

FlexLoadGenerator: Um Framework para Auxiliar o Desenvolvimento de Geradores de Carga de Trabalho

Débora Souza
dsls@cin.ufpe.br

Orientador: Paulo Maciel
prmm@cin.ufpe.br



Roteiro

- Introdução
- Objetivo
- FlexLoadGenerator
- Projeto concluído
 - Alguns resultados
- Projeto em andamento.

Introdução

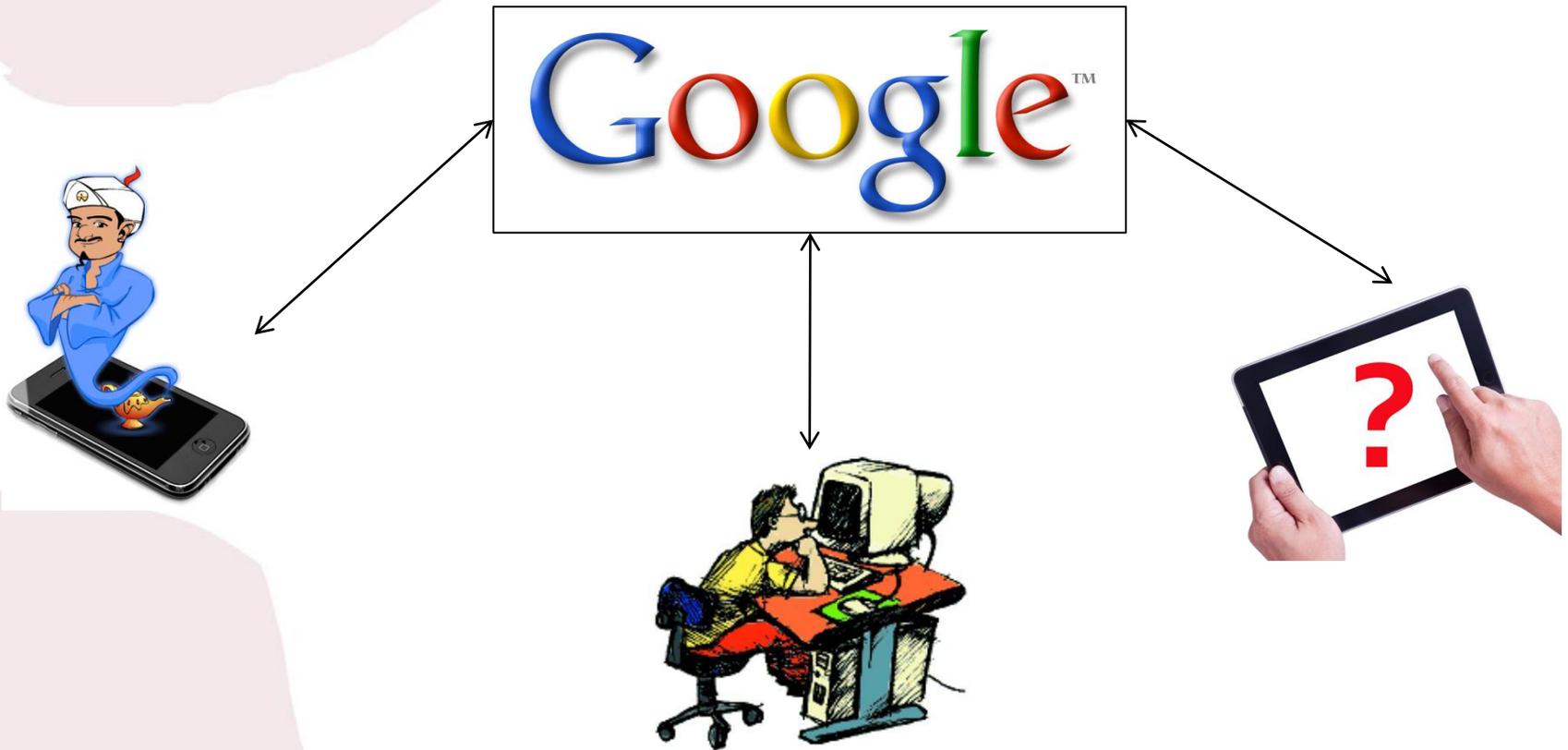


O que pode ser chamado de “carga”?

