

Supporting Dynamic Reconfiguration in Distributed Data Stream Systems

Rafael Oliveira Vasconcelos^{1,2}

Igor Vasconcelos^{1,2}

Markus Endler¹

Sérgio Colcher¹

¹Departamento de Informática (DI)

Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

²Universidade Tiradentes (UNIT)

{[rvasconcelos](mailto:rvasconcelos@inf.puc-rio.br), [ivasconcelos](mailto:ivasconcelos@inf.puc-rio.br), [endler](mailto:endler@inf.puc-rio.br), [colcher](mailto:colcher@inf.puc-rio.br)}@inf.puc-rio.br

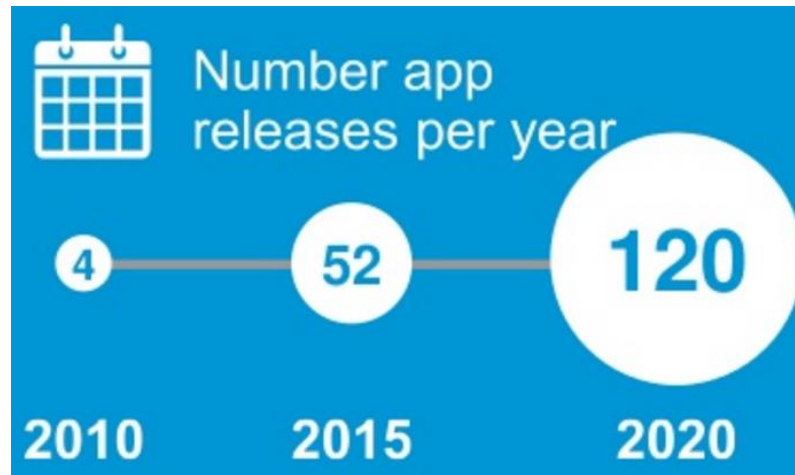
Agenda

- **Introdução**
- **Fundamentação Teórica**
- **Abordagem**
- **Avaliação Preliminar**
- **Conclusão**



Motivação

- Sistemas com operação 24/7
 - Interrupções não são aceitáveis
- Mudanças nos requisitos e ambiente de execução
 - Dificuldade de serem previstas em tempo de projeto



Fonte: [1]



Motivação

- Adaptação dinâmica de *software* é uma tarefa desafiadora
- Abordagens anteriores requerem uso de *hardware* adicional
- Sistemas de processamento de fluxo de dados
 - Processamento contínuo e *online*
 - Dispositivos móveis



Objetivo e Contribuições

■ Objetivo

- Desenvolver uma abordagem não bloqueante para reconfiguração dinâmica que preserve a consistência global em sistemas de processamento de fluxo de dados distribuídos

■ Contribuições

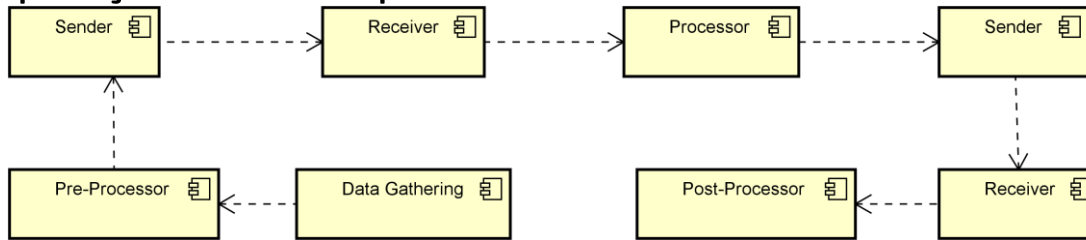
- O estudo de um mecanismo para garantir reconfiguração consistente e não bloqueante de sistemas de processamento de fluxo de dados distribuídos com suporte a nós móveis
- Um protótipo de middleware que implementa o mecanismo
- Experimentos que validam solução proposta



