Sequenciamento de DNA utilizando Hadoop (MR) (Java Vs. C++)

RAFFAEL BOTTOLI SCHEMMER

UFRGS - PPGC - POD 2014/I

Sumário

- Introdução.
- Motivação e Objetivos.
- Sequenciamento de DNA.
- Modelo Map Reduce.
- Implementação.
- Funcionamento da aplicação.
- Resultados.
- Conclusões e trabalhos futuros.

Introdução.

- Forte interesse do mercado por computação intensiva em dados.
- Cluster favorece o modelo de persistência distribuído:
  - Suporte a tolerância a falhas.
  - Suporte a leitura e escrita paralela de informação.
  - Alta disponibilidade de persistência.
- Sequenciamento de DNA:
  - Alto volume de geração de dados.
  - Demanda de computação intensiva em processamento de dados.

Motivação e Objetivos.

- Hadoop:
  - Ferramenta de big data voltada para:
    - Persistência distribuída de informações.
    - Suporte a tolerância a falhas de tempo real.
    - Suporte a processamento distribuído (MR).
  - Implementado em Java:
    - RMI implementa o mecanismo de troca de mensagem do FW.
    - Rede pode se tornar um gargalo dado o modelo MR utilizado.
    - Sockets é suportado através de Hadoop Pipes (C++).

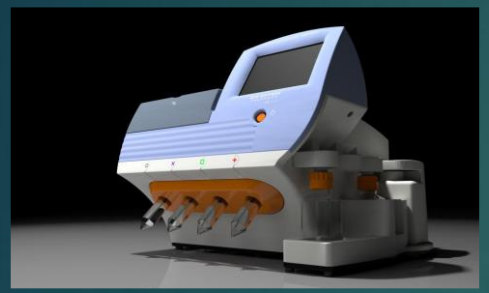
- Autor deste trabalho acredita que:
  - Uso de Hadoop Pipes pode melhorar o tempo de execução e o consumo de recursos da aplicação.

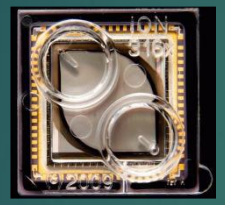
Motivação e Objetivos.

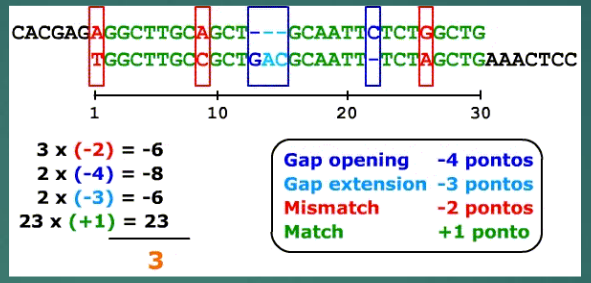
- GPPD:
  - Desenvolve cooperação com HCPA.
  - Desenvolveu um TG que propôs uma aplicação de sequenciamento de DNA usando de MR utilizando o Hadoop.

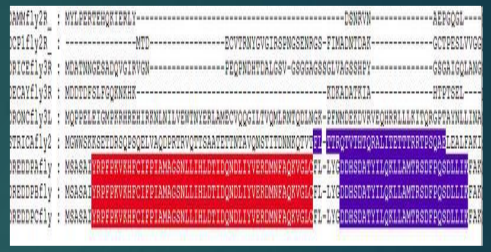
- Este trabalho têm como proposta:
  - Entender o trabalho implementado.
  - Rescrever o código Java em C++ com suporte a Hadoop Pipes.
  - Avaliar ambas as implementações.
  - Uso do cluster GradeP.

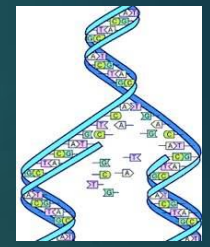
Sequenciamento de DNA.

