

Implantação automatizada de composições de serviços web de grande escala

Leonardo Leite

IME - USP

26 de maio de 2014

Orientador: Marco Aurélio Gerosa

Coorientador: Fabio Kon

Conteúdo

- 1 Introdução
- 2 Definição da pesquisa
- 3 Trabalhos relacionados
- 4 O CHOReOS Enactment Engine
- 5 Avaliação
- 6 Conclusões

Implantação automatizada de composições de serviços web de grande escala

Implantação automatizada de composições de serviços web de grande escala

Implantação automatizada de composições de serviços web **de grande escala**

Implantação automatizada de composições de serviços web de grande escala

O processo de implantação manual se torna moroso e propenso a erros, principalmente na implantação de sistemas distribuídos.

Processo de implantação automatizada

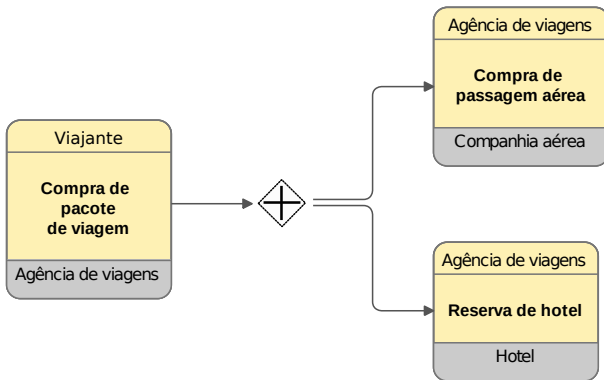
- Reprodutível
- Confiável
- Fácil de ser executado
- Entrega contínua
- DevOps

Processo de implantação automatizada

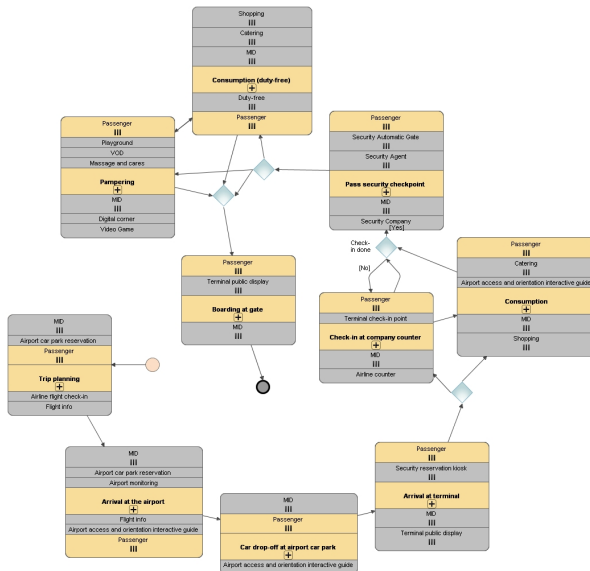
- Abordagens *ad-hoc* vs baseadas em middleware

Composições de serviços web

- Interfaces acessíveis por máquinas
- Automação de fluxos de negócios
- Aeroporto do Futuro

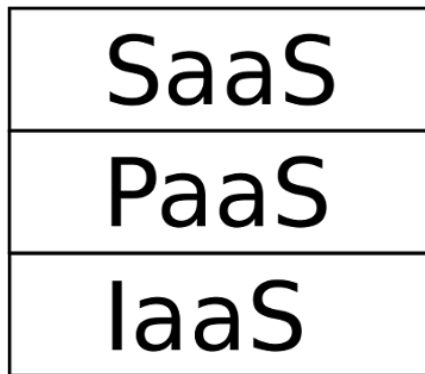


Composições de serviços web



Desafios na implantação em grande-escala

- Processo
- Falhas
- Disponibilidade
- Escalabilidade
- Heterogeneidade
- Múltiplas organizações
- Adaptabilidade



Implicações na implantação

- Não se conhece os IPs antes da implantação
- Servidores são efêmeros

Definição da pesquisa

Composições de serviços web de grande escala.