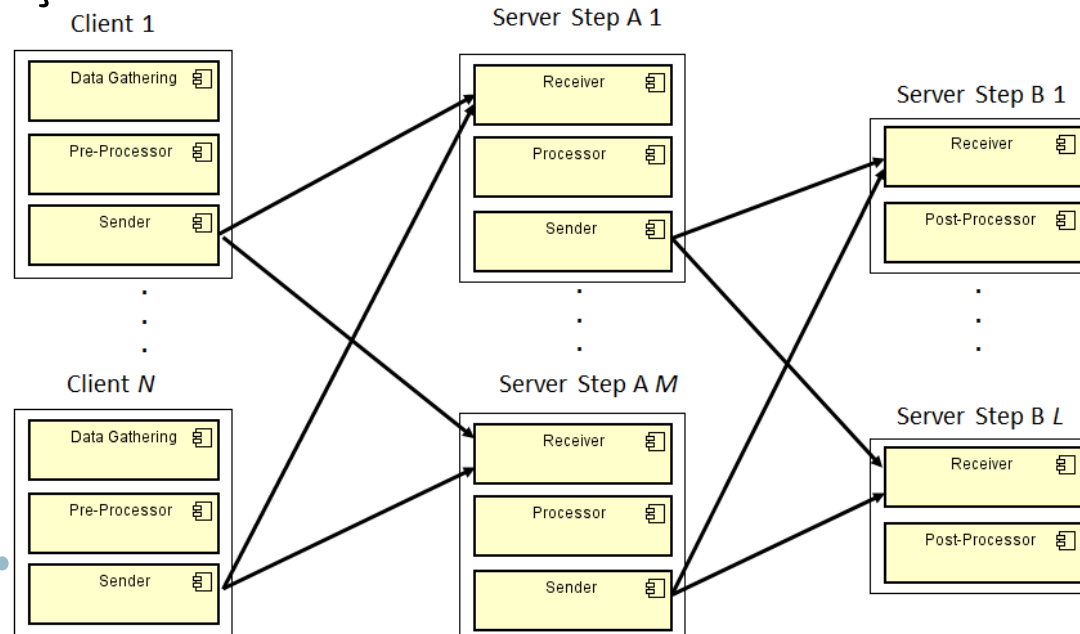
Cenário Motivador

- Aplicação colaborativa móvel para compartilhamento da qualidade de rotas de bicicleta

- Decomposição dos componentes



- Implantação do sistema



Fundamentação Teórica

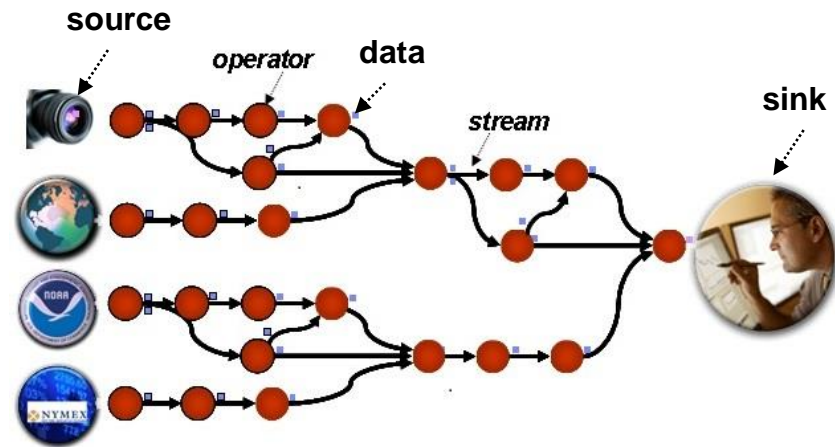
■ Processamento de fluxo de dados

■ O sistema deve

- Manter os dados em movimento
- Gerar resultados previsíveis
- Garantir a disponibilidade dos dados
- Processar e responder rapidamente

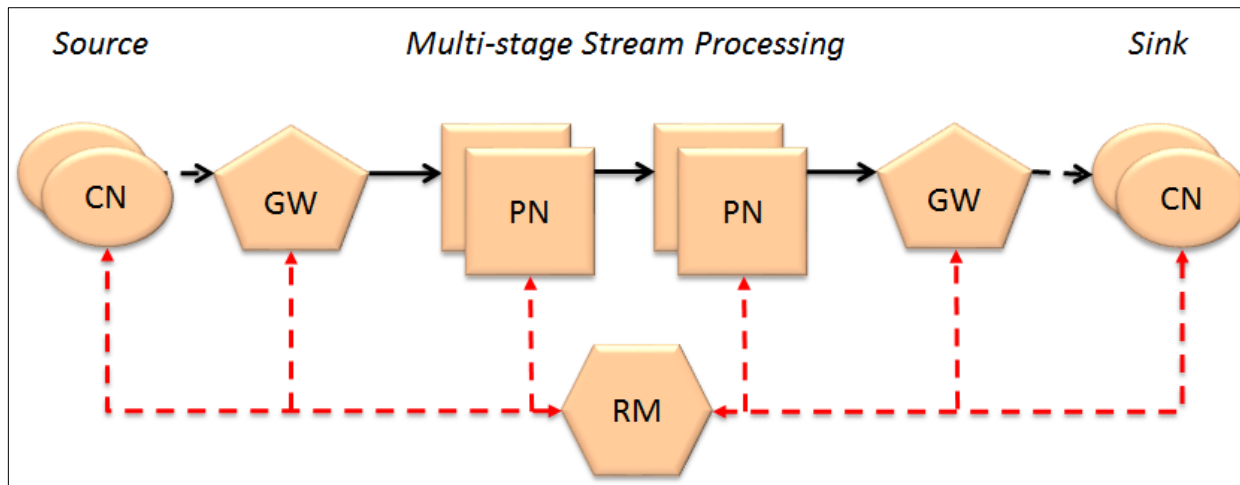
■ A plataforma de reconfiguração deve

- Ser confiável
 - Não produzir resultados errôneos e impactar a disponibilidade do sistema
- Não interromper ou bloquear o fluxo de dados



