CLOUD CRAWLER

Ana Flávia Dias
Luiza Ávila
Pedro Achilles
Stefany Gaspar

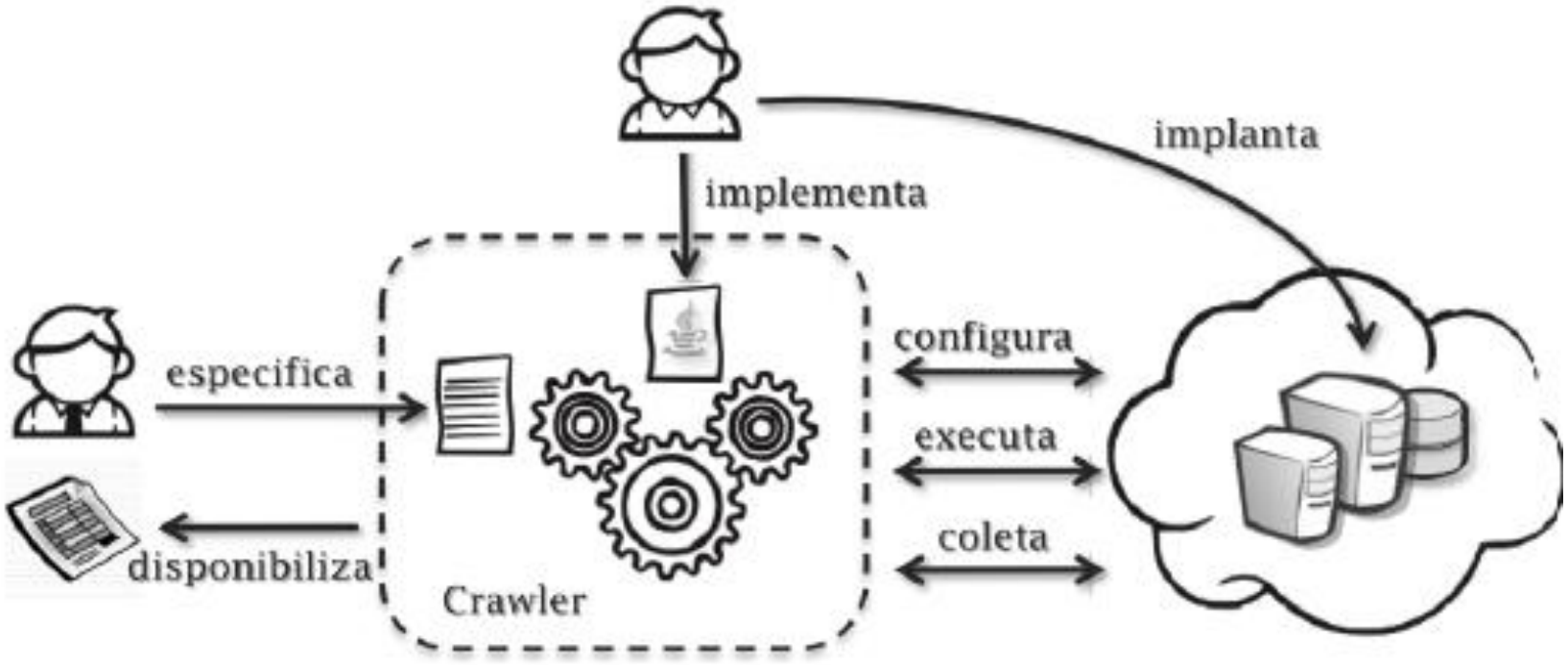
INTRODUÇÃO

- Crescente interesse por nuvens do tipo IaaS;
- O preço cobrado pelos recursos;
- Como identificar os provedores e os perfis de recursos virtuais com o melhor custo/benefício para uma determinada aplicação?;
- Principais contribuições do ambiente;
- Dois experimentos.

AMBIENTE CLOUD CRAWLER

Composto pela linguagem de descrição de cenários, Crawl, e o motor de execução dos cenários descritos em Crawl, de nome Crawler.