Segundo o dicionário Michaelis, carga significa: Ato de carregar; fardo, peso; porção, grande quantidade; etc.



Introdução



Caracterização de Carga

Introdução



Por que gerar carga de trabalho?

- Avaliar o desempenho de sistemas.
- Auxiliar no planejamento de capacidade.

Quanto tempo se leva para desenvolver um gerador de carga???



Objetivo



Desenvolver um framework flexível que englobe funcionalidades básicas para o desenvolvimento de ferramentas de geração de carga de trabalho.

FlexLoadGenerator



Atualmente composto por nove classes e uma biblioteca de geração de números aleatórios.

- Algumas de suas funcionalidades são:
 - Estabelecer conexão com sistemas alvo através dos protocolos TCP, UDP ou SSH2.
 - Faz tanto o envio como o recebimento de mensagens.
 - O usuário pode escolher como “criar” a carga.

FlexLoadGenerator

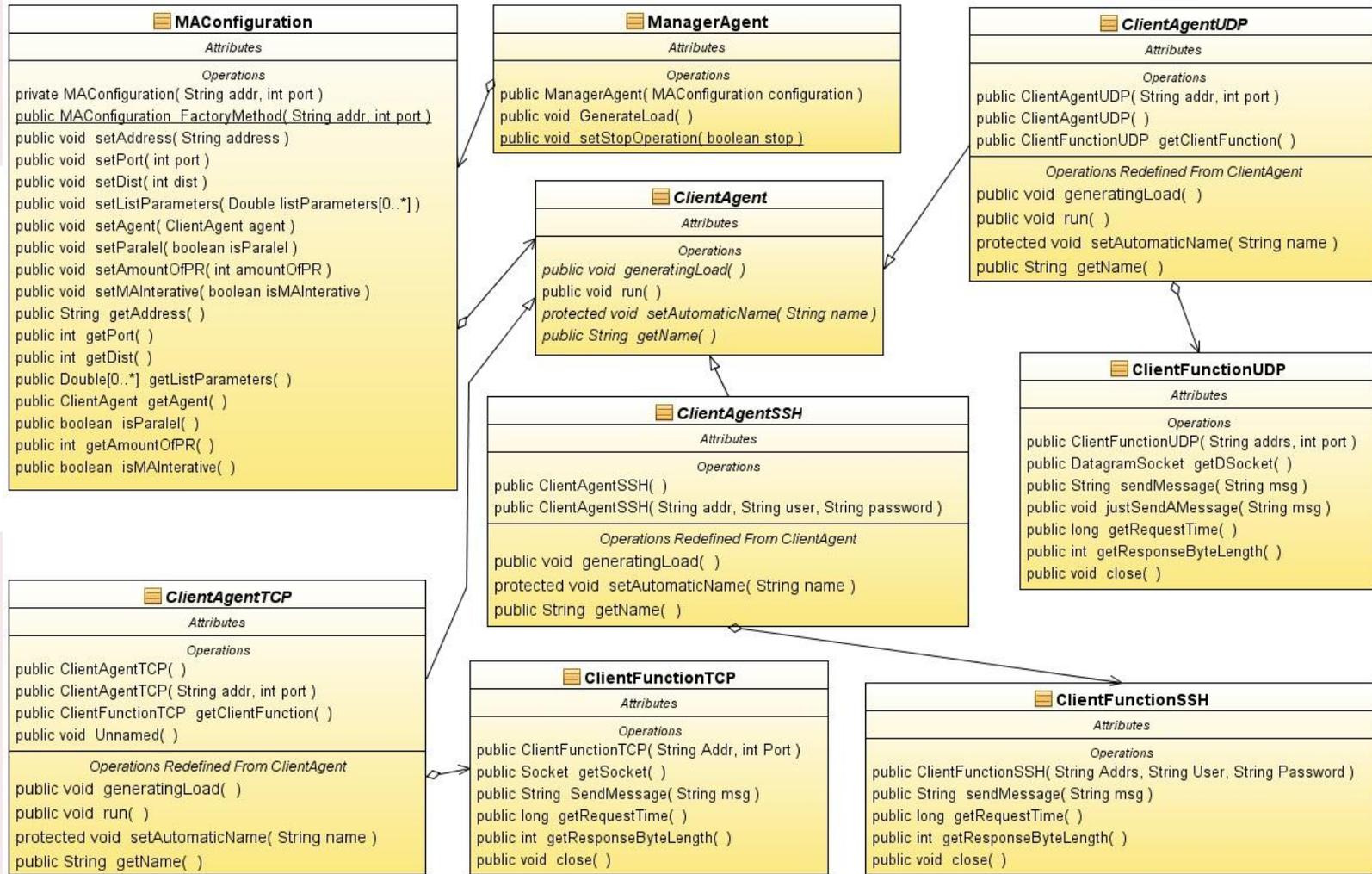


Diagrama de classe FlexLoadGenerator

FlexLoadGenerator



A ideia é que o desenvolvedor descreva o funcionamento da unidade geração de carga, o FlexLoadGenerator se encarregará da criação de agentes que representarão as unidades de geração de carga.

- Antes de criar um gerador, o desenvolvedor deve escolher um protocolo de comunicação (TCP, UDP ou SSH).
- Após a escolha ele deverá criar uma classe agente que herde de um dos agentes especificados no diagrama de classes anterior.
- Após a herança, o desenvolvedor deverá sobrescrever o método *generatingLoad()*, e neste método deverá desenvolver o código responsável pela geração de carga.

FlexLoadGenerator



O gerenciamento da geração de carga é feito pela classe *ManagerAgent* e pode ocorrer de forma interativa e paralela.

- Geração interativa: O *ManagerAgent* criará de tempos em tempo novos agentes.
- Geração paralela: O *ManagerAgent* criará um conjunto de agentes de forma paralela.



FlexLoadGenerator



O FlexLoadGenerator fornece quatro opções para gerenciamento de carga:

- Interativo e paralelo: Cria agentes indefinidamente e não espera a “morte” destes para a criação de outros.
- Interativo e não paralelo: Cria o agente quando o que estiver em execução “morrer”.
- Não interativo e paralelo: Cria vários agentes, mas é necessário criar loops dentro destes para que se possa manter a execução.
- Não interativo e não paralelo: Cria um agente e após a “morte” deste nenhum outro será criado.