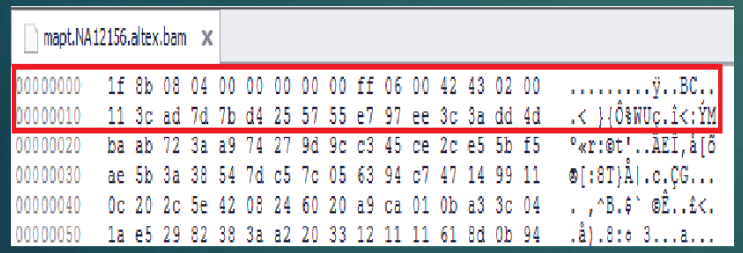


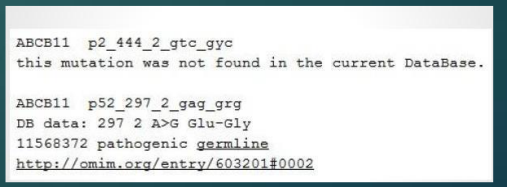










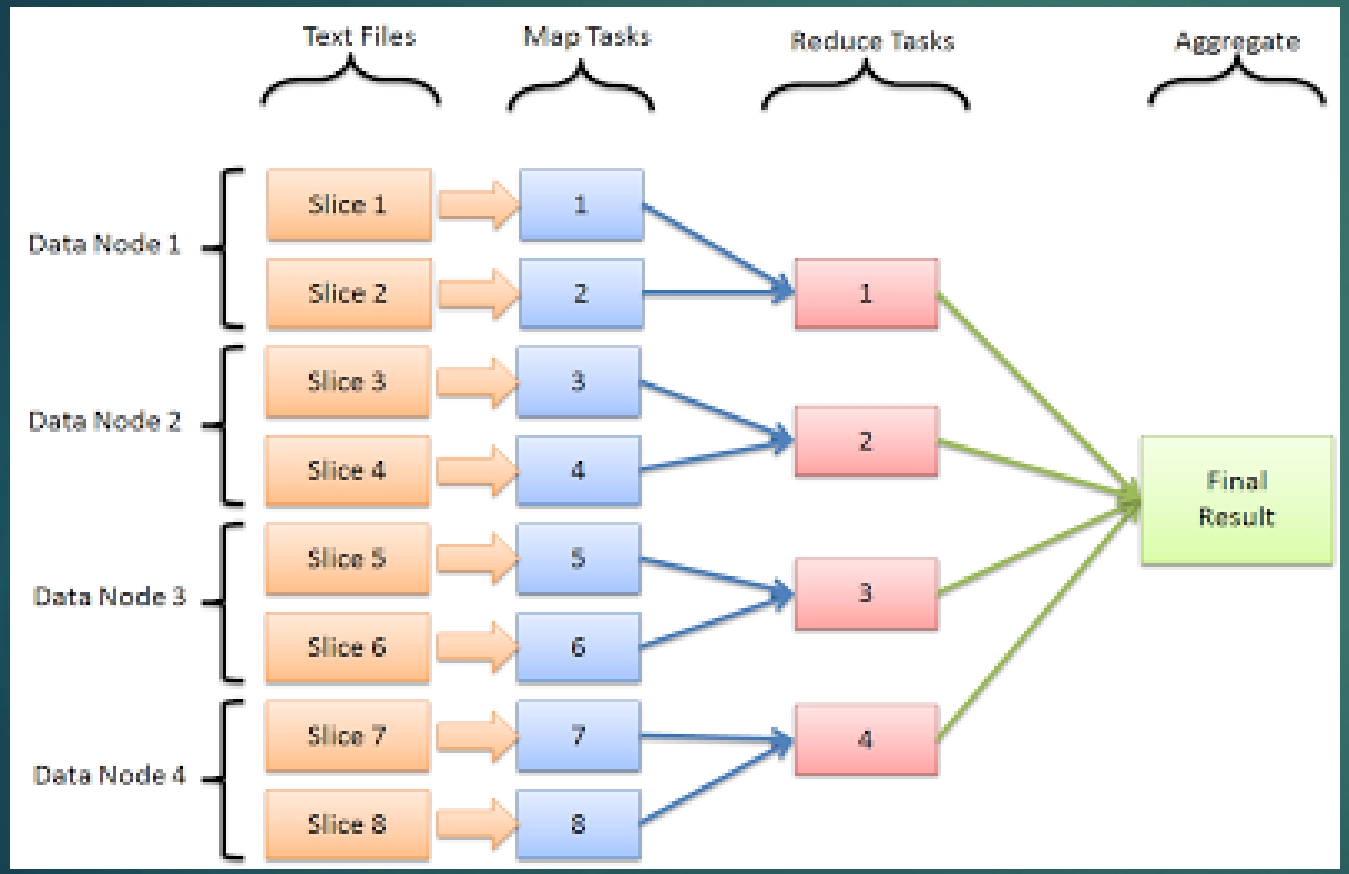


1

2

3

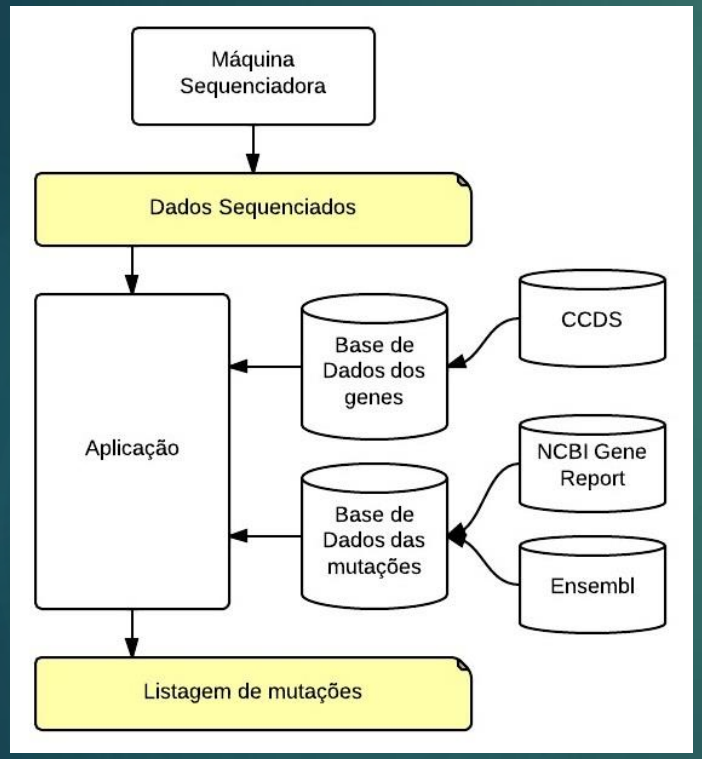
Modelo Map Reduce.