Fundamentação Teórica

- Modelo do sistema
 - Grafo acíclico direcionado $G = (V, E)$
 - Vértice representa um operador (i.e., um componente)
 - Funções $f_{seleção}$, $f_{saída}$ e $f_{atualização}$
 - Aresta representa um canal de fluxo (i.e., *stream channel*)
 - Aresta $e = (v1, v2)$
 - Tupla $t = (valor, caminho^*)$
 - Tupla do vértice fonte $VF1$ para o vértice consumidor $VC1$ através dos operadores $O1$ e $O2$
 - Caminho de t ao chegar em $VC1 = \{VF1, O1, O2\}$



Fundamentação Teórica

■ Trabalhos Relacionados

Critério	(Ertel & Felber, 2014)	(Ma et al., 2011)	(Ghafari et al., 2012a)	Abordagem proposta
Interrupção	Requer parada do componente e transferência do estado	Versões antiga e nova executam em paralelo	Versões antiga e nova executam em paralelo	Múltiplas versões executam em paralelo
Método de reconfiguração	Uso de marker para troca de versão, reconf. transacional	Uso de grafo para atualização gradativa	Uso de timestamp para atualização gradativa	Uso de função de seleção e marker para atualização gradativa
Suporte a múltiplas instâncias	Sim	Não	Não	Sim
Distribuído	Sim	Não	Não	Sim
Suporte a nós móveis	Não	Não	Não	Sim
Reconfiguração progressiva	Não	Sim	Sim	Sim



Abordagem Proposta

- Componentes com múltiplas versões
 - Dado produzido deve ser processado por uma versão específica de cada componente
 - Componente
 - Pode ser atualizado enquanto fluxo de dados está em movimento
 - Tem uma ou mais funções $f^{\text{seleção}}$, $f^{\text{atualização}}$ e $f^{\text{saída}}$
 - Função $f^{\text{seleção}}$ escolhe a $f^{\text{atualização}}$ correta para processar a tupla
 - Uso do caminho da tupla

