OTIMIZANDO STARVZ PARA CARGA DE GRANDES VOLUMES DE DADOS

Alexandre Miyazaki

Orientador: Lucas Mello Schnorr

Especialização em Big Data & Data Science — 13 de Setembro de 2019

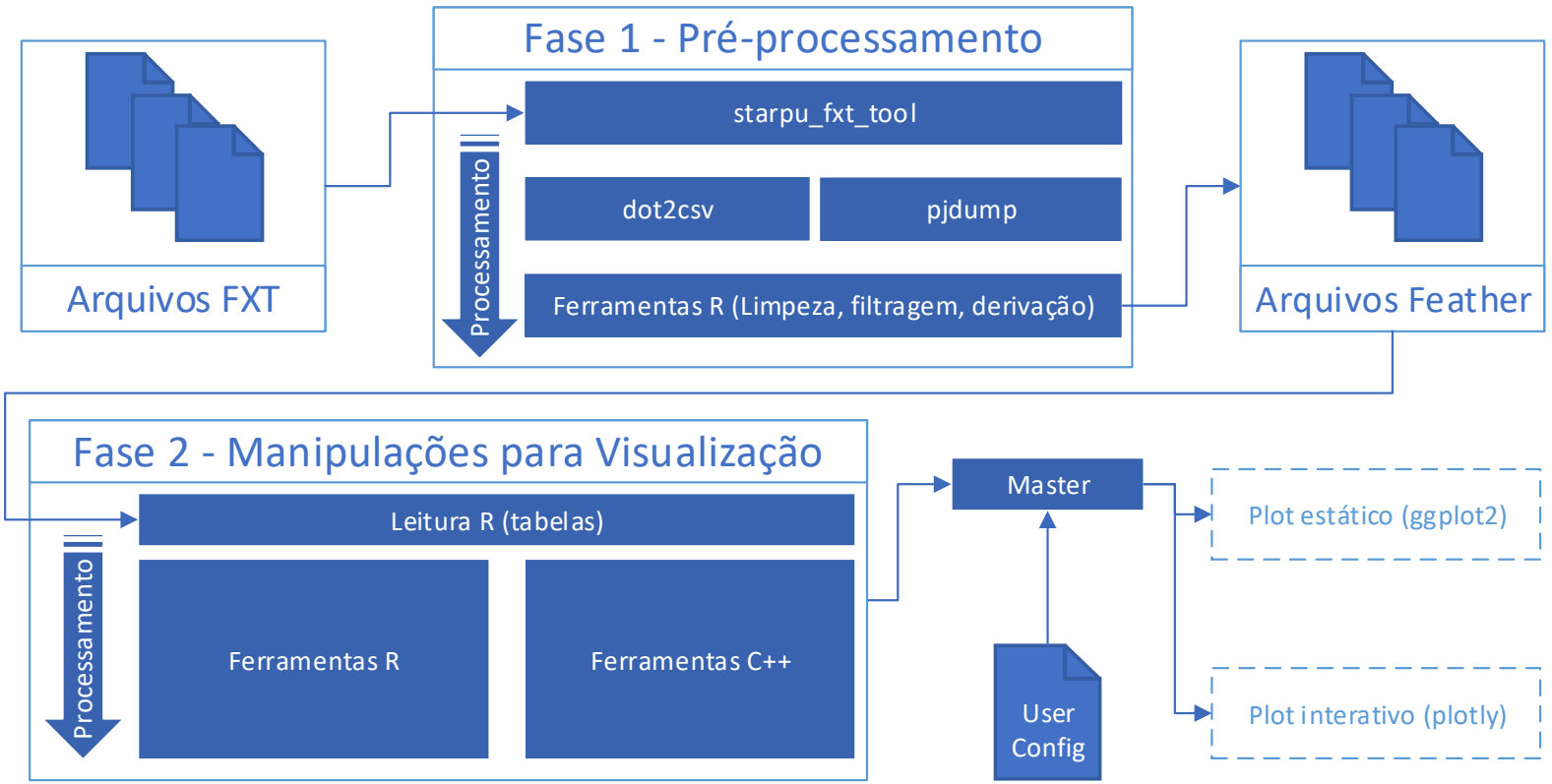
AGENDA

1. Introdução
2. Ferramentas
3. Implementação
4. Avaliação e Resultados
5. Conclusão

STARVZ

- StarVZ é um arcabouço de análise de desempenho cujo objetivo é auxiliar na verificação de hipóteses sobre aplicações baseadas em tarefas.
- Este domínio carece de ferramentas de visualização voltadas para este tipo de aplicação, devido a dominância do modelo predecessor (BSP - *Bulk-Synchronous Parallel*).
- Ele é separado em duas fases sendo uma de pré-processamento e outra de visualização.