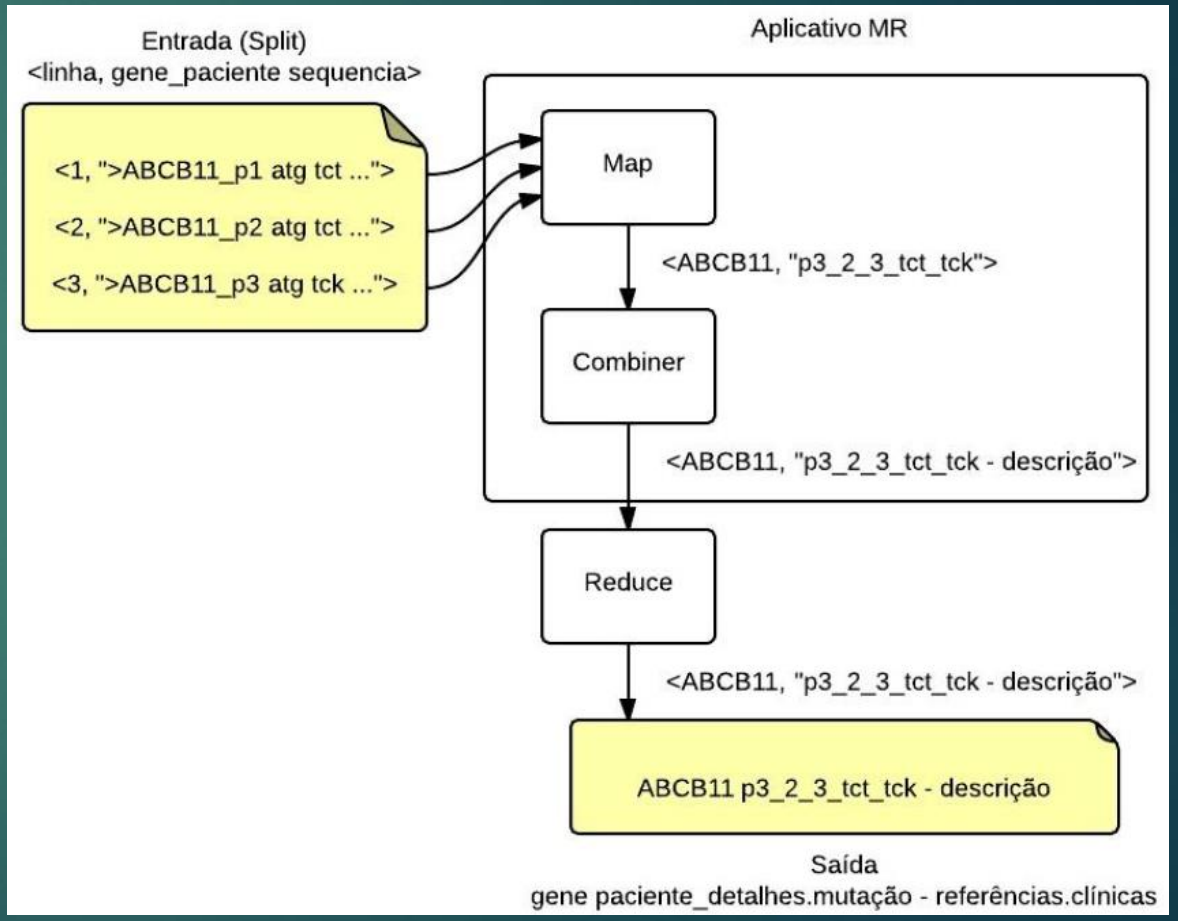


Implementação.

- Dificuldades encontradas:
  - Número de linhas de código: 253 (Java) x 476 (C++) : +88%
  - Tamanho do código executável 6.6KB (Jar) 1.3MB (Bin).
  - Pouca ou nenhuma documentação existente para C++.
  - Poucas as estruturas suportadas em C++ para manipulação de Strings.
  - Código C++ declarou em código as bases de referência:
    - Java utilizou a cache distribuída do Hadoop.
  - Várias bibliotecas auxiliares foram necessárias ao C++.
  - Processo de compilação e execução extremamente complexo.
  - Foi necessário correção e recompilação de bibliotecas (util, pipes).
  - Processo de instalação e configuração do Hadoop é o mesmo que Java.

