Lista de Exercício I - Sistemas Distribuídos

Rafael Gonçalves de Oliveira Viana[1]
13 de novembro de 2017

[1]Sistemas de Informação – Universidade Federal do Mato Grosso do Sul (UFMS)
Caixa Postal 79400-000 – Coxim – MS – Brazil

rafael.viana@aluno.ufms.br

1. Quais são as propriedades que identificam os Sistemas Distribuídos. Quais são as vantagens e desvantagens em relação a sistemas não distribuídos.

1. As propriedades de um sd são as seguintes:
   (a) Acesso a recursos
      i. Recursos antes monopolizados podem ser compartilhados entre usuários.
   (b) Transparência da distribuição
      i. Capacidade de se anunciar na rede, sendo ocultando ou não sua existência.
   (c) Abertura
      i. Divulgam os servios na rede por IDL
      ii. Interoperável
      iii. Portável
      iv. Extensível: facilidade em configurar e evoluir.
   (d) Escalabilidade
      i. Facil integração com outros sistemas.
      ii. facil evolução do sistema.
2. Vantagens VS Desvantagens
   (a) Vantagens
      i. Acesso a recurso como poder computacional, informação ou recursos compartilhados como impressoras.
      ii. Disponobilidade por replicas.
      iii. Portátil,Escalável, Extensível.
      iv. Ocultar parcialmente a transparência.
   (b) Desvantagens
      i. Ocultar totalmente a transparência, nem sempre é uma boa ideia.
      ii. A relação entre desempenho e transparência esta diretamente ligadas.
      iii. Segurança fica menos eficênte
      iv. Nivel de complexidade alto.

2. Por que pode ser necessário projetar a arquitetura do sistema antes da fase de especificação?

A especificação está ligado a como executar o serviço através de interfaces, em determinado ambiente arquitetônico, mediante atender o serviço pretendido. Quando um projeto tem sua arquitetura definida antes da especificação, fica claro quais definições de interface serão tomadas pelo projeto, removendo algumas definições que a arquitetura pretendida não oferta, antes de chegar na etapa de especificação.

3. Escreva sobre cada uma das arquiteturas abaixo.

1. **Centrado em dados**:
   (a) Visa o acesso e atualização de repositorio .
   (b) Os componentes se cominicam apenas por meio do repositorio compartilhado.
   (c) Se comunicam por conectores como banco de dados ou arquivo diretório.
   (d) Podem ser passivos (repossitorios) ou
      i. tem a desvantagem no modelo de dados padronizado, evolução é dificial para outro modelo.
      ii. vantegem em grande repositorios de dados.
   (e) Ativos(surgiiu na IA);
      i. baseados em agentes, inteligentes
      ii. Podem organizar em sociedade complexas para realiza determinada tarefa1.
      iii. um exemplo seria o quadro negro.
      iv. quando um especialista tem uma poderação ele adiciona a mesma.
   (f) o modelo pode ser definido unicamente pelo reposiório.
   (g) tem a desvantagem de lidar com um modelo unico de sistema,dificil evolucao e custosa, dificil implementar algo eficiênte, politicas de gerenciamento mais dificil de ser criada.
2. **Cliente-servidor**:
   (a) É derivado do estilo em camadas, é separado entre uma maquina sendo o cliente e outra o servidor, essa separação pode ser fisica ou logicamente.
   (b) cont
   (c) O cliente requisita informações e o Servidor responde as requisições.
   (d) Contem as caracteriscas do modelo em cadamadas.
3. **Em camadas**:
   (a) O sistema de camadas é um exemplo de sistema de componentes que interagem, entre si hierarquicamente. Requisições descem as camadas e respostas sobem as camadas.
   (b) Cada componente 'L' comunicasse somente com L-1.
   (c) unica desvantagem é atravessar todas camadas par fazer o processamento
   (d) Baixo acoplamento entre os componentes.
4. **Call-return**:
   Um exemplo de conector.
5. **Pipes and filters**:
6. O sistema é visto como uma série de transformações complementares
7. Possui o objetivo de maximizar a reutilização de funcionalidades pequenas e muito bem definidas;
8.
9. **Baseados em eventos**:
   (a) Componentes: Unidades de processamento de dados que produzem e tratam eventos que acontecem no sistema
   (b) Conectores: Elementos que disseminam os eventos entre os componentes do sistema. Não fazem ligação entre componentes ponto-a-ponto.
   (c) Restrições

        i. A comunicação entre os componentes deve acontecer exclusivamente através dos eventos do sistema
        ii. Os conectores apenas comunicam eventos, não executam transformação de dados
    (d) É possível um componente gerar um evento para todos os componentes, ou para alguns.
    (e) Componente publique um evento em um middleware de controle. Somente os componentes que estiverem subscritos no middleware poderão receber a mensagem.
    (f) Permite combinar outros estilos arquitetônicos no middleware.
10. **Orientado a agentes**:
11. **Máquina virtual**:
    (a) Estilo arquitetural também conhecido como máquina abstrata.
    (b) Este estilo arquitetural é indicado quando é necessário executar atividades intermediárias, antes da execução da funcionalidade propriamente dita
    (c) Maquina como java container.
    (d) desaclopamento

4. Analisando os enunciados de sistemas de software apresentados a seguir, recomende um estilo arquitetural para cada um deles, justificando a sua opção:

1. Funcionarios processamento de salario
   Batch
2. Sistema de comunicação tipo walkie-talkie para smartphones.
   per-to-perto ou pipe
3. Uma loja virtual no ramo de calçados. Em que usuários possam efetuar compras a partir de dispositivos móveis e computadores, de acesso à Internet. **cliente servidor**
4. Um sistema de vídeoconferência entre vários usuários localizados em locais distintos. Esse sistema deve permitir a exibição de áudio e vídeo (em "tempo real") para vários participantes ao mesmo tempo.
   Per-to-Per ou MCU
5.

5. Os requisitos abaixo correspondem a um sistema distribuído a ser implementado. Descreva uma solução arquitetônica que contemple tais requisitos. Justifique sua resposta de acordo com os requisitos de qualidade. Observação: a solução pode agregar um ou mais estilos arquitetônicos.

1. É um sistema WEB;
2. O acesso deve ser feito de maneira segura;
3. A interface gráfica deve ser desacoplada do restante da aplicação;
4. A interface gráfica deve ser facilmente personalizável;
5. O sistema deve implementar persistência de dados;
6. As regras do negócio devem ser facilmente evoluídas;
7. O sistema deve ser escalável, de acordo com o número de acessos simultâneos.

   Cliente Servidor