O quanto e como soluções de implantação baseadas em middleware trazem benefícios no dado contexto quando confrontadas com soluções *ad-hoc*?

Projetar, implementar e avaliar um middleware que forneça suporte à implantação automatizada de composições de serviços web de grande escala.

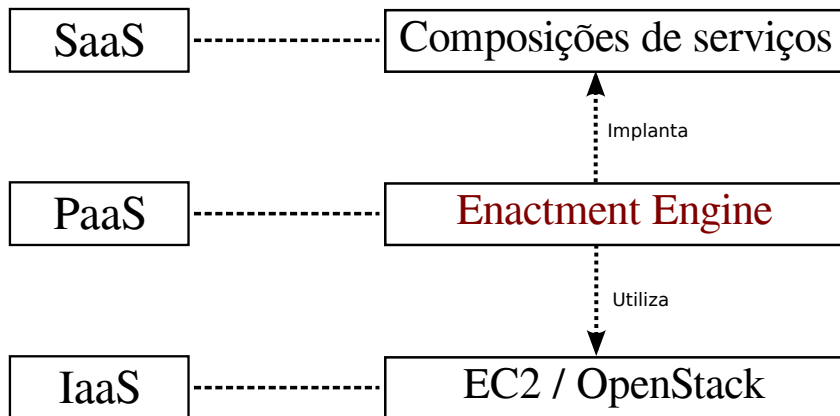
Trabalhos relacionados

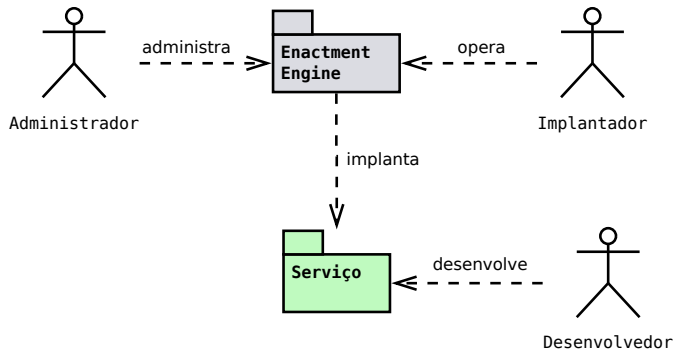
Trabalhos relacionados

<i>Trabalho</i>	<i>ADL</i>	<i>Escala</i>	<i>Composições</i>	<i>Nuvem</i>	<i>Heterog.</i>
Chef	x	-	-	-	-
Capistrano	x	-	-	-	-
Nix	x	x	✓	x	-
Darwin/Regis	✓	x	✓	x	x
Olan	✓	x	✓	x	x
Quema et al.	✓	✓	✓	x	x
J2EE app deployment	✓	x	✓	x	x
Globus Toolkit	✓	x	✓	x	x
Dynasoar	-	x	x	x	?
Open Knowledge	✓	x	✓	x	x
TOSCA	✓	x	✓	✓	✓
Juju	-	x	x	✓	✓
Cloud Foundry	-	?	x	✓	✓
Enactment Engine	✓	✓	✓	✓	✓