Funcionamento da aplicação

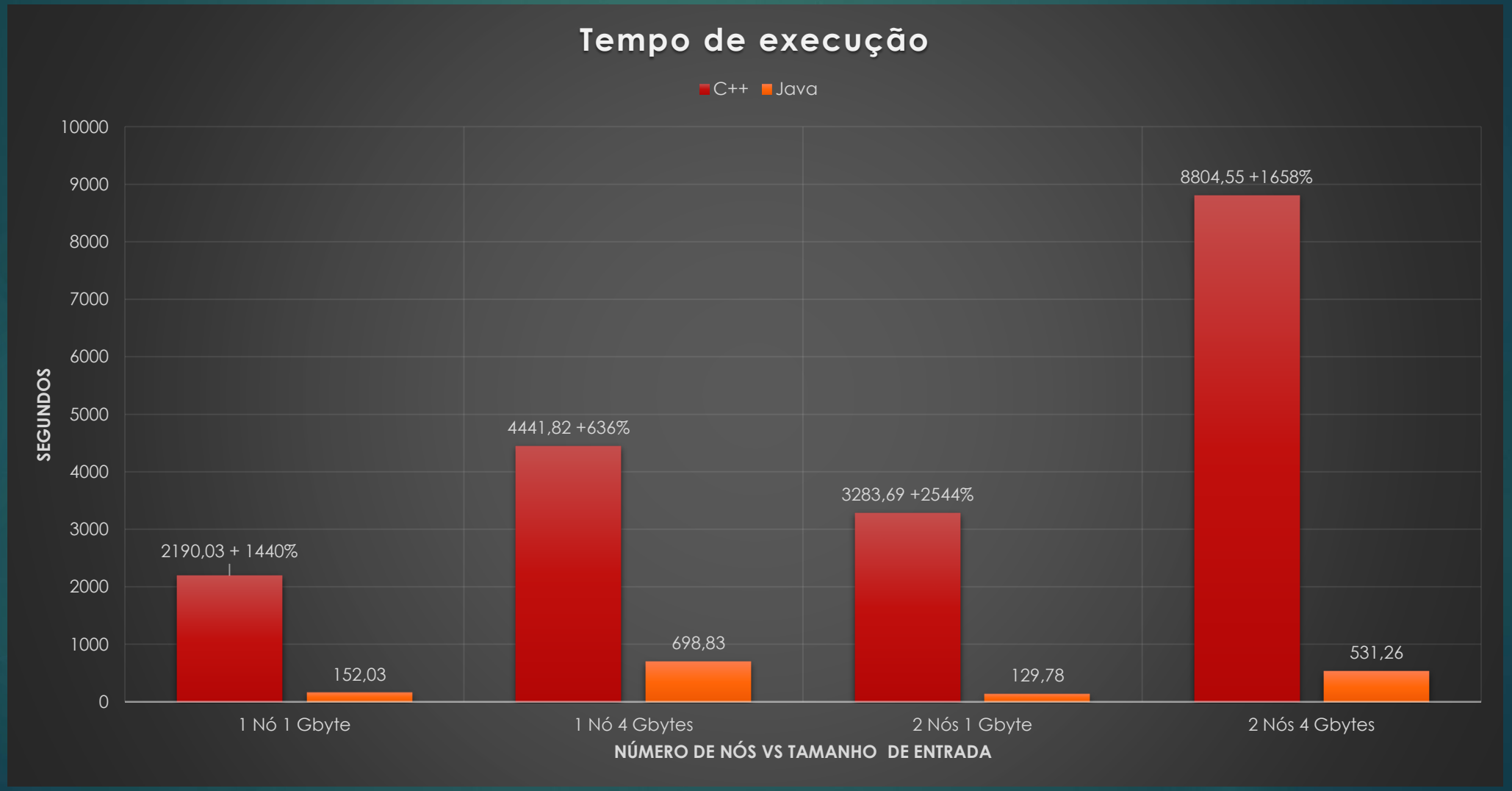




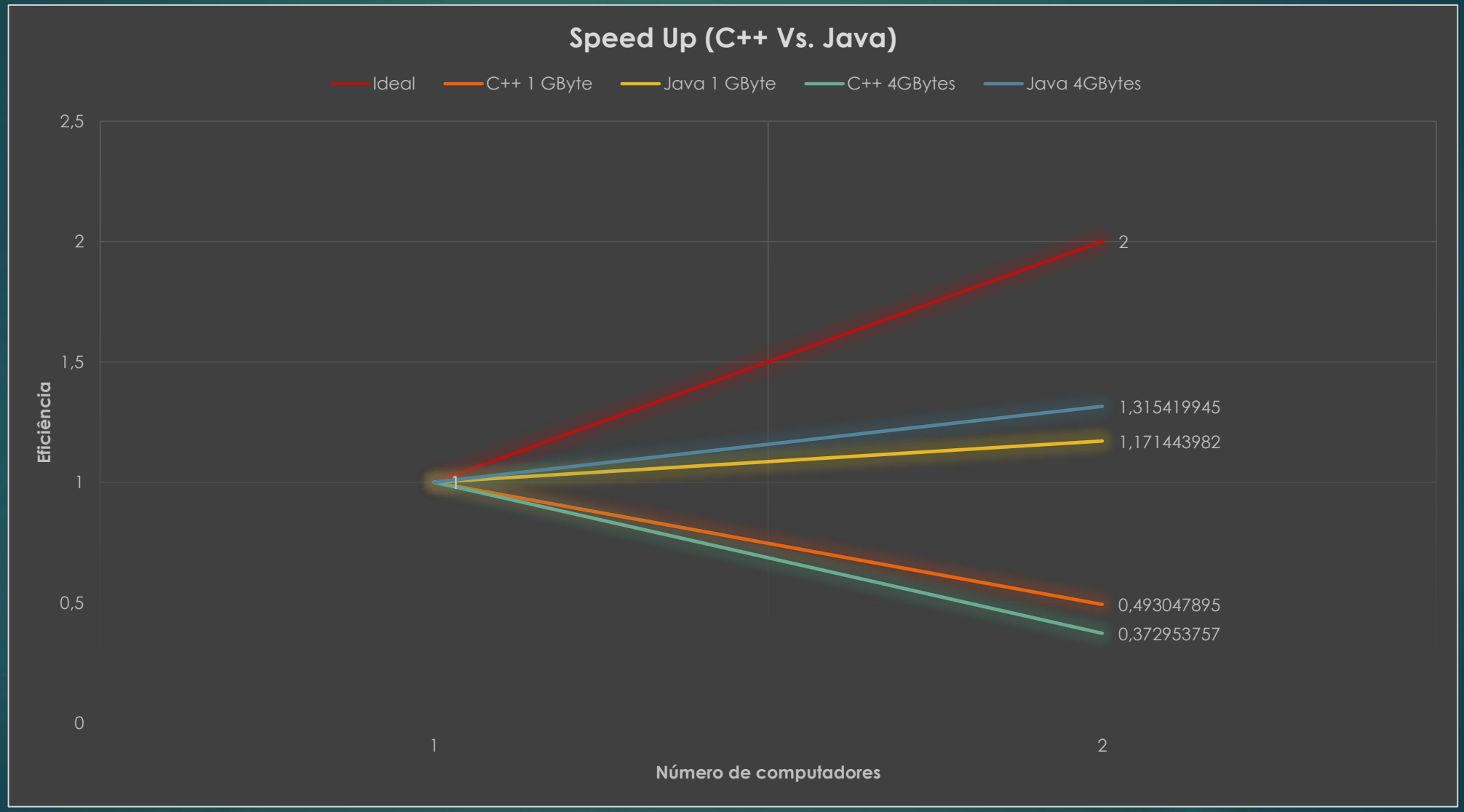
Resultados.