STARVZ - FASES



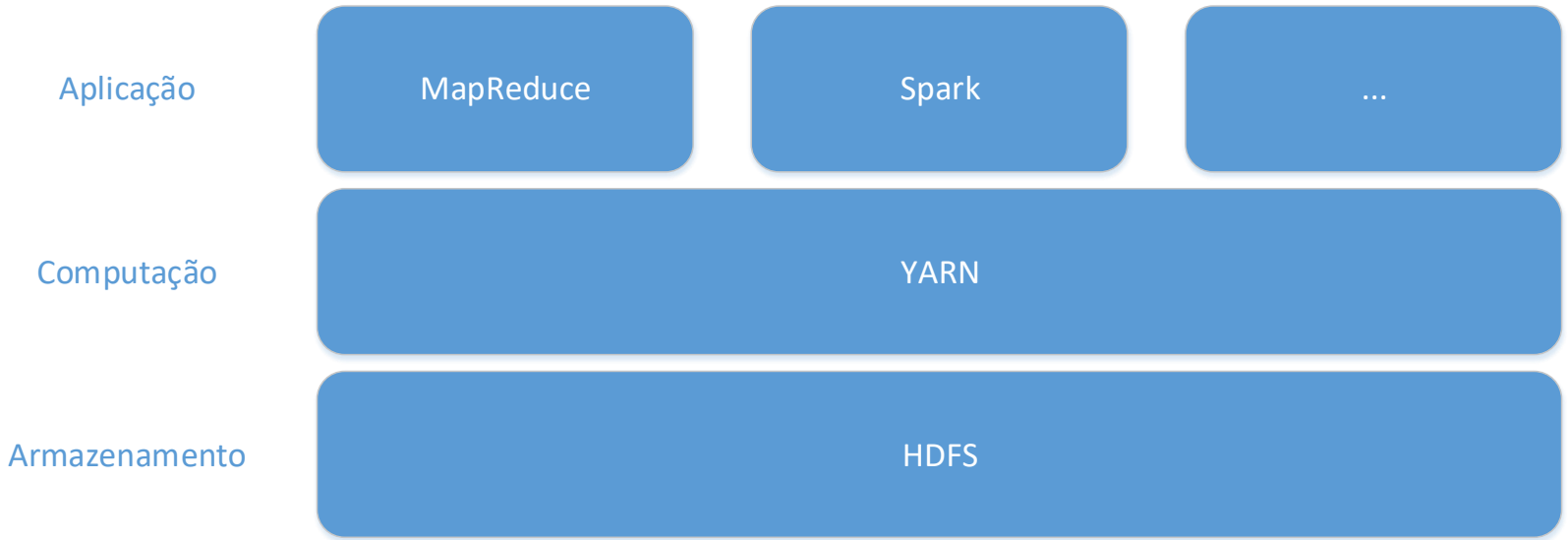
MOTIVAÇÃO/TRABALHOS RELACIONADOS

- Em estudos anteriores, analisou-se logs com 18 GB, derivados de uma pequena execução envolvendo poucos nós computacionais.
- A primeira fase do StarVZ levou cerca de 32 minutos para processar.
- Tal desempenho pode inviabilizar o uso do StarVZ para cargas de grandes volumes de dados.
- Houve uma tentativa de otimizar o fluxo do StarVZ com `Drake`, biblioteca cujo foco é executar apenas as fases necessárias de um fluxo de processamento e análise.
- Não foi possível otimizar o fluxo com `Drake`, devido a cache de resultados intermediários da biblioteca ter um uso intensivo de disco.

