

# Escalonador de Transações para arquitetura NUMA

**Michael Alexandre Costa**

Prof. Dr. André Rauber Du Bois (Orientador)

Mestrado em Computação  
Centro de Desenvolvimento Tecnológico  
Universidade Federal de Pelotas  
`macosta@inf.ufpel.edu.br`

8 de junho de 2021



- 1 **Introdução**
- 2 **Memórias Transacionais**
- 3 **Escalonadores**
- 4 **Arquiteturas**
- 5 **Objetivos**
- 6 **Próximas Atividades**
- 7 **Cronograma de Atividades**



# Introdução

## Motivação

- Programação Paralela;
- Memórias Transacionais;
- Escalonadores de Transações; e
- Arquiteturas NUMA.

## Uso Comercial

- Wyatt Technology;
- Sistema de coleta e análise de dados na biomedicina; e
- Apresentam problemas de desempenho conforme aumentam o paralelismo.



# Introdução

## Objetivos

- Analisar o comportamento de escalonadores de transações em arquiteturas NUMA; e
- Fornecer um escalonador de transações voltado a arquitetura NUMA.



# Memórias Transacionais

## Características

- Fornece abstração de código;
- Reuso de código; e
- Ausência de deadlocks.

## Transações

- Atomicidade;
- Consistência; e
- Isolamento.



# Memórias Transacionais

## Controle das transações

- Versionamento de Dados:
  - Adiantado / Tardio.
- Detecção de Conflitos:
  - Adiantado / Tardio.
- Gerenciamento de Contenção:
  - Suicide, Delay, Backoff ou Modular.



# Memórias Transacionais

## Versionamento Adiantado

- Escreve os dados especulativos direto na memória; e
- Em caso de um cancelamento, a operação deve ser desfeita.

## Versionamento Tardio

- Escreve os dados especulativos em um *buffer* local; e
- Em caso de efetivação, os dados devem ser copiados para a memória.



# Memórias Transacionais

## Detecção de Conflitos Adiantada

- Detecta conflito no momento do acesso a memória.

## Detecção de Conflitos Tardia

- Detecta conflito somente na validação.





# Memórias Transacionais

## Gerenciador de Contenção

- Possui ação reativa;
- Suicide, Delay, Backoff ou Modular.

## Gerenciador de Contenção

- Modular:
  - Suicide, Delay, Aggressive, e Timestamp.



# Memórias Transacionais

## Problemas

- Somente reinicia a transação conflitante;
- Não evita que conflitos futuros aconteçam; e
- Em ambientes de alta contenção, tende a perder desempenho.



# Escalonadores

## Escalonadores de Transações

- Buscam reduzir os números de conflitos;
- Utilizam diferentes Heurísticas de escalonamento; e
- Serializa as transações conflitantes.



# Escalonadores

## Classificação das técnicas

- Baseado em Heurística:
  - Feedback;
  - Predição;
  - Reativo; e
  - Heurística Mista.
- Baseado em Modelo:
  - Aprendizado de Máquina;
  - Modelo Analítico; e
  - Modelo Misto.



# Escalonadores

## Trabalhos Relacionados

**Tabela:** Algoritmos e técnicas de escalonamento

Escalonador	Técnica
ATS	Feedback
Probe	Feedback
F2C2	Feedback
Shrink	Predição
SCA	Predição
CAR-STM	Reativo
RelSTM	Reativo
LUTS	Heurística Mista
ProVIT	Heurística Mista
SAC-STM	Aprendizado de Máquina
CSR-STM	Modelo Analítico
MCATS	Modelo Analítico
AML	Modelo Misto



# Escalonadores

## Trabalhos Relacionados

**Tabela:** Algoritmos que estamos trabalhando

Escalonador	Técnica
Probe	Feedback
F2C2	Feedback
Shrink	Predição
MCATS	Modelo Analítico



# Escalonadores

## Shrink

- Bloom filter: Utiliza os dados de leitura e escrita por thread:
  - Conjunto de leitura: Localidade temporal; e
  - Conjunto de escrita: Ocorre apenas nos aborts.
- Serialization affinity: Serializa uma thread de acordo com a contenção do sistema; e
- O escalonador é ativado com base no número de contenção existente.