- Cluster utilizado:
  - 4 Máquinas GPPD:
    - 4 Nós SMP (16 Cores):
      - 3 C2Q Q8200 (2.33GHz) (4 CPUs) com 4Gbytes cada.
      - i5 3570 (3.2GHz) (4 CPUs) com 6Gbytes cada.
      - Rede Ethernet 100Mbps (Compartilhada).
      - Discos Sata-I ~40Mbytes (Read/Write).
  - Hadoop 1.0.4.

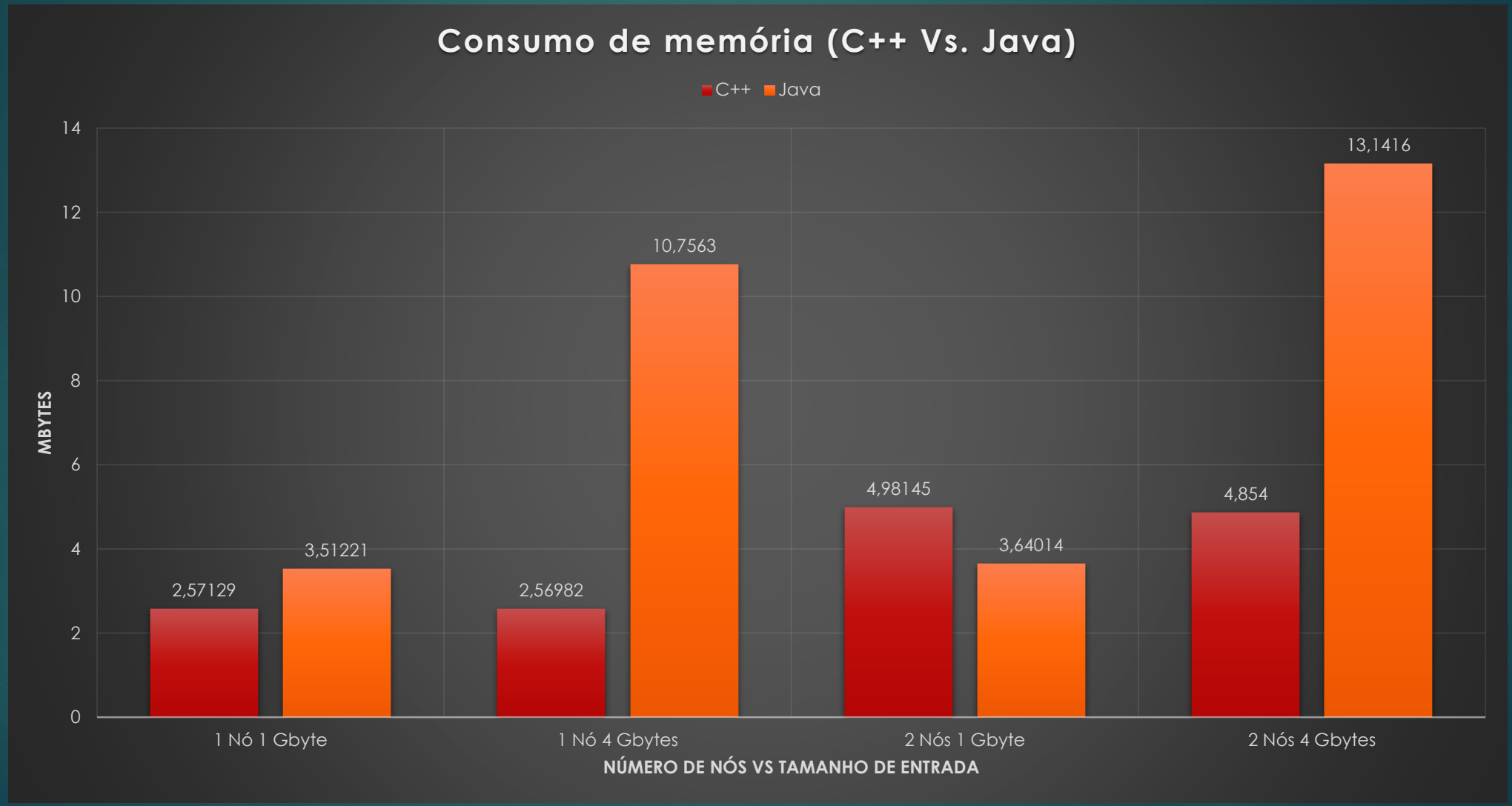
Resultados. (Tempo de execução)