FlexLoadGenerator

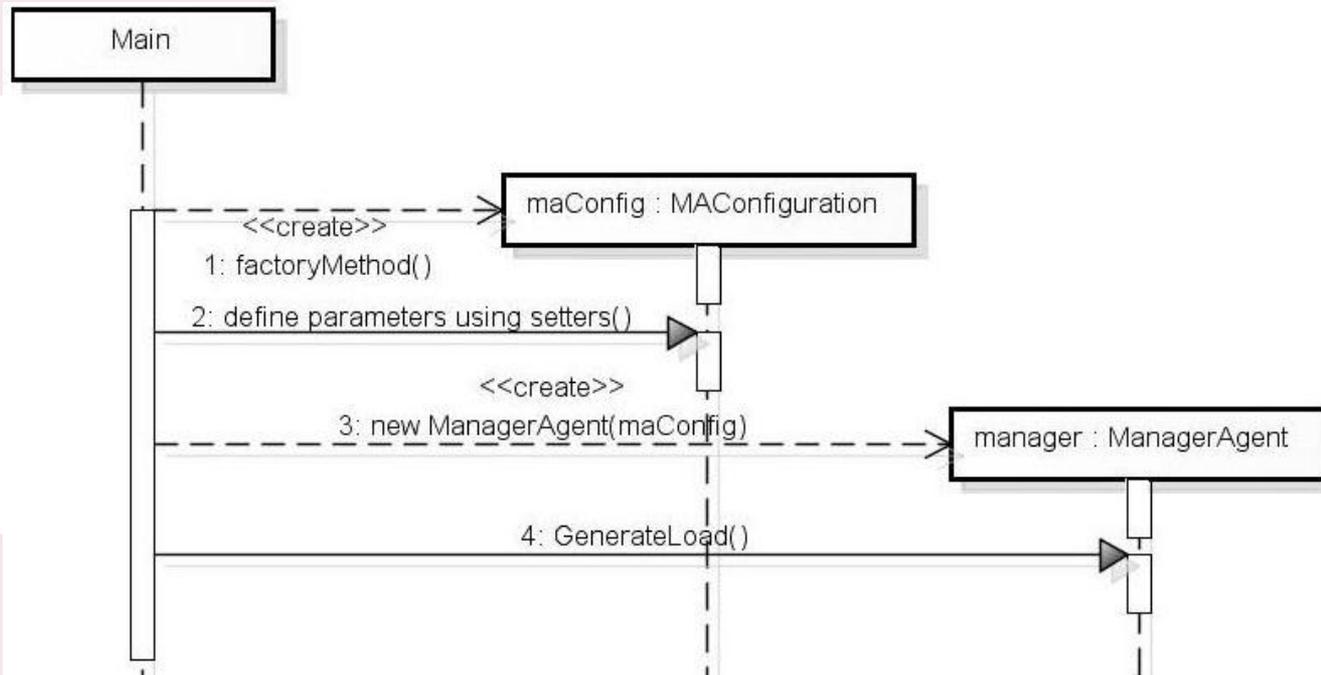


Diagrama de sequência FlexLoadGenerator

Projeto concluído



Projeto Concluído: WGSysEFT

Breve descrição: Gerador de carga de trabalho que opera realizando transações de transferência eletrônica de fundos (crédito e débito).

Alguns resultados



WGSysEFT – Descrição do cenário:

- Intel Core i5 650 3.20GHz.
- RAM Memory 4GB DDR3.
- HD (Hard Disk) 1TB Sata II.
- Windows Server 2003 Enterprise Edition Operating System.

Descrição dos experimentos:

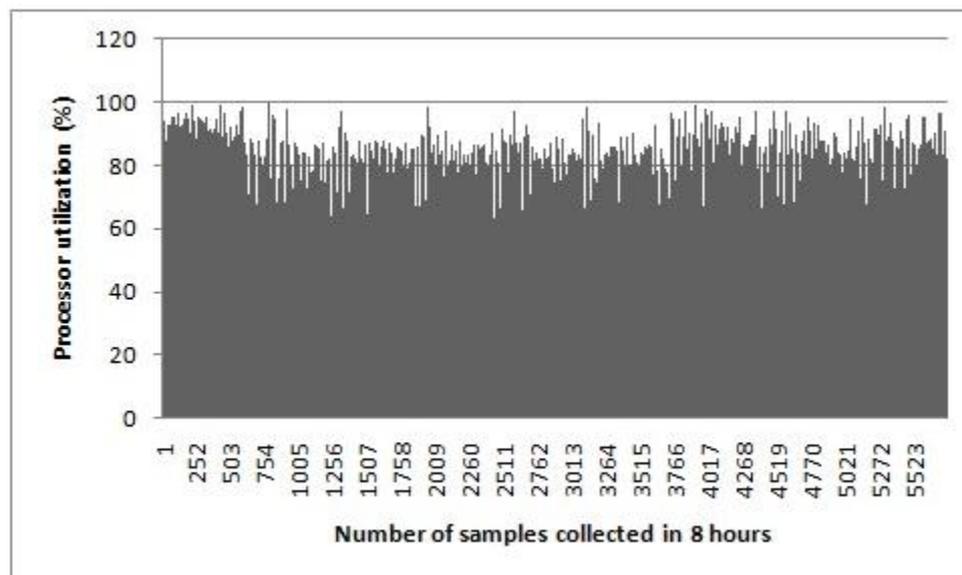
- 350 POS (point of sale).
- 8 horas de duração.
- Transações de crédito e débito.
- Percentual crédito a vista (43%); percentual crédito parcelado (23%); percentual débito (33%).
- Distribuição Exponencial.
- Tempo de 1000ms.

Alguns resultados

Resource	Average utilization (%)
Processor time	68.30
Disk Time	52.39
Disk writing time	52.29
Disk reading time	0.10

Uso de alguns dos recursos do servidor SCOPE

Uso de processamento servidor SCOPE



Projeto em andamento



Projeto em andamento: Gerador de falhas e reparos.

Breve descrição: Gerador de falhas e reparos que realiza envio de comandos para criar falhas e reparos tanto para hardware como para software.

- Ainda em fase de testes.
- Aplicabilidade para ambientes de cloud computing.