OBJETIVO

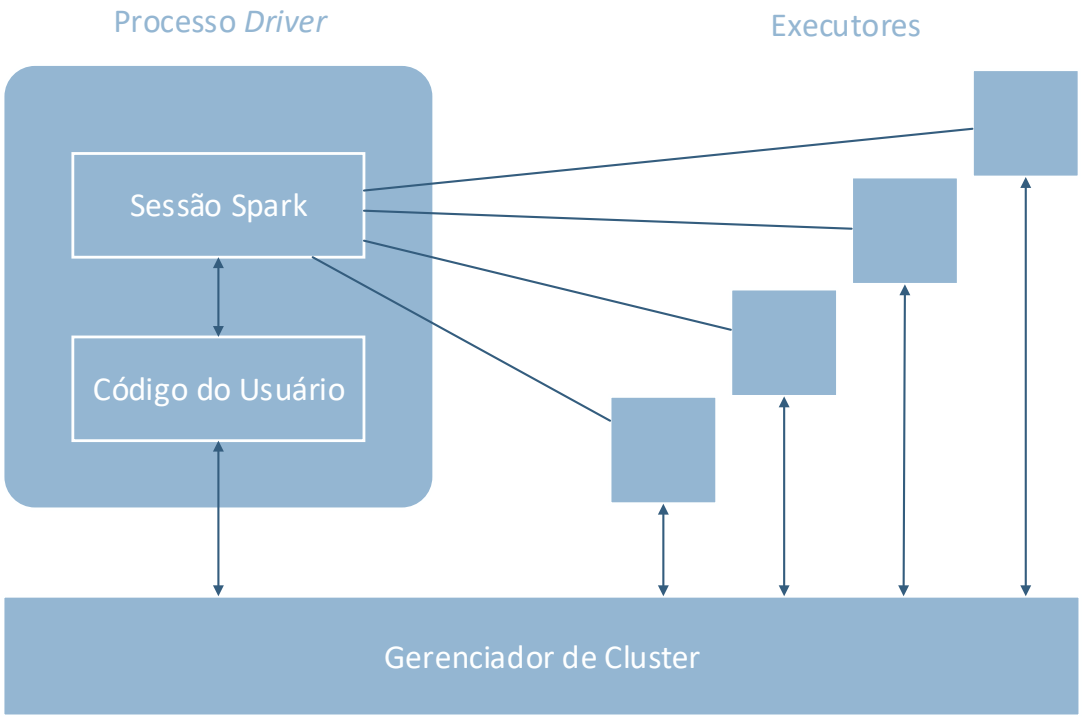
Viabilizar o uso do StarVZ para análise de grandes volumes de dados utilizando ferramentas voltadas para este fim.

HADOOP



- Arcabouço que permite o processamento de grandes volumes de dados.
- Organização em camadas.