VISÃO GERAL

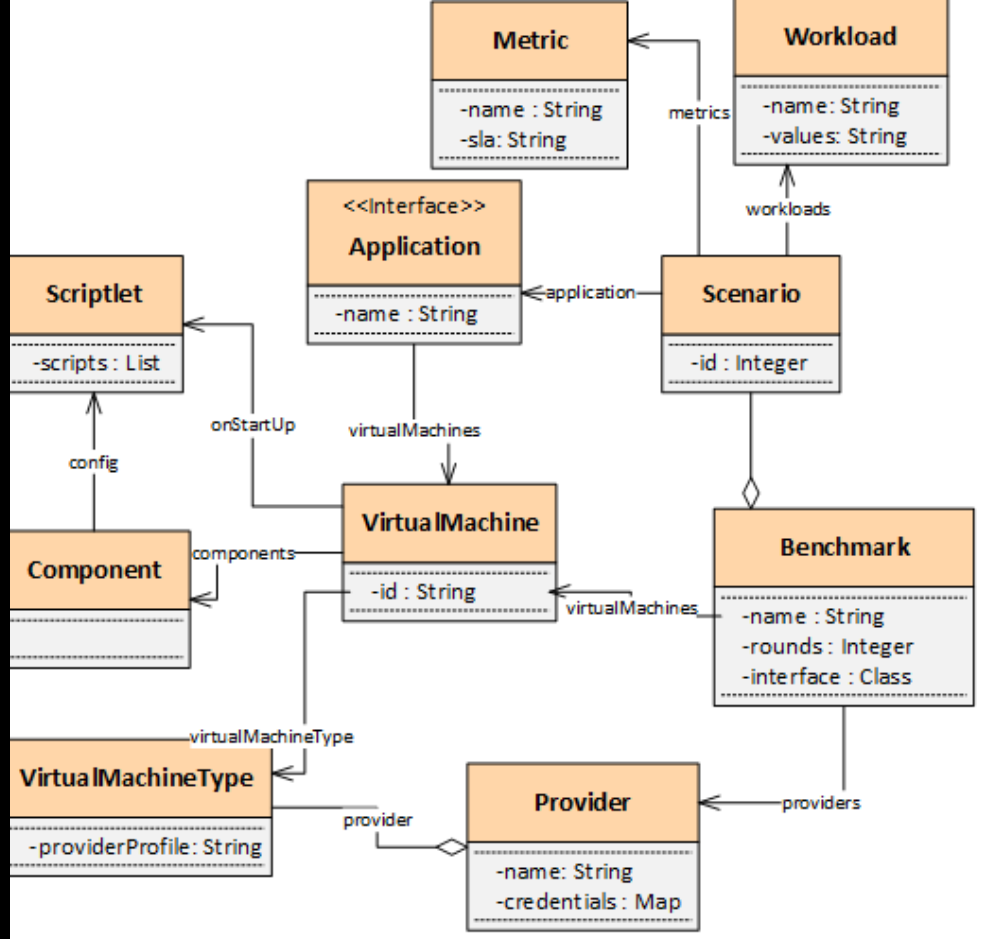

implanta
implementa
especifica
configura
executa
coleta
disponibiliza
Crawler

LINGUAGEM CRAWL

- Linguagem declarativa;

- Especificação de cenários de avaliação para esse tipo de infraestrutura complicada.
  - Configuração de certos componentes que só são conhecidos em tempo de execução.
  - Dinamicidade relacionada ao perfil dos recursos virtuais alocados à aplicação.

Usuário do ambiente pode especificar uma ampla variedade de cenários de avaliação de desempenho em nuvens IaaS.



10 ENTIDADES DO CRAWL

REPRESENTAÇÃO TEXTUAL

- Yaml foi escolhida como base para a representação textual da linguagem Crawl.

Símbolo	Função
:	Atribui o valor especificado no lado direito à chave declarada no lado esquerdo
–	Declara um elemento de uma sequência
!	Denota um tipo de entidade
&	Denota uma entidade referenciável
*	Referencia uma entidade previamente declarada

- Crawl oferece duas extensões à sintaxe original de Yaml.