# Arquiteturas

## UMA

- Uniform Memory access;
- Possui um único barramento de acesso à memória; e
- Único custo de acesso à memória.

## NUMA

- Non-uniform Memory access;
- Possui mais de um barramento de acesso à memória; e
- O custo de acesso à memória é diferente conforme o núcleo utilizado.





# Objetivos

## Objetivos

- Estudar o comportamento dos escalonadores na arquitetura NUMA;
- Inserir as novas regras de escalonamento para arquitetura NUMA.



# Metodologia

## Ferramentas utilizadas

- Shrink;
- TinySTM;
- Hwloc; e
- STAMP.



# Metodologia

## O que foi feito

- Foi implementado um escalonador com filas de threads para cada núcleo;
- Foi feito um escalonador que migra threads;
- Foram estudados os algoritmos de escalonamentos atuais; e
- Foi desenvolvido um novo fluxo de execução para o Shrink.



# Metodologia

## O que será feito

- Modificar a implementação de threads do Shrink para utilizar filas;
- Coletar informações da latência de acordo com o Bloom Filter; e
- Adicionar a migração de threads ao Shrink.



# Metodologia

## Modificações e nomenclatura

- Cada núcleo possuirá uma fila de threads que chamamos de  $Q_n$ ;
- O escalonador possuirá uma fila de threads inicial chamada de  $P_t$ ; e
- Uma Thread ( $T_n$ ) pode ter  $n$  transações que chamamos de  $Tr$ .



# Metodologia

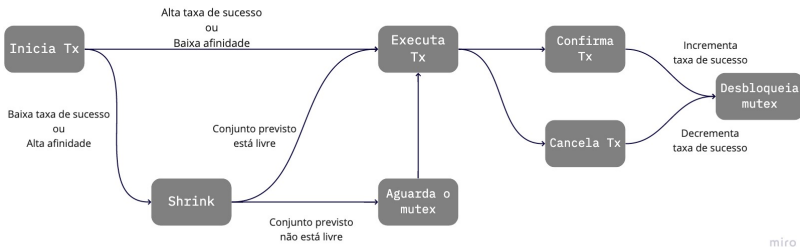


Figura: Shrink

# Metodologia

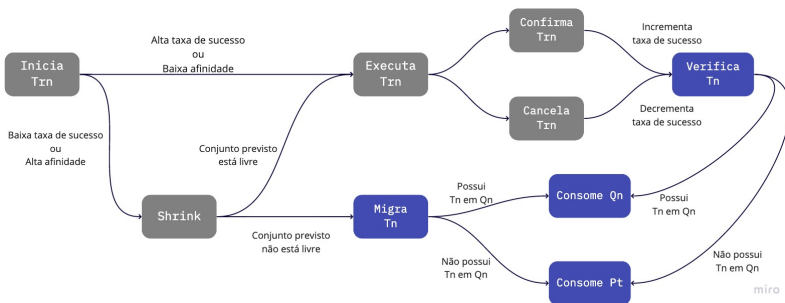


Figura: Modificações

# Próximas Atividades

## Atividades a serem realizadas

- Modificar o escalonador Shrink;
- Executar os testes;
- Analisar resultados; e
- Escrever a Dissertação.





# Cronograma

- ➊ Modificações no Shrink coletando informações sobre a arquitetura;
- ➋ Modificações no método de escalonamento do Shrink;
- ➌ Validação do novo método de escalonamento;
- ➍ Execução de testes em arquitetura NUMA e UMA;
- ➎ Coleta de resultados obtidos por meio dos testes;
- ➏ Escrita da dissertação; e
- ➐ Entrega e apresentação da dissertação.



# Cronograma

**Tabela:** Cronograma de atividades mensal para o restante do mestrado

Ano	2020					2021	
Mês	Ago	Set	Out	Nov	Dez	Jan	Fev
1							
2							
3							
4							
5							
6							
7							

# Escalonador de Transações para arquitetura NUMA

**Michael Alexandre Costa**

Prof. Dr. André Rauber Du Bois (Orientador)

Mestrado em Computação  
Centro de Desenvolvimento Tecnológico  
Universidade Federal de Pelotas  
`macosta@inf.ufpel.edu.br`

8 de junho de 2021

