

# FlexLoadGenerator – Um Framework para Apoiar o Desenvolvimento de Ferramentas Voltadas a Estudos de Avaliação de Desempenho e Dependabilidade

Débora Souza  
dsls@cin.ufpe.br

**Orientador:** Paulo Maciel  
prmm@cin.ufpe.br

---

# Roteiro

---

- Introdução
- Objetivo
- FlexLoadGenerator
- Ferramental desenvolvido
  - WGSysETF
  - EucaBomber
- Alguns resultados

# Introdução

---



Diversidade de sistemas computacionais esta em fase de desenvolvimento ou já em produção.

Assegurar que os serviços providos confiáveis de alto desempenho.

Estudos de avaliação de desempenho e dependabilidade podem contribuir para:

- Verificar a melhor alternativa de design de projeto:
- Se melhorias no sistema são recomendáveis;
- Comportamento do sistema mediante falhas.

# Introdução

---



Dificuldade de encontrar programas sintéticos que possa ser aplicado em experimentos de:

- Desempenho – Geração de eventos de carga simulado o usuários;
- Dependabilidade – Eventos de falhas e reparos.

O desenvolvimento de programas sintéticos se mostra muitas vezes custoso devido a necessidade de se implementar todos os métodos que sejam necessários a seu funcionamento.

# Objetivo

---



Propor um *framework* que englobe algumas das funcionalidades necessárias ao desenvolvimento de ferramentas geradoras de eventos que podem oferecer suporte a estudos de desempenho e dependabilidade.

# FlexLoadGenerator

---



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

- Algumas de suas funcionalidades são:
  - Estabelecer conexão com o sistema alvo através dos protocolos TCP e UDP;
  - Oferece também o estabelecimento de comunicação por meio de SSH2;
  - Criação e gerenciamento de eventos; e
  - Uso de distribuição de probabilidade.