SPARK

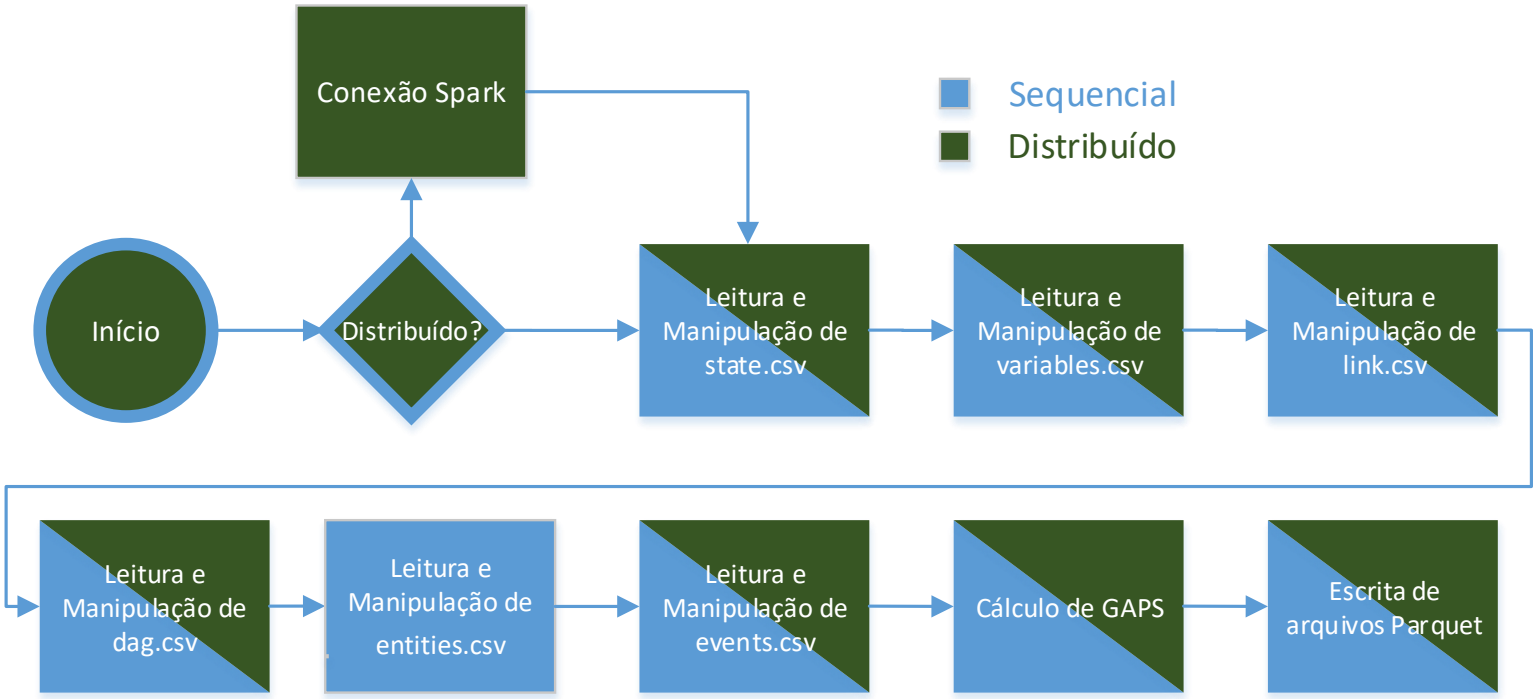


- *Engine* unificada + conjunto de bibliotecas para processamento de dados distribuídos.
- Utilizado via biblioteca `sparklyr`.

ABORDAGEM

- Otimizar processamento de tabelas, não o fluxo da aplicação.
- Focado nas ferramentas R, que consistem em ~40% do tempo total da primeira fase.
- Utilizar equivalências entre `sparklyr` e `dplyr`.
- Processo foi facilitado pois a `sparklyr` é baseada na dplyr.