EXEMPLO DA ESPECIFICAÇÃO DE UM CENÁRIO EM CRAWL

```
1  !benchmark
2  name: bench_1
3  rounds: 3
4  interval: 5000
5
6  # definição do provedor
7  providers:
8   -&rackspace !provider
9      credentialPath: Rackspace.txt
10     -virtualMachineTypes:
11        -&flavor1 !virtualMachineType
12           providerProfile: 1
13           provider: *rackspace
14        -&flavor2 !virtualMachineType
15           providerProfile: 2
16           provider: *rackspace
17
18 # definição das máquinas virtuais
19 virtualMachines:
20  - &web_vm !virtualMachine
21     id: 7HJ38K
22     type: *flavor1
23     components:
24      - &apache !component
25          id: 1
26          onStartup:
27           -&apacheConfig !scriptlet
28             id: 1
29             scripts:
30              db.sh ${db.IP}
31
32  - &database_vm !virtualMachine
33     id: YH838K
34     type: *flavor2
35     components:
36        - &postgres !component
37           id: 2
38
39 # definição do cenário
40 scenarios:
41   - !scenario
42     id: 1
43     workloads:
44      - &load !workload
45          id: 1
46          values: [100,200,300,400]
47     metrics:
48      - &responseTime !metric
49        sla: 90
50     web_app: !application
51       db_layer: *database_vm
52       web_layer: *web_vm
```

EXTENSÕES CRIADAS

```
1   # definição dos perfis do provedor Amazon EC2
2   !include
3     file: /lib/providers/ec2/vm_profiles.crawl
4
5   # definição dos perfis do provedor Rackspace
6   virtualMachineTypes:
7   - !foreach
8       list: [1, 2, 3, 4, 5, 6, 7, 8, 9]
9       count: 0
10      var: flavor
11      statement:
12    - &flavor$[flavor] !virtualMachineType
13        providerProfile: $[flavor]
14        provider: *rackspace
```

MOTOR CRAWLER

- Mecanismo encarregado de validar, executar e coletar os resultados dos cenários de avaliação descritos em Crawl;
- A arquitetura do Crawler é composta de quatro módulos: Engine, Parser, Builder e Executer;
- O Crawler reutiliza as máquinas virtuais previamente criadas pelo usuário, apenas alterando os seus perfis, caso necessário.

EXPERIMENTOS

Ilustrar o uso e as facilidades propiciadas pelo ambiente Cloud Crawler.

METODOLOGIA E ESPECIFICAÇÃO DOS CENÁRIOS

1 aplicação ⟶ 2 cenários, cada um com um perfil.

Olio ⟶ Amazon EC2 (*c1.xlarge*) e Rackstar (*flavor 5*)

Gerador de catga ⟶ Faban

Estrutura básica:

- Servidor web: Nginx
- Servidor de aplicação: Thin
- Banco de dados: MySQL

METODOLOGIA E ESPECIFICAÇÃO DOS CENÁRIOS

Cada cenário passa por duas demandas, sendo executado 3 vezes para cada valor de usuários concorrentes:

1. Demanda baixa: entre 25 e 150 usuários concorrentes
2. Demanda moderada: entre 200 e 800 usuários concorrentes

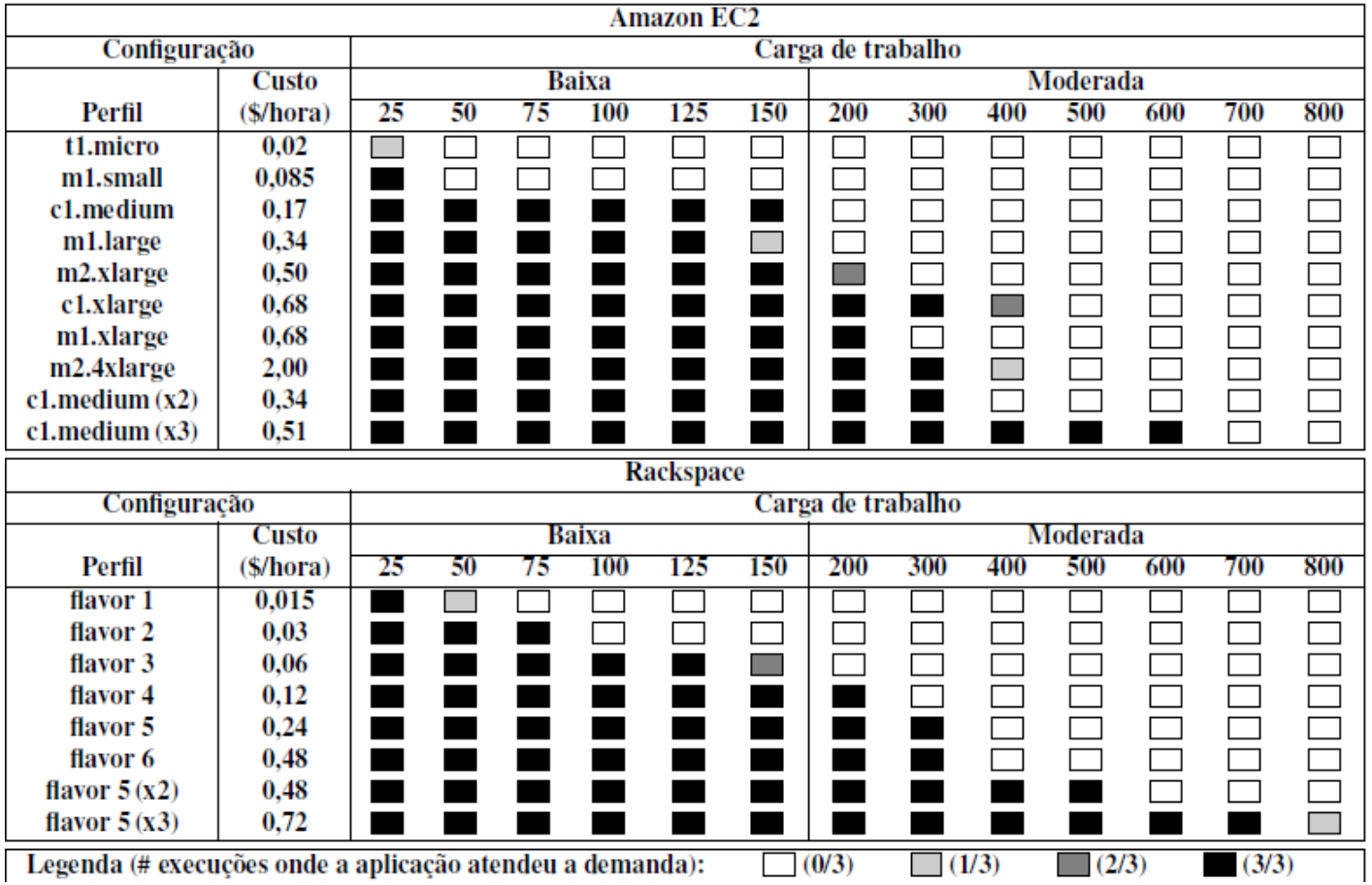
Métrica: tempo de resposta da aplicação

Valor mínimo aceito: 90% das respostas dentro do tempo limite.