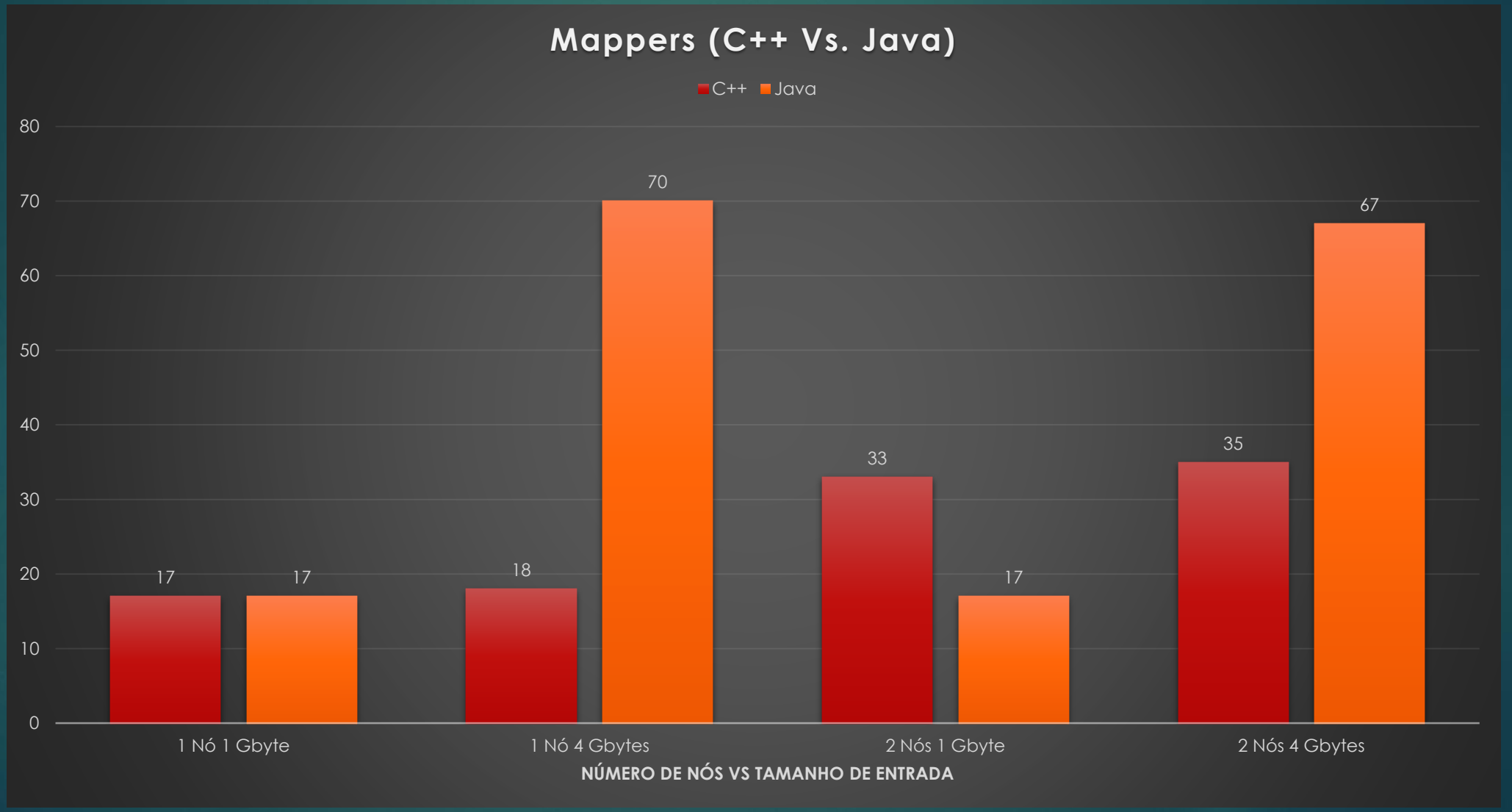
Resultados. (Speed Up)