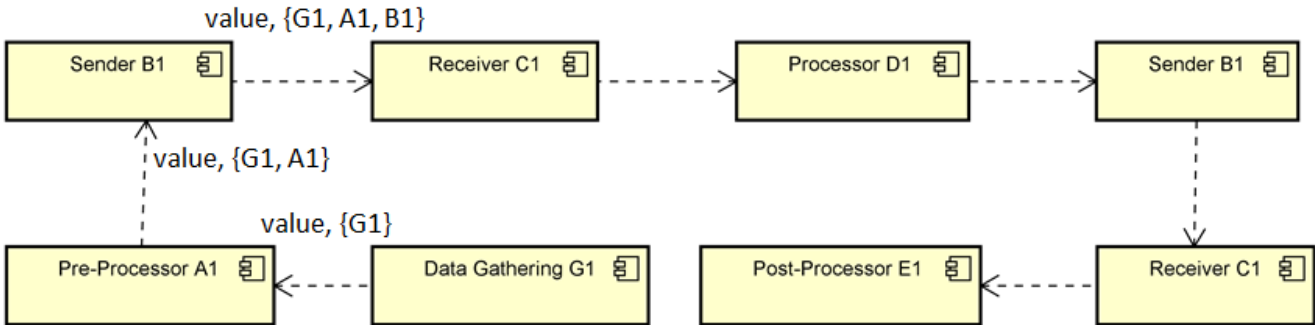


Abordagem Proposta

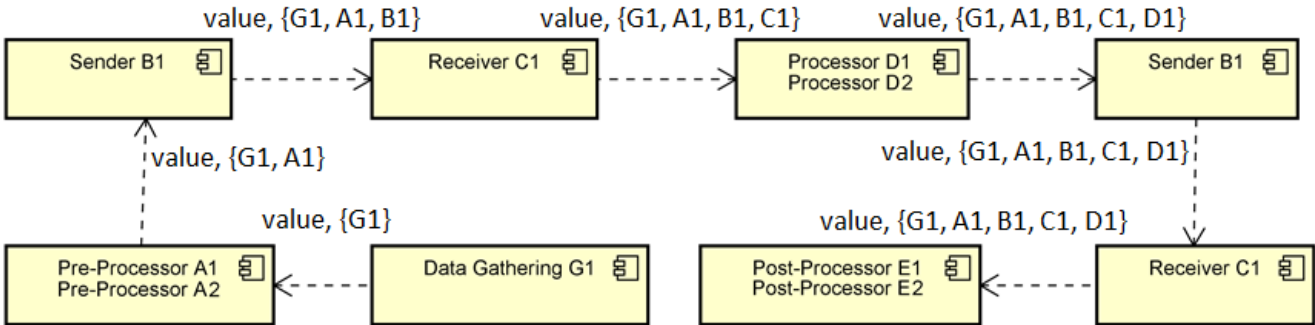
- Gerenciamento de dependência

 - Gerenciamento estático

 - Fluxo parcial



 - Fluxo com sistema parcialmente reconfigurado



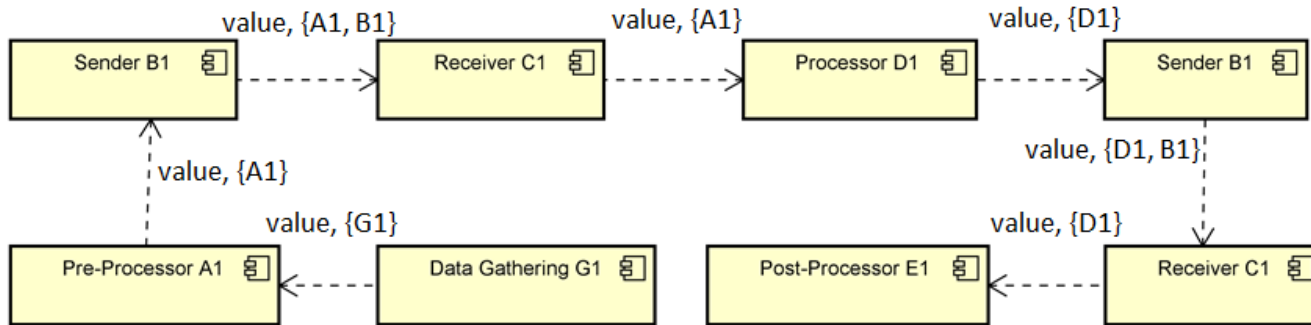
Abordagem Proposta

- Gerenciamento de dependência
 - Gerenciamento estático
 - Vantagens
 - Simplicidade
 - Custo computacional
 - Alteração de dependência (e.g., inserção ou remoção de componentes) não afeta a reconfiguração
 - Desvantagens
 - Caminho de execução grande pode degradar o desempenho do sistema por conta dos custos de rede e memória



Abordagem Proposta

- Gerenciamento de dependência
 - Gerenciamento dinâmico
 - Adiciona versão somente se houver algum componente *downstream* dependente
 - Remove as versões que não têm mais um componente *downstream* dependente



Abordagem Proposta

- Gerenciamento de dependência
 - Gerenciamento dinâmico
 - Vantagens
 - Caminho de execução reduzido
 - » Não desperdiça rede e memória visto que o caminho de execução contém apenas informação útil
 - Desvantagens
 - Custo para manter o caminho de execução o mais curto possível e a consistência do sistema quando a dependência de parte do sistema é alterada

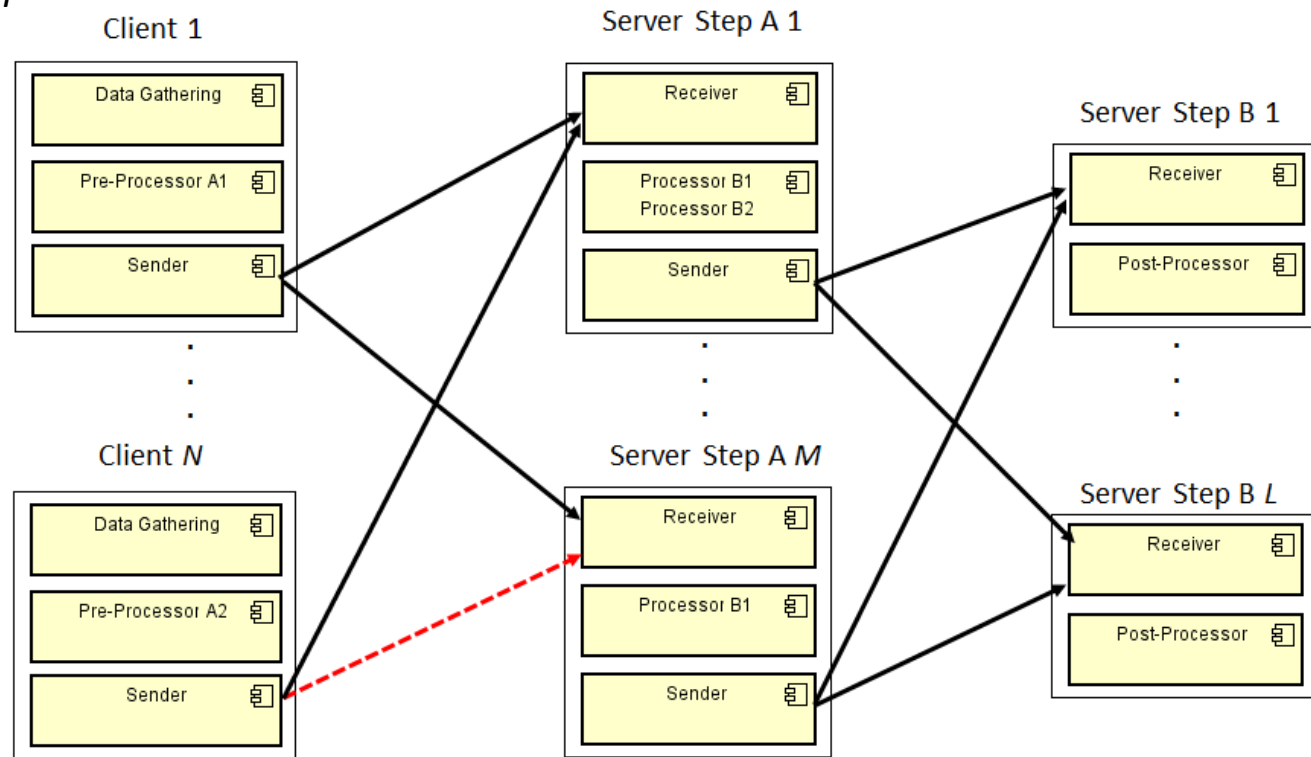


Abordagem Proposta

■ Reconfiguração Distribuída Multi-Instância

■ Atualização gradativa

- Versão antiga de componente *downstream* removida após atualização dos componentes *upstream*



