

Gestão do Autoscaling Group

Grace Period

x min

É cerca do dobro do tempo que o sistema leva a ficar operacional. Assim é um tempo razoável, pois não detecta falsos positivos frequentemente. Por outro lado, no pior dos casos uma máquina que esteja de facto a falhar é detectada um pouco mais tarde, o que é raro.

É adicionada mais uma máquina quando

A utilização do CPU do sistema cresce acima dos x%. Isto acontece pois pode estar eminente um pico de pedidos e é necessário ter tempo para ligar novas instâncias antes da carga máxima ser atingida.

É removida uma instância quando

A Utilização do CPU desce abaixo dos 40%, pois existem recursos subaproveitados que podem ser removidos.

TODO HEALTH CHECK

Métricas

É recolhida a contagem total do instruções gastas na factorização de um dado pedido. Desta forma em pedidos subsequentes é possível estimar o seu custo se for um pedido já conhecido.

Para pedidos desconhecidos a decisão será feita com base no tamanho do input, bem como algumas heurísticas a experimentar.

Para além disto pretendemos recolher no load balancer o número de pedidos a serem efectuados em cada máquina, para evitar uma sobrecarga de threads.

Script

Compilar o ficheiro *MyICount.java* com Java 4 e com a biblioteca BIT no classpath

```
java [-cp ] -source 1.4 MyIcount.java
```

Compilar os ficheiros *WebServer.java* e *IntFactorization.java* com java 7

```
java WebServer.java IntFactorization.java
```

Instrumentar a class *IntFactorization.class* executando MyICount. Por omissão faz override da class instrumentada, na mesma pasta.

```
java -XX:-UseSplitVerifier MyIcount []
```

Criar o ficheiro *log.txt* na directoria do servidor

```
touch log.txt
```

Correr o *WebServer*

```
java -XX:-UseSplitVerifier WebServer
```