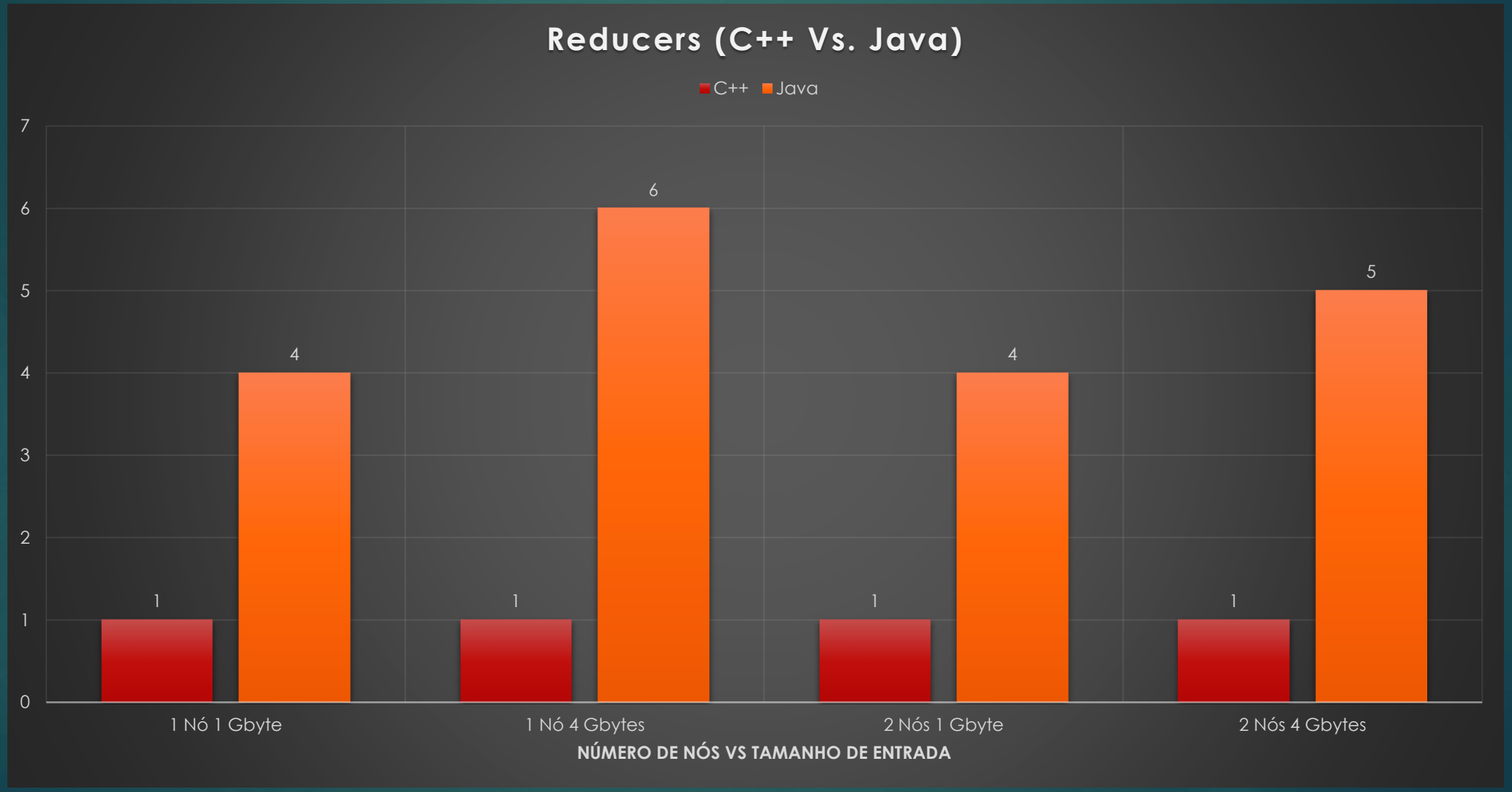
Resultados. (Consumo de Memória)



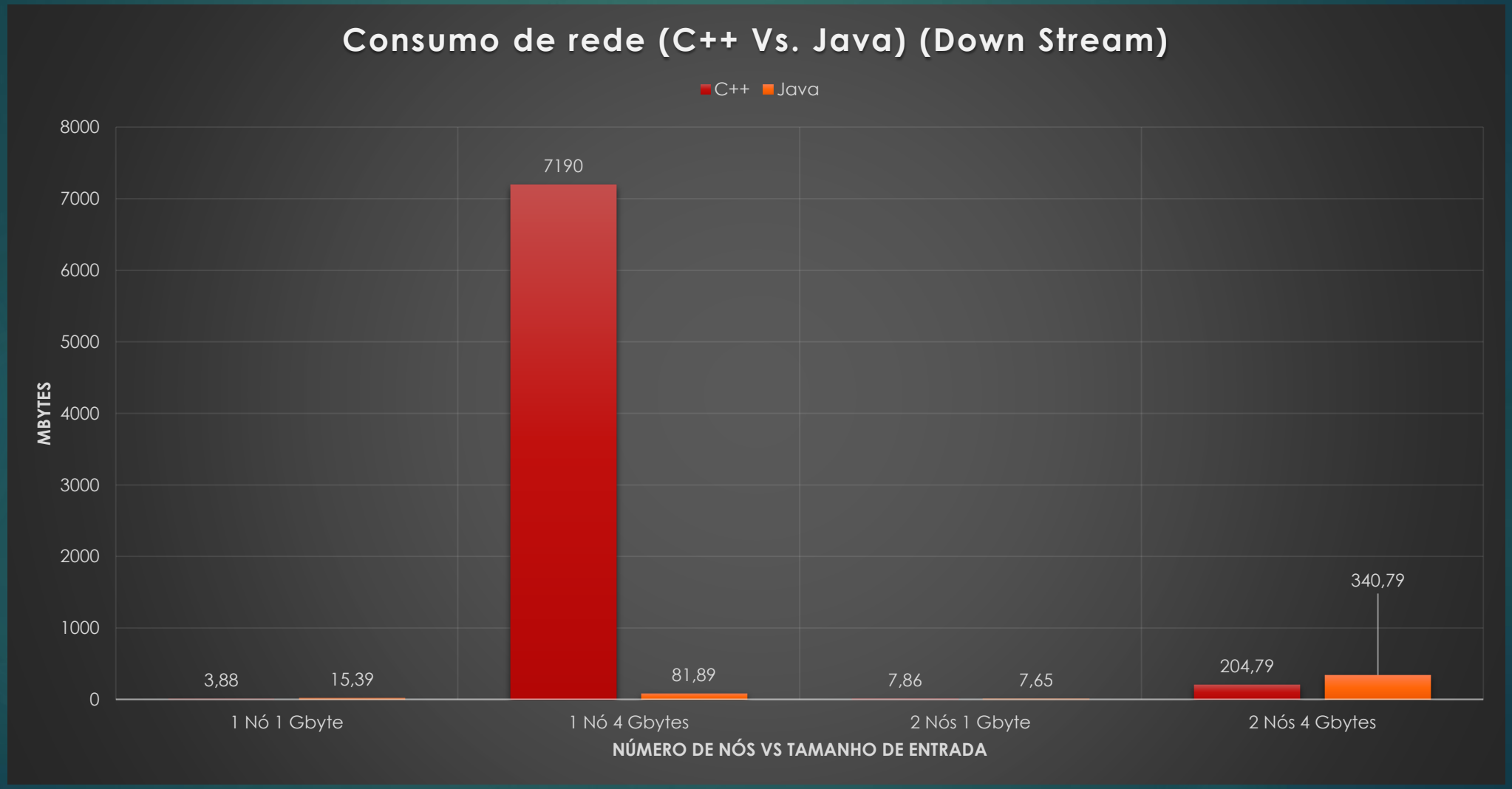
Resultados. (Mappers)