Reconfiguração parcial **inconsistente**

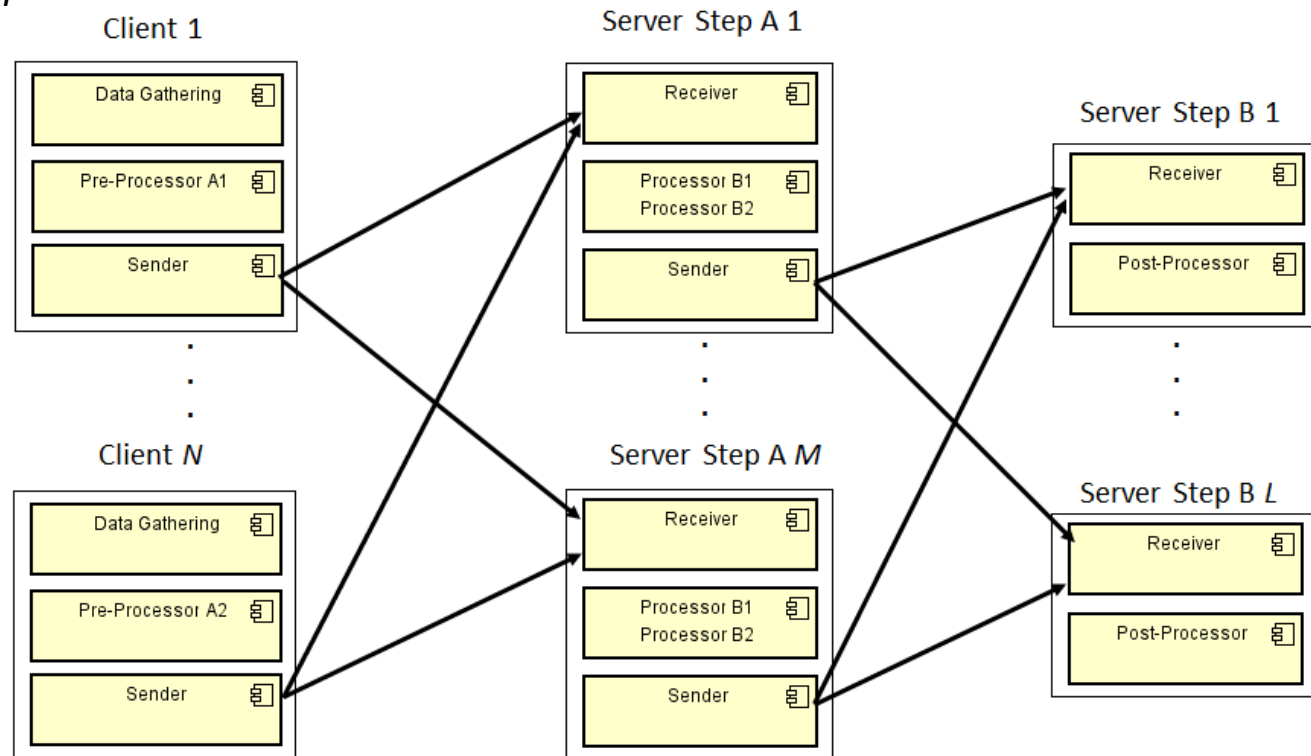


Abordagem Proposta

■ Reconfiguração Distribuída Multi-Instância

■ Atualização gradativa

- Versão antiga de componente *downstream* removida após atualização dos componentes *upstream*

