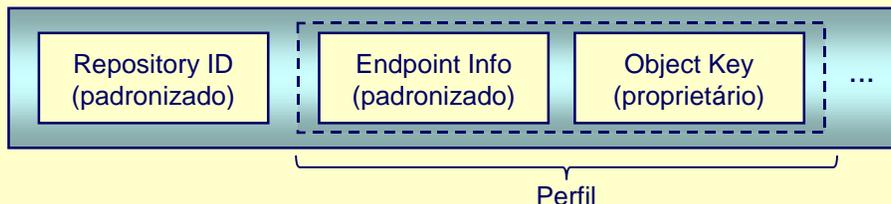
- CDR (Common Data Representation)
- Mensagens:
 - Request
 - Reply
 - pode ter status LOCATION_FORWARD
 - CancelRequest
 - LocateRequest
 - LocateReply
 - CloseConnection
 - MessageError
 - Fragment

CORBA Object References

- Podem ser transientes ou persistentes
- Referência transiente: referencia um objeto com tempo de vida limitado pelo tempo de vida do processo servidor no qual o objeto foi criado
- Referência persistente: referencia um objeto com tempo de vida independente do tempo de vida de qualquer processo servidor

Anatomia de uma IOR

CORBA Interoperable Object Reference (IOR):



- Um repository ID + uma seqüência de perfis
 - Repository ID: identifica o tipo mais derivado do objeto
 - É o ID de uma entrada do repositório de interfaces
 - Endpoint Info: host + port (no caso de um perfil IOP)
 - Object Key: nome do POA + object ID

Binding

- Considere um cliente que tem uma object reference
- O binding dessa referência é processo de:
 - estabelecimento de conexão com o servidor (caso ainda não haja uma conexão aberta)
 - associação da object reference a um servente
- Esse processo é transparente para o cliente
 - O run-time do ORB faz binding de referências automaticamente
 - O binding ocorre quando a referência é usada pela primeira vez

Binding Modes

- **Modo direto (direct binding)**
 - Suportado por todos os ORBs
 - É o modo usado para referências transientes
 - Pode também ser usado para referências persistentes (mas geralmente não é)
- **Modo indireto (indirect binding)**
 - Através do repositório de implementações, que desempenha o papel de location broker
 - É o modo geralmente usado para referências persistentes

Binding Modes (cont.)

- Quando um servidor cria uma referência, o POA embute dentro da referência a informação host + port
- **Direct binding:**
 - o host e o port são do próprio servidor
- **Indirect binding:**
 - o host e o port são de um location broker (repositório de implementações) responsável pelo servidor

Três Casos a Considerar:

- Direct binding de referências transientes
 - Direct binding de referências persistentes
 - Indirect binding de referências persistentes
- Haveria um quarto caso (indirect binding de referências transientes), mas veremos que ele não faz sentido na prática

Referências Transientes

- Para criar uma referência transiente, um servidor usa um POA com a política TRANSIENT
- A IOR é criada com o seguinte perfil:
 - Endpoint info: host e port do próprio servidor
 - Object key composto de dois elementos:
 - o nome (único no espaço e no tempo) do POA usado para criar a IOR
 - um object ID único dentro desse POA (ORBs geralmente usam um contador para produzir object IDs)

Direct Binding de Referências Transientes

- Antes de mandar a primeira requisição, o run-time do ORB no cliente tenta abrir uma conexão usando o host e port na IOR
- Quatro possibilidades:
 - O servidor que criou a referência continua rodando nesse host e escutando nesse port
 - Nenhum processo está nesse host e port
 - A execução do servidor original foi encerrada e agora há outro servidor nesse host e port
 - A execução do servidor original foi encerrada e depois recomeçada no mesmo host e port



Análise das Quatro Possibilidades

- O servidor que criou a referência continua rodando
 - O cliente manda a requisição: binding bem sucedido
- Nenhum processo está no host e port da IOR
 - O run-time do cliente lança a exceção TRANSIENT
- A execução do servidor original foi encerrada e agora há outro servidor no host e port da IOR
 - O cliente manda a requisição e recebe a exceção OBJECT_NOT_EXIST, pois no servidor não há um POA cujo nome seja igual ao que está na object key da IOR
- A execução do servidor original foi encerrada e depois recomeçada no mesmo host e port
 - Igual ao caso anterior



Referências Persistentes

- Para criar uma referência transiente, um servidor usa um POA com a política PERSISTENT
- Pressupostos:
 - As várias execuções do servidor, quando (re)criam um certo POA, devem usar o mesmo nome para esse POA
 - O nome do POA não é mais único no tempo
 - Quando criam IORs para um mesmo objeto, todas as execuções do servidor devem usar o mesmo object ID
 - Nada de contadores!
- Para direct binding, há um pressuposto adicional:
 - O servidor está sempre no mesmo host e port
 - Todas as execuções do servidor usam esse host e port

Direct Binding de Referências Persistentes

- Simples e eficiente: antes de mandar a primeira requisição, o run-time do ORB no cliente abre uma conexão usando o host e port na IOR
- Mas... problemas:
 - Servidores ficam presos a máquinas
 - A migração de servidores invalida as referências armazenadas por clientes
 - Ports precisam ser reservados
 - Dor de cabeça administrativa
 - O servidor precisa estar rodando quando o cliente mandar uma requisição.
 - Se não estiver, o binding falha e o run-time do cliente lança a exceção TRANSIENT

Indirect Binding

- Resolve esses problemas
- Custo: desempenho um pouco menor na primeira requisição de um cliente a um objeto
 - Só na primeira!
- Requer um repositório de implementações, que é uma parte opcional de CORBA
 - A partir de CORBA 2.2 o OMG tem se concentrado em especificar coisas que afetam a portabilidade de programas
 - Questões de qualidade de serviço são deliberadamente deixadas em aberto

Repositório de Implementações

- Responsabilidades:
 - Mantém um cadastro dos servidores conhecidos
 - Sabe que servidores estão rodando agora, e em que ports eles estão aguardando requisições
 - Põe para rodar os servidores que estiverem registrados
- Roda como um processo especial, num host e port fixados
- Mantém uma tabela de servidores

A Tabela de Servidores

Nome Lógico do Servidor	Nome do POA	Comando de Ativação	Host e Port
Fabrica	area1	rsh foo /opt/app/serv_fabr	foo.acme.com:1780
Fabrica	area2		foo.acme.com:1780
NameService	ns_poa	/opt/myorb/bin/name_server -v	xoo.acme.com:1889
DeptoPessoal	DP1		
Estoque1	E1		boo.acme.com:1270
Estoque2	E2		

- ORBs incluem ferramentas de administração que manipulam o repositório de implementações



Otimizações

- Uso de LocateRequest
- Evitar indirect bindings desnecessários
- Guardar na IOR o “endereço natal” do objeto
- Protocolos proprietários entre o servidor e o repositório de implementações
- Balanceamento de carga