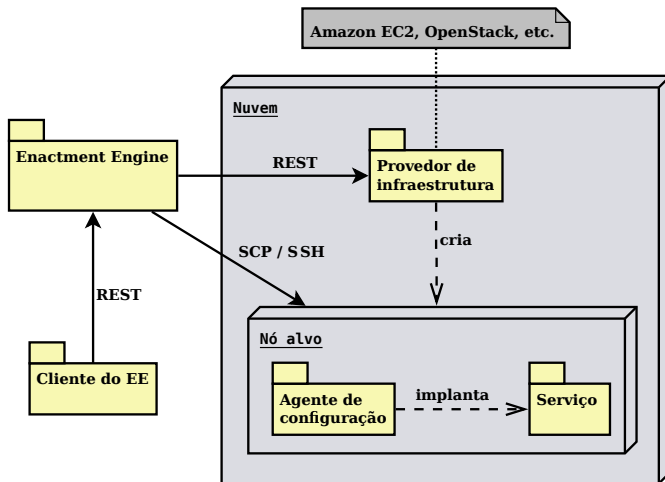
O CHOReOS Enactment Engine

O EE e os modelos de computação nuvem

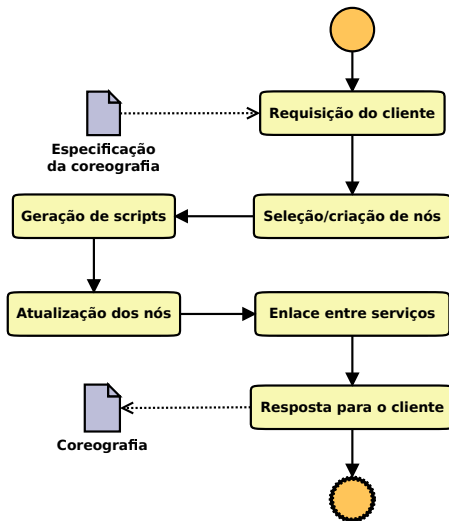




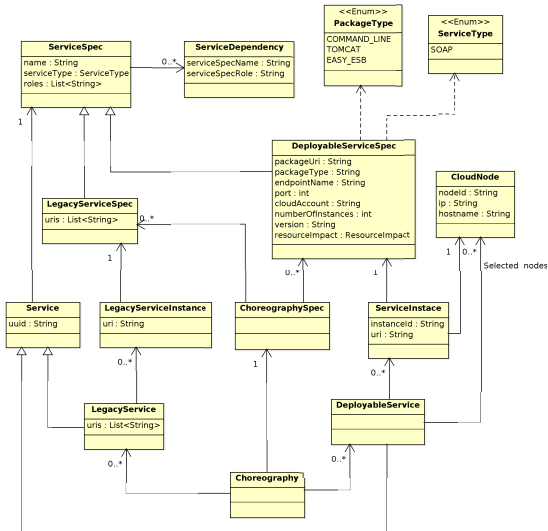
Ambiente de execução do EE



Processo de implantação implementado pelo EE



Estrutura da descrição arquitetural de uma coreografia



Exemplo de descrição arquitetural de uma coreografia

```
airportBusCompanySpec =  
    new DeployableServiceSpec(AIRPORT_BUS_COMPANY_NAME,  
        ServiceType.SOAP,  
        PackageType.COMMAND_LINE,  
        resourceImpact ,  
        serviceVersion ,  
        AIRPORT_BUS_COMPANY_JAR_URL ,  
        AIRPORT_BUS_COMPANY_PORT ,  
        AIRPORT_BUS_COMPANY_ENDPOINT ,  
        numberOfReplicas);  
  
airportBusCompanySpec.setRoles(  
    Collections.singletonList(AIRPORT_BUS_COMPANY_ROLE));  
  
airportBusCompanySpec.addDependency(  
    new ServiceDependency(AIRPORT_NAME, AIRPORT_ROLE));
```

Interface remota

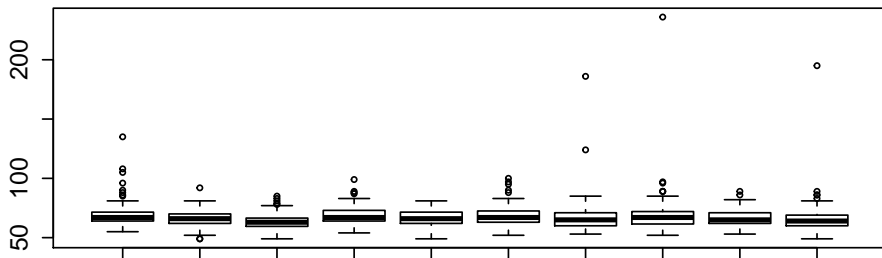
```
public interface EnactmentEngine {  
    public String createChoreography(ChoreographySpec chor);  
    public Choreography getChoreography(String chorId) throws  
        ChoreographyNotFoundException;  
    public Choreography deployChoreography(String chorId) throws  
        DeploymentException, ChoreographyNotFoundException;  
    public void updateChoreography(String chorId, ChoreographySpec  
        spec) throws DeploymentException,  
        ChoreographyNotFoundException;  
}
```