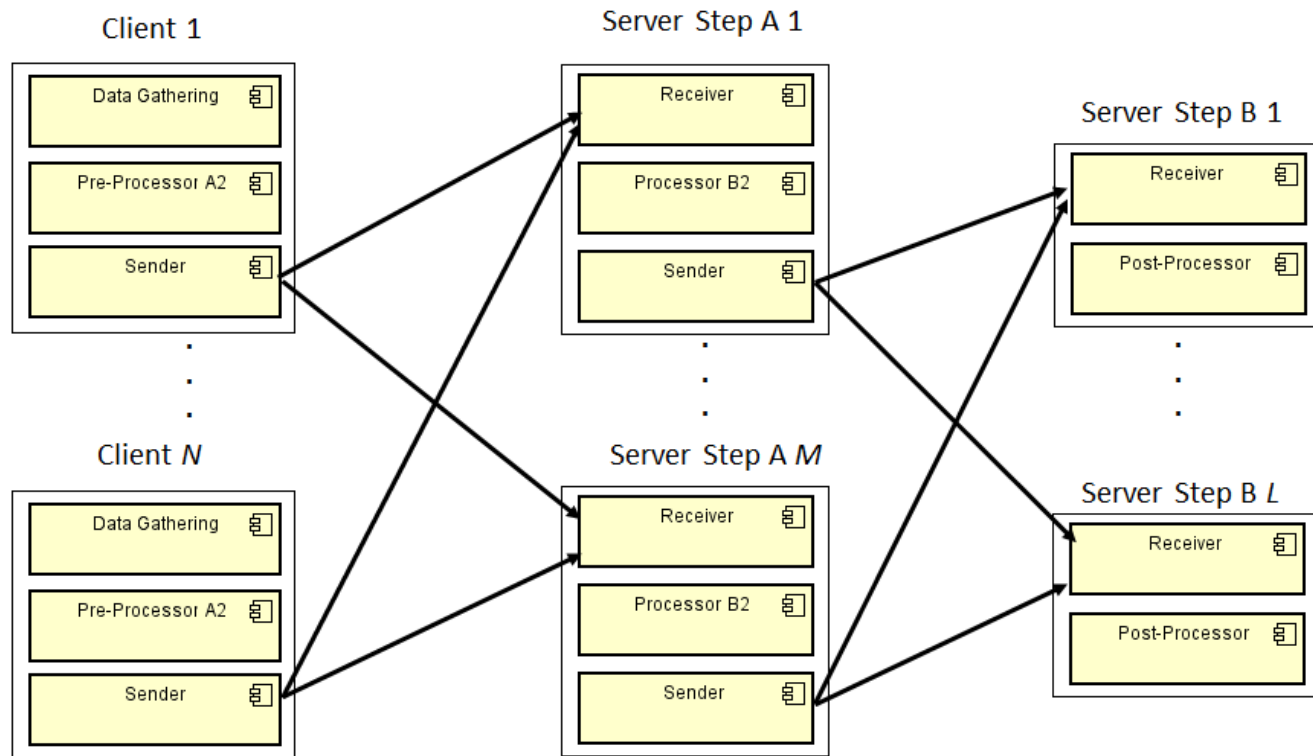


Reconfiguração parcial consistente



Abordagem Proposta

- Reconfiguração Distribuída Multi-Instância
 - Remoção utilizando marcadores
 - Componente adiciona marcador em todas suas portas de saída



Sistema reconfigurado



Abordagem Proposta

- Reconfiguração Distribuída Multi-Instância
 - Nó com conexão intermitente
 - Reconfiguração adiada até a conexão ser reestabelecida
 - Desconexão (permanente) de nó causa sua remoção do sistema



Avaliação Preliminar

■ Métricas

- Tempo de atualização
- Interrupção do serviço
- Desconexão de nós móveis

■ Configuração dos testes



Intel i5-3210M,
8GB RAM



802.11n router



Intel i5 M480,
8GB RAM

Avaliação Preliminar

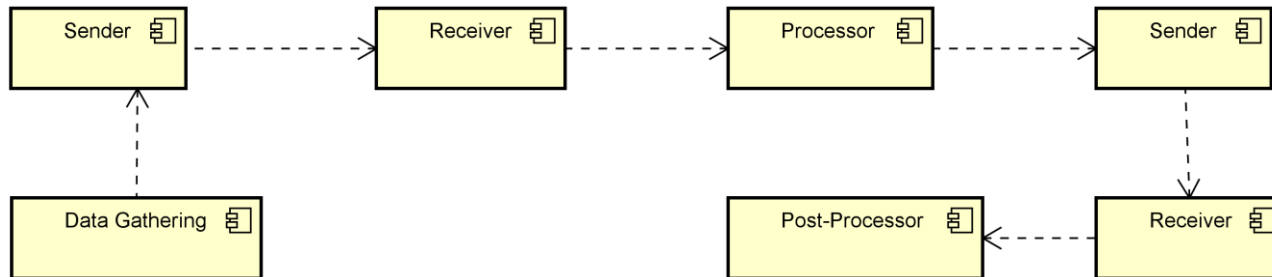
- Configuração dos testes
 - Variação do:
 - Número de clientes
 - 1, 10, 100 e 1.000
 - Taxa de produção de tuplas do sistema (tuplas/s)
 - 10, 100 e 1.000