# FlexLoadGenerator

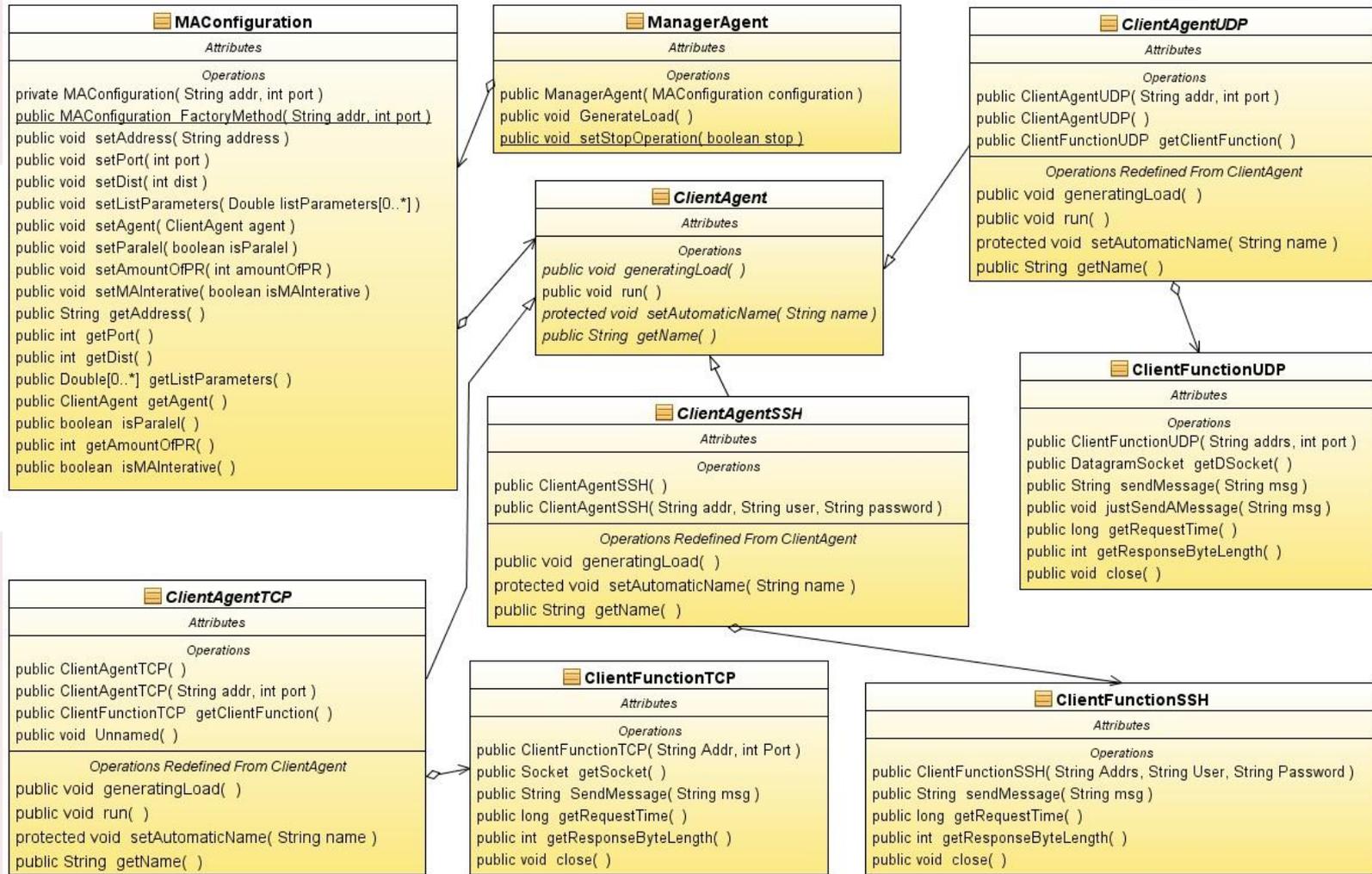


Diagrama de classe FlexLoadGenerator

# FlexLoadGenerator

---



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

- Para criar um gerador de eventos, o desenvolvedor deve escolher um protocolo de comunicação (TCP, UDP ou SSH).
- Especificar o evento que será criado através da escrita do método *generatingLoad()*;

# FlexLoadGenerator

---



- Determinar os parâmetros necessários para geração de eventos solicitados pela classe *MAConfiguration*; e
- Fazer a chamada do método *GeneratingLoad()* responsável pelo gerenciamento dos eventos.



# FlexLoadGenerator

---



## Possíveis modos de execução:

- Paralelo e interativo: Cria a quantidade de processos paralelos determinada pelo usuário e aguarda um tempo aleatório a partir de uma distribuição para criar novamente a mesma quantidade de processos. Neste modo de execução não é necessário a “morte” dos processos em execução para a criação de outros;
- Não paralelo e interativo: Cria apenas um processo e este processo se repetirá ao longo da execução da ferramenta. Aguarda o tempo determinado a partir da distribuição escolhida para que o mesmo processo seja chamado novamente;

# FlexLoadGenerator

---



Possíveis modos de execução:

- Paralelo e não interativo: Cria a quantidade de processos determinada pelo usuário e ao final da execução os processos são terminados;
- Não paralelo e não interativo: O mais simples dos métodos de execução. Cria apenas um único processo e ao final da execução este é terminado.

Nestes dois últimos modos de operação, se o usuário desejar que o processo continue a executar é preciso implementar um *loop* dentro do método *generatingLoad()*;

# Ferramentas Desenvolvidos

---



## WGSysEFT

- Gerador de carga de trabalho para sistemas de Transferência Eletrônica de Fundos;
- Atua realizando transações completas de crédito e débito nas modalidades a vista ou parcelado;
- Faz uso da distribuição de probabilidade escolhida pelo usuário para intervalo de tempo entre o final de uma requisição e o início de outra;



# Ferramentas Desenvolvidos

---



## WGSysEFT

- O usuário pode informar percentuais de para geração de carga de acordo com as transações escolhidas; e
- Usuário determina o tempo de operação da ferramenta;
- Pode ser utilizado de forma distribuída.



# Ferramentas Desenvolvidos

---



## EucaBomber

- Gerador de eventos de falhas e reparos voltado para ambiente de nuvens criados e gerenciados a partir do Eucalyptus;
- Usuário escolhe se quer criar apenas falhas ou se também deseja reparar as falhas injetadas;
- Atua sobre os processos dos componentes do Eucalyptus: Cloud Controller, Node Controller, Cluster Controller, Storage Controller e Walrus;

# Ferramentas Desenvolvidos

---



## EucaBomber

- Atua também suspendendo a execução do hardware no qual a nuvem está inserida;
- Faz uso da distribuição de probabilidade escolhida pelo usuário para intervalo de tempo entre eventos; e
- Usuário determina o tempo de operação da ferramenta.



# Alguns resultados

## EucaBomber: Cenários

- Execução com duração de 48hs;
- Cinco máquinas compondo a *cloud* e um cliente injetando falhas e reparos.

Type of component	Experiment		Real	
	MTTF	MTTR	MTTF	MTTR
Software	2836.8 s	14.4 s	788 h	4 h
Hardware	31536 s	28.8 s	8760 h	9 h

Cenário 1a e 2a - Exponencial

Type of component	Experiment		Real	
	MTTF	MTTR	MTTF	MTTR
Software	2836.8 s	17.3 s	788 h	4.8 h
Hardware	31536 s	34.5 s	8760 h	9.6 h

Cenário 1b e 2b – Erlang

Metric	Scenarios			
	1a	1b	2a	2b
Availability (%)	94.4213	93.1626	95.4167	94.8438
Unavailability (hardware) (%)	1.1892	1.3194	1.1140	1.1574
Unavailability (software) (%)	4.3895	5.5179	3.4693	3.9988
Unavailability (total) (%)	5.5787	6.8374	4.5833	5.1563

Resultados cenários 1a, 1b, 2a e 2b

