



OTIMIZANDO STARVZ PARA CARGA DE GRANDES VOLUMES DE DADOS

Alexandre Miyazaki

Orientador: Lucas Schnorr

Especialização em Big Data & Data Science — Setembro
de 2019

AGENDA

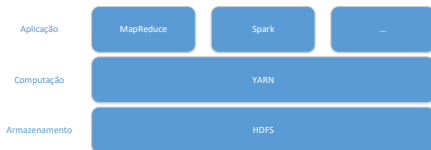
1. Introdução
2. Ferramentas
3. Implementação
4. Avaliação e Resultados
5. Conclusão

- 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.

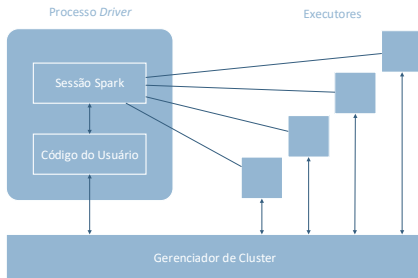
- Em estudos anteriores, analisou-se logs com 18 GB.
- A primeira fase do StarVZ levou cerca de 32 minutos para processar.
- Tal desempenho pode inviabilizar o uso do StarVZ para cargas de grandes volumes de dados.

- Viabilizar o uso do StarVZ para análise de grandes volumes de dados. Para isso, iniciamos pela otimização da parte mais onerosa, o fluxo de processamento R.
- Isto deverá ser atingido utilizando ferramentas de processamento de grandes volumes de dados, como o Hadoop e o Spark.

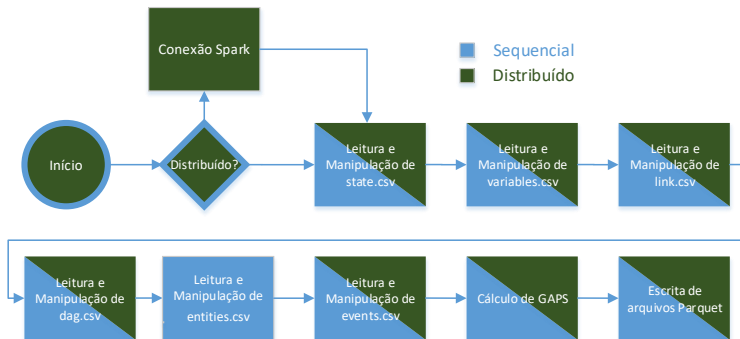
- Arcabouço que permite o processamento de grandes volumes de dados.
- Organizado em camadas.



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

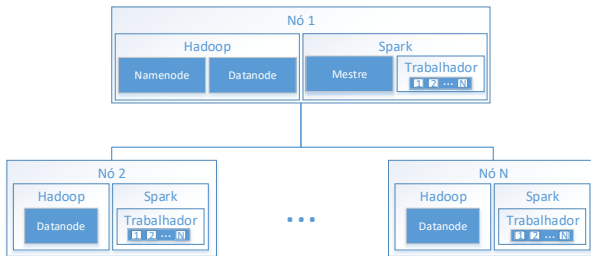


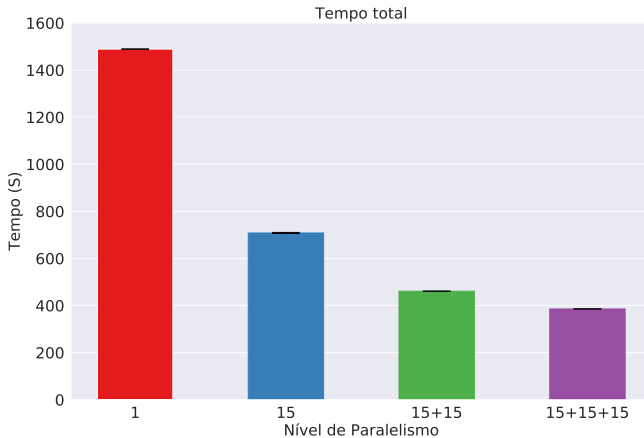
- Otimizar processamento de tabelas.
- Utilizar equivalências entre `sparklyr` e `dplyr`, pois o fluxo de processamento do StarVZ é implementado em R.
- Processo foi facilitado pois a `sparklyr` é baseada na `dplyr`.
- Validação realizada com um conjunto de dados pequeno (835 MB), comparação de desempenho realizada com uma carga de trabalho maior (12 GB).



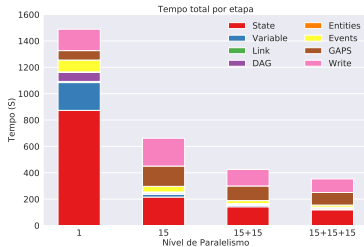
Operação dplyr	Operação sparklyr
distinct	unique
sort	sdf_sort
gsub	regexp_replace
rbind	union_all
grepl	rlike
separate	ft_regex_tokenizer + sdf_separate_column

- Realizada por tabela.
- Diversos agrupamentos realizados em suas colunas.
- Comparação de resultados entre execução sequencial e distribuída.





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	110.83	110.83	95.22
Write	162.19	211.06	125.40	102.87



- Viabilizar o uso do StarVZ para análise de grandes volumes de dados. Para isso, iniciamos pela otimização da parte mais onerosa, o fluxo de processamento R.
- Isto deverá ser atingido utilizando ferramentas de processamento de grandes volumes de dados, como o Hadoop e o Spark.

- 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 StaVZ.
- 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.

- Testes com grandes volumes de dados.
- Testes com arquivos de registro de outras aplicações.
- Acabamento ao pacote StarVZ.
- Otimização de demais etapas da fase de pré-processamento do StarVZ.

OBRIGADO!

Alexandre Miyazaki

Orientador: Lucas Schnorr

Instituto de Informática — UFRGS

`inf.ufrgs.br/~bmenegola`