FLUXO DA APLICAÇÃO



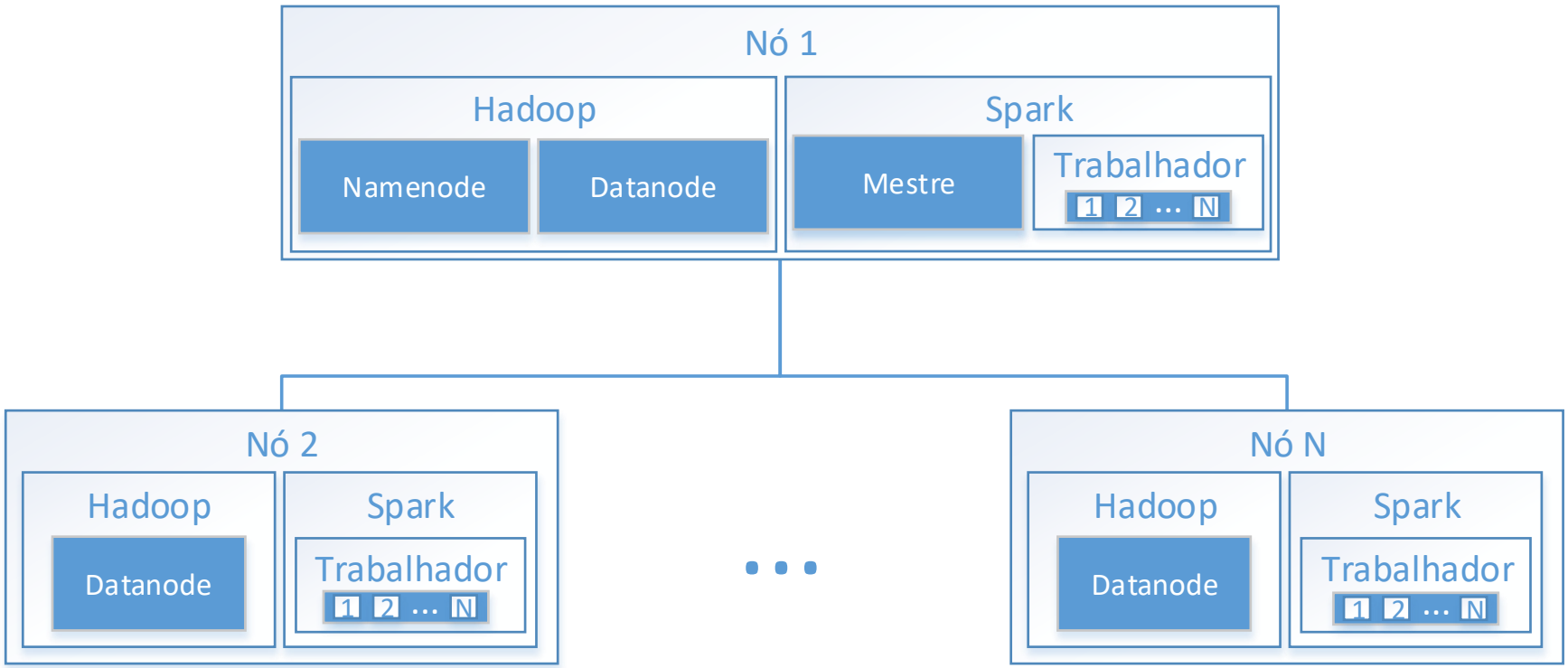
EQUIVALÊNCIAS

Operação `dplyr`	**Operação `sparklyr`**
distinct	unique
sort	sdf_sort
gsub	regexp_replace
rbind	union_all
grepl	rlike
separate	ft_regex_tokenizer + sdf_separate_column

VALIDAÇÃO

- Conjunto de dados pequeno (835 MB).
- Objetivo foi validar se o resultado da versão modificada era equivalente ao da original.
- Realizada por tabela.
- Diversos agrupamentos realizados em suas colunas.
- Comparação de resultados entre ambas as execuções.

PROJETO EXPERIMENTAL



Organização de ferramentas nos nós.