- Provedores de infraestrutura
- Políticas de seleção de nós
- Tipos de pacotes
- Tipos de serviços

Tratando falhas de terceiros

Invoker
<ul style="list-style-type: none">-task: Callable<T>-trialTimeout-trials: int-pauseBetweenTrials: int-timeUnit: TimeUnit
<ul style="list-style-type: none">+invoke(): T

Reserva de nós ociosos



O EE e os desafios de implantação em grande escala

Como abordagens de implantação baseadas em middleware auxiliam o implantador em relação aos desafios listados?

- Processo
- Falhas
- Disponibilidade
- Escalabilidade
- Heterogeneidade
- Múltiplas organizações
- Adaptabilidade

Processo

- Automação
- Interface REST
- Descrição declarativa
- Infraestrutura virtualizada

Falhas

- Invoker
- Reservoir
- Degradação suave
- Idempotência

Disponibilidade

- Replicação
- Dados

Escalabilidade

- Concorrência
- Tratamento de falhas
- Evitar gargalos (Chef Server → Chef Solo)

Heterogeneidade

- Extensibilidade

Múltiplas organizações

- Serviços legados
- Implantação multi-nuvem
- Federação

Adaptabilidade

- Atualização das coreografias
- Migração de serviços
- Replicação de serviços
- Implantação de infraestrutura de monitoramento

Avaliação

Comparação EE vs *ad-hoc*

EE

- Desenvolvimento: 45 min.
- Execução: 4 min.
- Tamanho: 180 LoC Java.

ad-hoc

- Desenvolvimento: 9 horas.
- Execução: 60 min.
- Tamanho: 100 LoC Shell Script, 220 LoC Java, e 85 LoC Ruby.

Dificuldades da abordagem *ad-hoc*

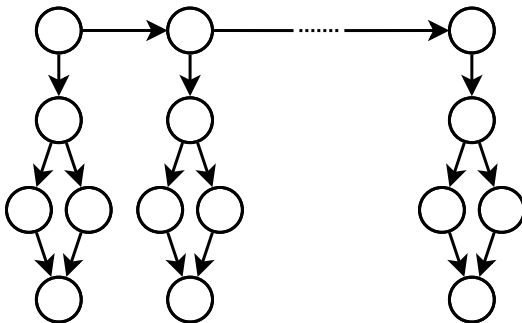
- Muitas tecnologias.
- Replicação de código.
- Passos manuais.
- Pouca paralelização.

Dificuldades da abordagem *ad-hoc*

Solução *ad-hoc* até poderia ficar melhor...
mas poderia ficar quase tão complexa quanto o próprio EE!

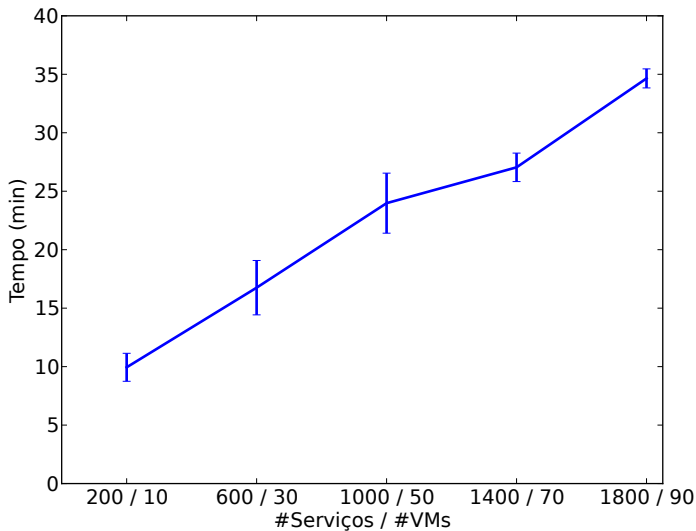
<i>Cenário</i>	<i>Composições</i>	<i>Tamanho</i>	<i>Nós</i>	<i>Serviços/Nós</i>
1	10	10	9	11 ou 12
2	10	100	90	11 ou 12
3	100	10	90	11 ou 12
4	10	10	5	20