ESPECIFICAÇÃO DOS CENÁRIOS DE AVALIAÇÃO DA OLIO NA AMAZON EC2.

```
!benchmark
name: OlioBenchmark
rounds: 3
- !include
  file: providers.crawl

properties:
 - &low_prof !virtualMachineType
   providerProfile: t1.micro
   provider: *ec2
   properties:
    - workers: 2
    - thinServers: 2
 ...
 - &high_prof !virtualMachineType
   providerProfile: c1.xlarge
   provider: *ec2
   properties:
    - workers: 6
    - thinServers: 18

virtualMachines:
 - &faban !virtualMachine
    id: 20408853
    type: *high_prof
    name: DRIVER
    components:
     - *fabanApp
 - &mysql !virtualMachine
    id: 20406221
    type: *high_prof
    name: MYSQL
    components:
     - *mysqlApp


workloads:
 - &low !workload
   values: [25,50,75,100,125,150]
 - &moderate !workload
   values: [200,300,400,500,600,
            700,800]

scenarios:
 # implantação vertical
 - !foreach
   list: [*low_prof, ..,
          *high_prof]
   var: profile
   statement:
    - !scenario
      workloads:
       - *low
       - *moderate
      metrics:
       - *responseTime
      application: !OlioApp
        database: *mysql
        driver: *faban
        web:
         - !virtualMachine
           id: 20400657
           name: THIN SERVER
           type: $[profile]
           components:
            - *railsApp
            - *tomcatApp
            - *nginxApp
            - !component
              config:
               - &db_script !scriptlet
                 scripts:
                  - sudo
                    /home/db.sh ${IP}

 # implantação horizontal
 - !scenario
   workloads:
    - *low
    - *moderate
   metrics:
    - *responseTime
   application: !OlioApp
     database: *mysql
     driver: *faban
     web:
      - !foreach
        list: [239999, ..,
               9989989]
        var: machine_id
        statement:
         - !virtualMachine
           id: $[machine_id]
           name: THIN SERVER
           type: *high_profile
           components:
            - *railsApp
            - *tomcatApp
            - *nginxApp
            - !component
              config:
               - *db_script
```

DESEMPENHO DA APLICAÇÃO OLIO NAS NUVENS AMAZON EC2 E RACKSPACE.



Amazon EC2														
Configuração		Carga de trabalho												
		Baixa						Moderada						
Perfil	Custo ($/hora)	25	50	75	100	125	150	200	300	400	500	600	700	800
t1.micro	0,02	1/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
m1.small	0,085	3/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
c1.medium	0,17	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
m1.large	0,34	3/3	3/3	3/3	3/3	3/3	1/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
m2.xlarge	0,50	3/3	3/3	3/3	3/3	3/3	3/3	2/3	0/3	0/3	0/3	0/3	0/3	0/3
c1.xlarge	0,68	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	2/3	0/3	0/3	0/3	0/3
m1.xlarge	0,68	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3	0/3
m2.4xlarge	2,00	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	1/3	0/3	0/3	0/3	0/3
c1.medium (x2)	0,34	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3
c1.medium (x3)	0,51	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3

Rackspace														
Configuração		Carga de trabalho												
		Baixa						Moderada						
Perfil	Custo ($/hora)	25	50	75	100	125	150	200	300	400	500	600	700	800
flavor 1	0,015	3/3	1/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
flavor 2	0,03	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
flavor 3	0,06	3/3	3/3	3/3	3/3	3/3	2/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
flavor 4	0,12	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3	0/3
flavor 5	0,24	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3
flavor 6	0,48	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3
flavor 5 (x2)	0,48	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3
flavor 5 (x3)	0,72	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	1/3

Legenda (# execuções onde a aplicação atendeu a demanda): (0/3) (1/3) (2/3) (3/3)

RESULTADOS

- Escalabilidade Horizontal.
- Provedor Rackspace.
  - Configuração: três máquinas virtuais de perfil c1.medium
  - Demanda: 600 usuários
- Provedor Amazon EC2.
  - Configuração: tres máquinas virtuais de perfil flavor 5
  - Demanda: 800 usuários

CONCLUSÃO

- O ambiente constitui-se em uma importante ferramenta de planejamento e alocação de recursos em nuvens IaaS;

- Atualmente, há uma série de linhas de pesquisa sendo consideradas visando melhorar e estender as funcionalidade do ambiente.

REFERÊNCIA DO ARTIGO

Cunha, Matheus Ciríaco Cerqueira. *Um Ambiente Programável Para Avaliar O Desempenho De Aplicações Em Nuvens De Infraestrutura*. 2012.