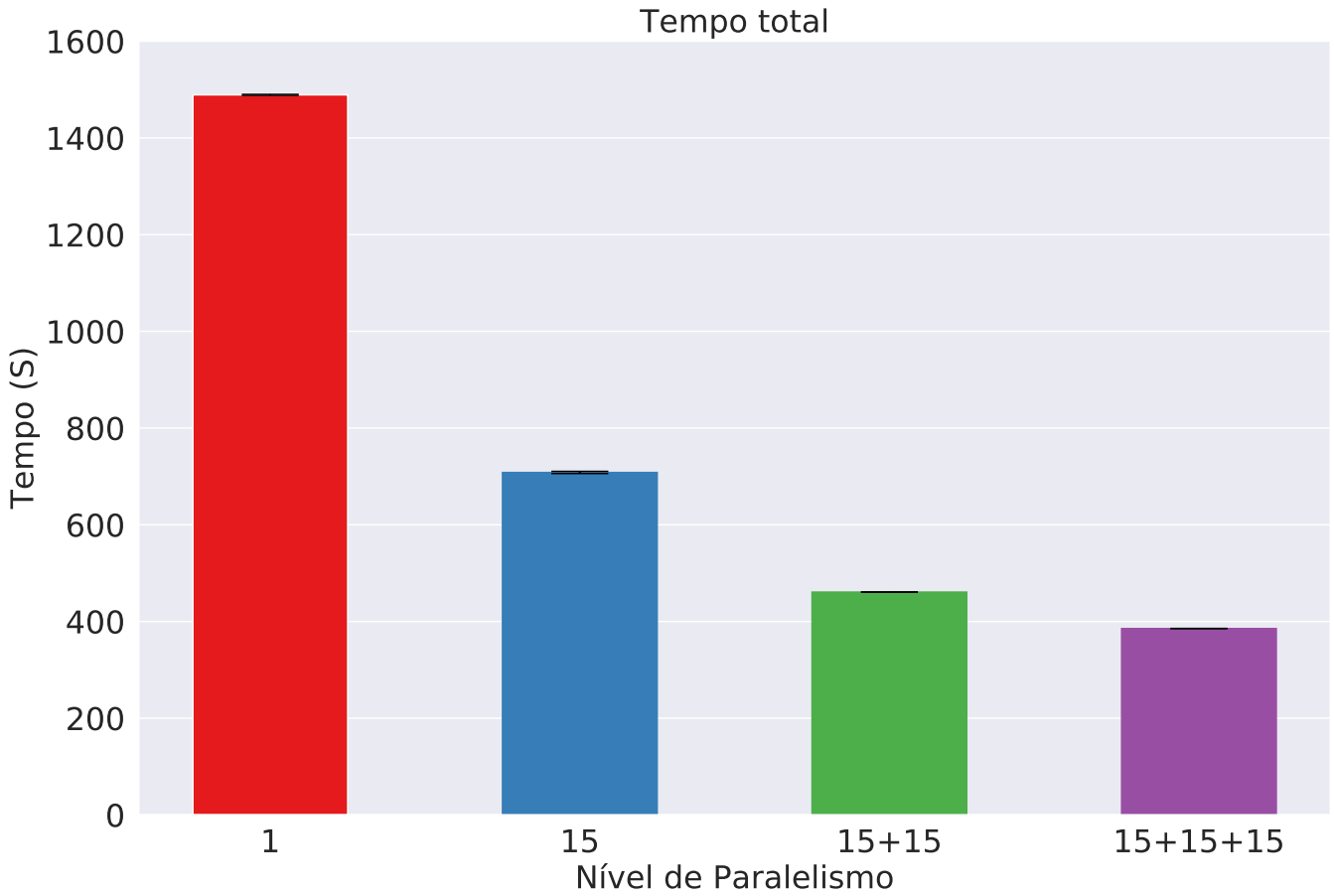
AMBIENTE E METODOLOGIA

Parâmetro	**Configuração**
Processador	2 x Intel Xeon, 2,5 GHz
Núcleos Físicos	16
Núcleos Lógicos	32
Memória	64 GB DDR3
Rede	10 Gigabit

- Testes com 1 nó com versão sequencial.
- Testes com 1, 2 e 3 nós, instanciando 15 executores por nó na versão distribuída.
- ~30 repetições por teste.
- Entrada de 12 GB.

RESULTADOS

- Speedups de respectivamente 2,10x, 3,23x e 3,86x em relação a aplicação original.