Avaliação Preliminar

■ Aplicação protótipo

■ Modelo de componentes

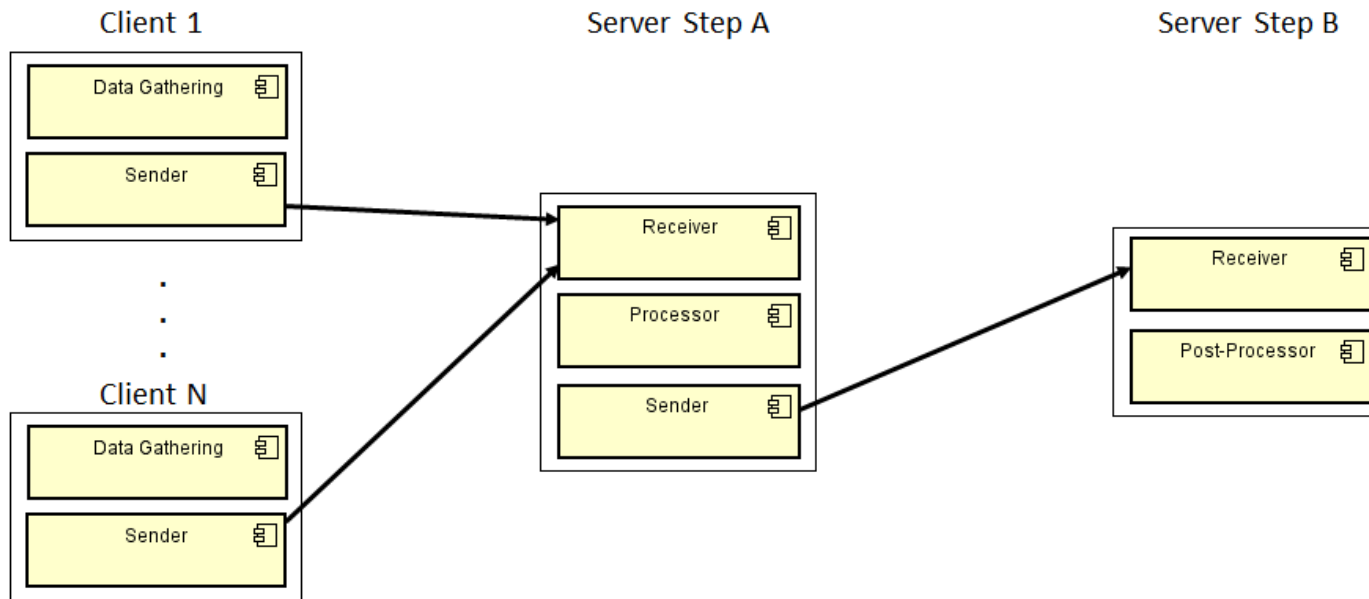


■ Reconfiguração

- Codificação de **ISO-8859-1** para **UTF-16**

Avaliação Preliminar

- Aplicação protótipo
 - Implantação dos componentes



Avaliação Preliminar

- Resultados
 - Tempo de atualização

# CNs	Taxa de Produção de Tuplas (tuplas/s)	Tempo de Atualização (ms)
1	10	10,05
10	10	9,86
100	10	10,77
1.000	10	10,37
1	100	9,64
10	100	10,01
100	100	9,87
1.000	100	10,49
1	1.000	10,39
10	1.000	10,72
100	1.000	11,02
1.000	1.000	11,88

