



Computação em cluster x computação em grade

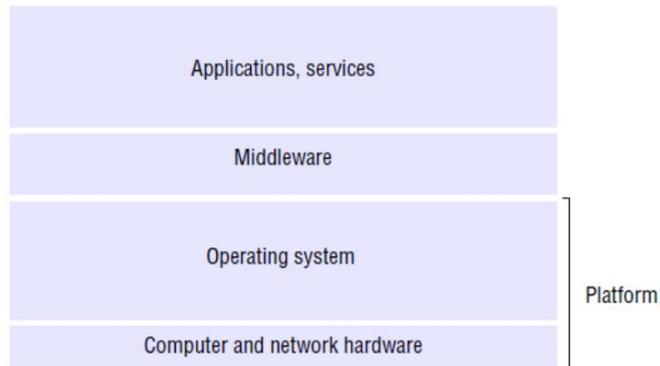
Clusters utilizam de várias **máquinas semelhantes**, com o mesmo sistema operacional rodando em uma **rede local**.

Computação em grade utiliza de várias **máquinas heterogêneas dispersas geograficamente**.

Modelos arquiteturais

- Arquitetura em camadas:
 - A camada inferior presta serviços a camada superior.
 - Modularidade.
 - O fluxo foi de camada a camada, de cima para baixo.
- Arquiteturas baseadas em objetos:
 - Arquitetura mais fracamente acoplada que a em camadas.
 - Cada objeto corresponde a um componente. Componentes são conectados por meio de uma RPC.
- Arquiteturas centradas em dados:
 - Processos se comunicam por meio de um repositório em comum.
 - Um repositório de dados, BD por exemplo, fica no centro dessa arquitetura.
- Arquiteturas baseadas em eventos:
 - Processos se comunicam por meio da propagação de eventos.
 - Associados a sistemas publish/subscribe.
 - Os processos são fracamente acoplados e referencialmente desacoplados.

Camadas de serviços de Software e Hardware em Sistemas Distribuídos



Middleware:

O middleware é uma camada intermediária de *software* utilizada para compatibilizar diferentes recursos, para possibilitar a heterogeneidade em sistemas distribuídos. Auxilia na comunicação entre processos em um SD.

IDL (Interface Definition Language):

Uma IDL é uma descrição em alto nível criada para compatibilizar linguagens de programação, recursos de programação. Digamos que uma IDL descreve o valor de retorno de uma função ou método, seu nome, seus parâmetros e suas devidas tipagens. O desenvolvimento do método ou função fica a cargo do programador. Boas especificações de IDL compatibilizam muitos recursos de programação.

Stub (stub cliente e stub servidor)

Stub é uma parte (fragmento) de algoritmo que provêm a abstração de uma chamada (local) de procedimento (ou método, no caso da orientação a objetos) fazendo a ligação deste com o mecanismo de comunicação. Stubs são como uma ligação para os objetos remotos, são partes do código que fazem a chamada remota, que possibilitam RPC e RMI no cliete e no servidor.

Fonte: Wikipédia.

IPC - Comunicação entre processos

É um conjunto de mecanismos de troca de informações entre processos ou múltiplas threads. Ao fornecer mecanismos de IPC o SO implementa “canais” de comunicação entre processos.

- **Message passing**

- A mensagem é encapsulada (semelhante a redes) e serializada para o envio. No receptor, ela é reconstruída.

- O processador IPC gera um código para realizar o **marshalling** da mensagem antes de ser enviada: Empacotamento dessa mensagem segundo alguma regra ou formato externo. Também gera código para realizar o **unmarshalling**: desempacotamento da mensagem no destino (receptor) para produzir um conjunto de dados equivalente e compatível com os utilizados no receptor.
 - Send bloqueante: Envio e só continuo a execução após receber uma resposta.
 - Send não bloqueante: Após a mensagem ir para o buffer de envio, continua a execução do processo emissor.
- **Comunicação via sockets**
 - Sockets realizam a comunicação lógica entre processos. Utilizam de endereços IP (identificação de destinatário) e número de portas (identificador de processo).
- **Representação externa de dados**

São métodos usados para que dois computadores troquem dados binários. Os valores são convertidos para um formato externo antes do envio e convertidos para forma local no receptor. **Ex.: Protocol buffers.**
- **Comunicação multicast**

Na comunicação multicast um processo envia uma mensagem para um grupo de processos de modo que a composição do grupo (integrantes) fique transparente para o emissor. Os serviços multicast consomem menos recursos de rede e são adequados para videoconferências.

É usado um IP multicast para atingir todos os componentes do grupo.

Remote invocation

- Request reply
 - Usada principalmente na arquitetura cliente-servidor, onde o cliente envia uma requisição e o servidor responde.
 - R: Request -> Apenas envia a requisição.
 - RR: Request reply: Envia a requisição e aguarda uma resposta.
 - RRA: Requesty-Reply-Acknowledge-Reply: Envia uma requisição,

recebe uma resposta e confirma o recebimento da resposta ao servidor.

- RPC - Chamada de procedimento remoto
 - Faz uma requisição por meio da chamada de um procedimento remoto.
 - Chama uma função em outro processo, podendo esse processo estar na mesma máquina ou não.
 - Dados devem ser passados por valor e não por referência. Um ponteiro na máquina cliente não é válido no servidor. Deve ser enviada uma cópia do valor como parâmetro da requisição, se necessário.
- RMI - Invocação de método remoto
 - É similar ao RPC, com a adição do suporte a objetos. Podem ser passados objetos como parâmetros e recebidos objetos como resposta.
 - Pode ser realizada uma chamada de método em determinado objeto. Por exemplo, o processo B tem 10 pessoas cadastradas. O processo A pode chamar o método `questoesProvaSD()` do objeto “João Flávio de Oliveira”, instância de pessoa no processo B.

Indirect communication

Tem como essência a comunicação através de um intermediário (broker), portanto, não existe nenhuma ligação direta entre o emissor e um ou mais receptores.

Há o **desacoplamento de espaço**: O remetente não sabe e nem precisa saber a identidade do emissor e vice-versa. Remetentes ou receptores, podem ser substituídos, atualizados, replicados ou migrados.

Há o **desacoplamento de tempo**: O remetente e o receptor não necessariamente precisam estar ativos no mesmo tempo para que a comunicação ocorra.

- Group communication
 - A mensagem é enviada para um grupo e repassada para todos os integrantes desse grupo.
 - Entrega ordenada das mensagens. O grupo é bem definido e

gerenciado. É tolerante a falhas. Garante a entrega das mensagens.

- Sistemas baseados na distribuição de eventos
 - Publish-Subscribe;
 - Publishers publicam um evento através da operação `publish(e)`.
 - Subscribers manifestam interesse em um conjunto de eventos através de inscrições. Há um filtro que determina que eventos são relevantes para o inscrito.
 - Os eventos são entregues aos inscritos de forma assíncrona.
 - O sistema funciona como um *broker* e direciona os eventos aos assinantes corretos.
 - Pode ser baseada em tópicos (mais comum), baseada em conteúdo, baseados no tipo dos objetos publicados, baseado em conceitos, baseado em localização...
- Espaço de tuplas
 - Comunicação indireta entre processos ao disponibilizar um espaço para compartilhamento que permite a leitura e escrita de tuplas. Essas tuplas são acessadas por padrão de correspondência de conteúdo.
 - **Write:** Processo insere uma nova tupla sem afetar tuplas existentes.
 - **Read:** Processo lê uma tupla sem afetar o conteúdo do espaço de tuplas.
 - **Take:** Remove uma tupla do espaço, para ser utilizada pelo processo que a removeu.