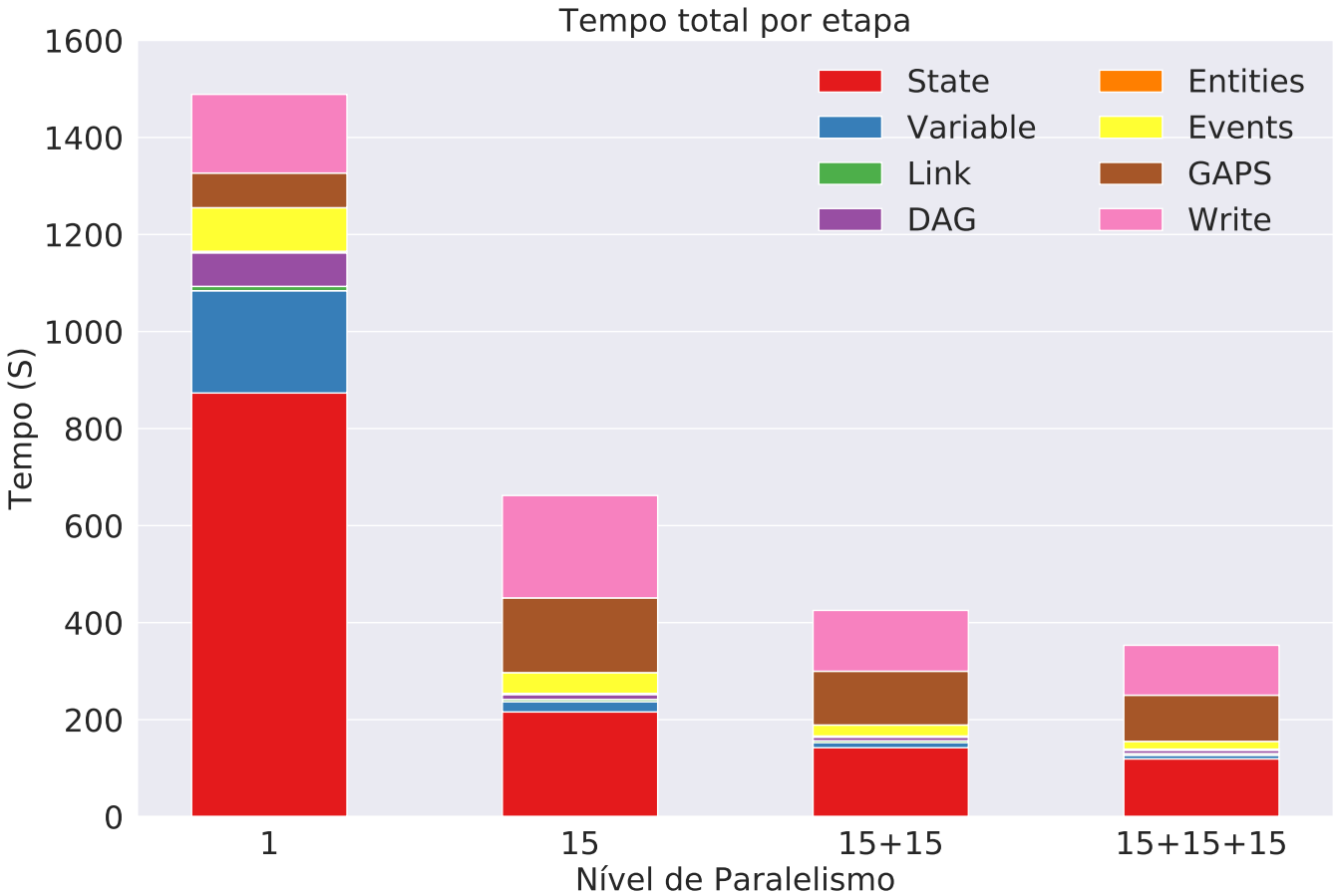
Teste de escalabilidade.

RESULTADOS

Etapa	**1**	**15**	**15+15**	**15+15+15**
State	873.31	215.93	142.15	119.20
Variable	210.77	21.17	10.44	7.50
Link	8.93	4.59	3.84	3.53
DAG	69.14	10.10	7.58	6.76
Entities	3.07	2.42	2.31	2.30
Events	89.87	42.68	22.65	15.91
GAPS	71.51	154.03	110.83	95.22
Write	162.19	211.06	125.40	102.87

- Quatro grupos identificados nos resultados por etapas.

RESULTADOS



Teste de escalabilidade.

- Quatro grupos identificados nos resultados por etapas.

REALIZAÇÕES

- Apenas com as equivalências entre `dplyr` e `sparklyr`, foi possível atingir *speedups* totais de:
  - 2,10x com 1 nó / 15 executores;
  - 3,23x com 2 nós / 30 executores;
  - 3,86x com 3 nós / 45 executores.
- Portanto, conseguimos otimizar a etapa mais custosa do arcabouço StarVZ.
- Vale salientar que com o formato de execução distribuído, o StarVZ é capaz de processar mais dados do que o tamanho de memória de uma única máquina.

TRABALHOS FUTUROS

- Testes com com volume ainda maior de dados.
- Testes com arquivos de registro de outras aplicações.
- Incorporação ao pacote StarVZ.
- Otimização de demais etapas da fase de pré-processamento do StarVZ.

OBRIGADO! PERGUNTAS?

Alexandre Miyazaki
Orientador: Lucas Mello Schnorr
Instituto de Informática — UFRGS