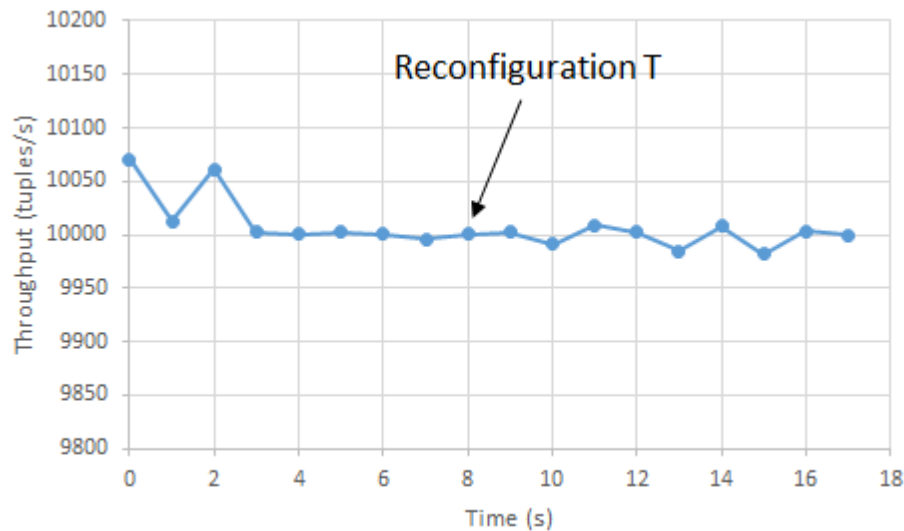


Avaliação Preliminar

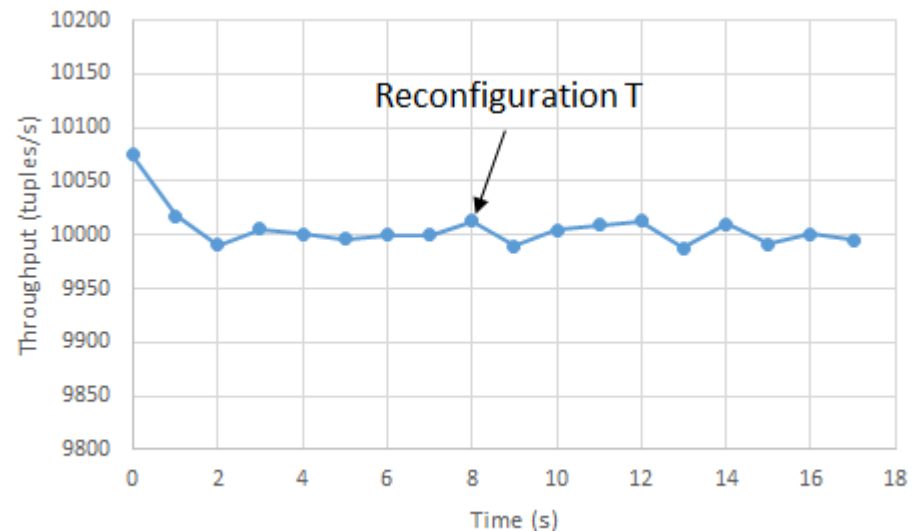
■ Resultados

■ Interrupção do serviço

- Vazão não afetada durante a reconfiguração em 2 dos 4 testes



Cenário com 1 CN e 1.000 tuplas/s



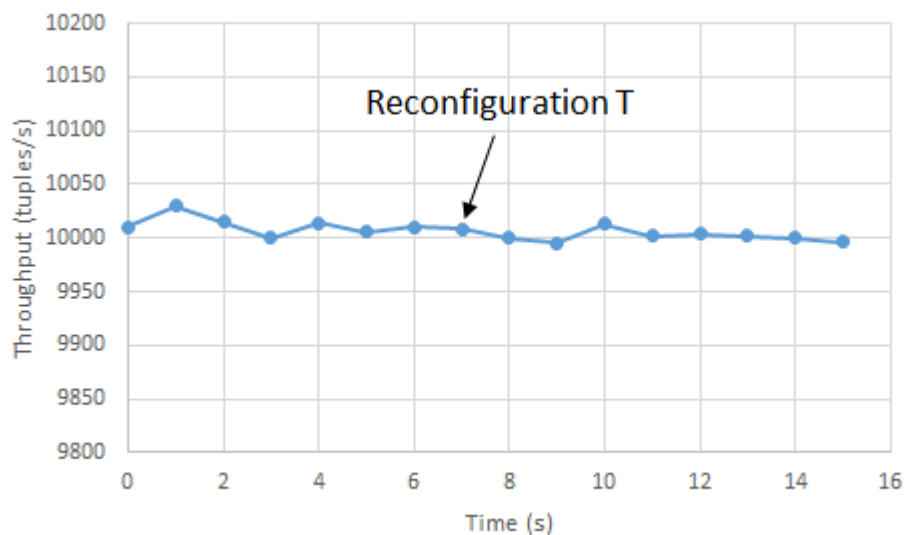
Cenário com 100 CNs e 1.000 tuplas/s

Avaliação Preliminar

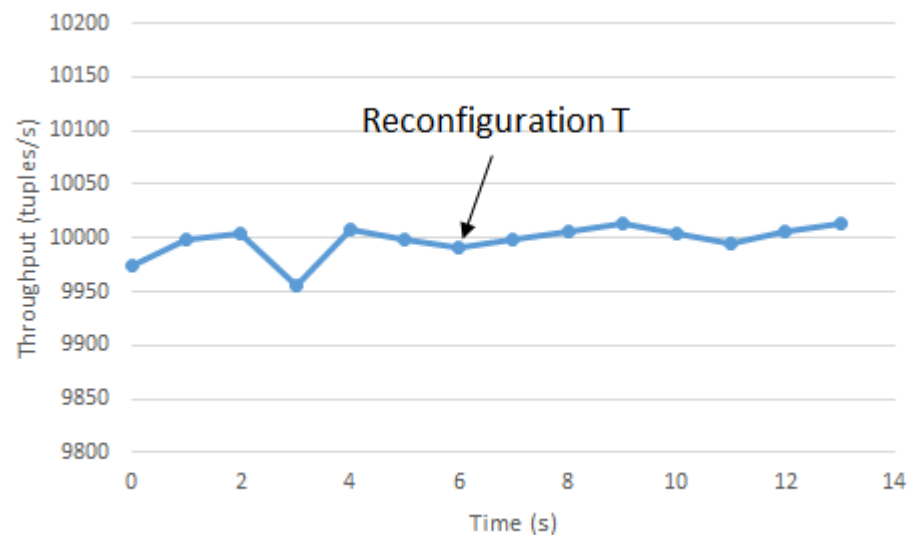
■ Resultados

■ Interrupção do serviço

- Vazão reduzida em no máximo 0.07% durante reconfiguração em 2 dos 4 testes



Cenário com 10 CNs e 1.000 tuplas/s



Cenário com 1.000 CNs e 1.000 tuplas/s



Avaliação Preliminar

- Resultados
 - Desconexão de nós móveis

# CNs	Taxa de Produção de Tuplas (tuplas/s)	Tempo de Reconexão (ms)
1.000	1.000	31,50



Conclusão

- Aplicações 24/7
- Processamento de fluxo de dados
- Interrupções não são aceitáveis
- Necessidade do sistema evoluir
- Abordagem distribuída com múltiplas versões em paralelo
- Remoção segura de componente
- Suporte a dispositivos móveis
- Avaliação preliminar



Obrigado!

Q&A

Para mais informações:

rvasconcelos@inf.puc-rio.br

endler@inf.puc-rio.br

www.lac.inf.puc-rio.br