Topografia da coreografia sintética utilizada nos experimentos

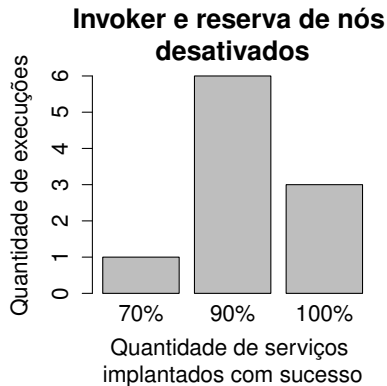
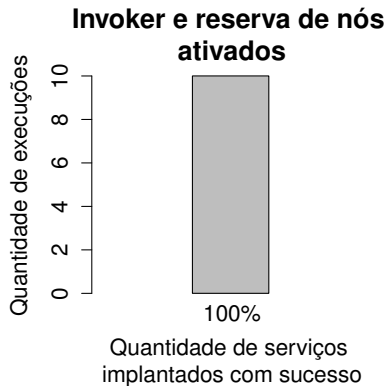


<i>Cenário</i>	<i>Tempo (s)</i>	<i>Composições com sucesso</i>	<i>Serviços com sucesso</i>
1	467.9 ± 34.8	10.0 ± 0	100.0 ± 0 (100%)
2	1477.1 ± 130.0	9.3 ± 0.3	999.3 ± 0.4 (99.9%)
3	1455.2 ± 159.1	98.9 ± 0.8	998.5 ± 1.3 (99.9%)
4	585.2 ± 38.1	10.0 ± 0.1	100.0 ± 0.1 (100%)

Análise de escalabilidade



Efetividade do tratamento de falhas



Cada execução: 1 coreografia de 100 serviços.

Conclusões

- A implementação de um middleware que possibilita a implantação automatizada de composições de serviços.
- Uma comparação, baseada na literatura e em evidências empíricas, entre soluções de implantação automatizada com abordagens *ad-hoc* e baseadas em middleware.

SBRC

Leonardo Leite, Nelson Lago, Marco Aurélio Gerosa e Fabio Kon. Um Middleware para Encenação Automatizada de Coreografias de Serviços Web em Ambientes de Computação em Nuvem. Em *31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2013.

MiniPlop Brasil

Leonardo Leite. Fábrica dinâmica de dublês: testando classes que possuem dependências não injetáveis. Em *Miniconferência Latino-Americana de Linguagens de Padrões para Programação*, 2013.

SOCA

Leonardo Leite, Gustavo Oliva, Guilherme Nogueira, Marco Aurélio Gerosa, Fabio Kon e Dejan Milojicic. A systematic literature review of service choreography adaptation. *Service Oriented Computing and Applications*, 3(7):201–218, 2013.

- Análise multivariável de fatores que influenciam a escalabilidade.
- Experimentos com desenvolvedores.
- Algoritmos adaptativos para tratamento de falhas.
- Federação de instâncias do EE.
- Utilização de um balanceador de carga.
- Utilização de um barramento de serviços.
- Atualização dinâmica de composições de serviços.

Obrigado!

Leonardo Alexandre Ferreira Leite

<http://www.ime.usp.br/~leofl>

leofl@ime.usp.br

CHOReOS Enactment Engine

<http://ccsl.ime.usp.br/EnactmentEngine>

https://github.com/choreos/enactment_engine