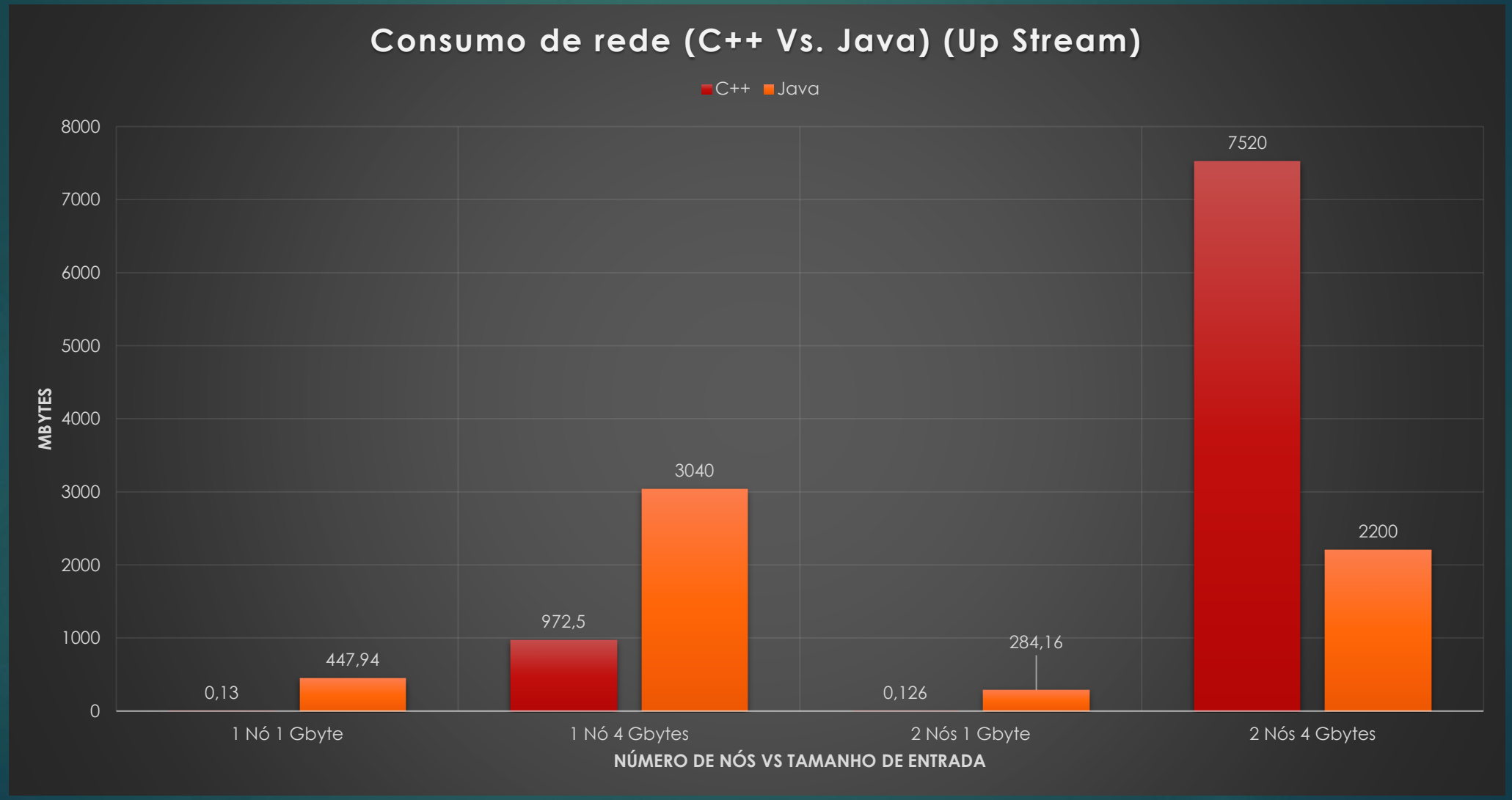
Resultados. (Reducers)



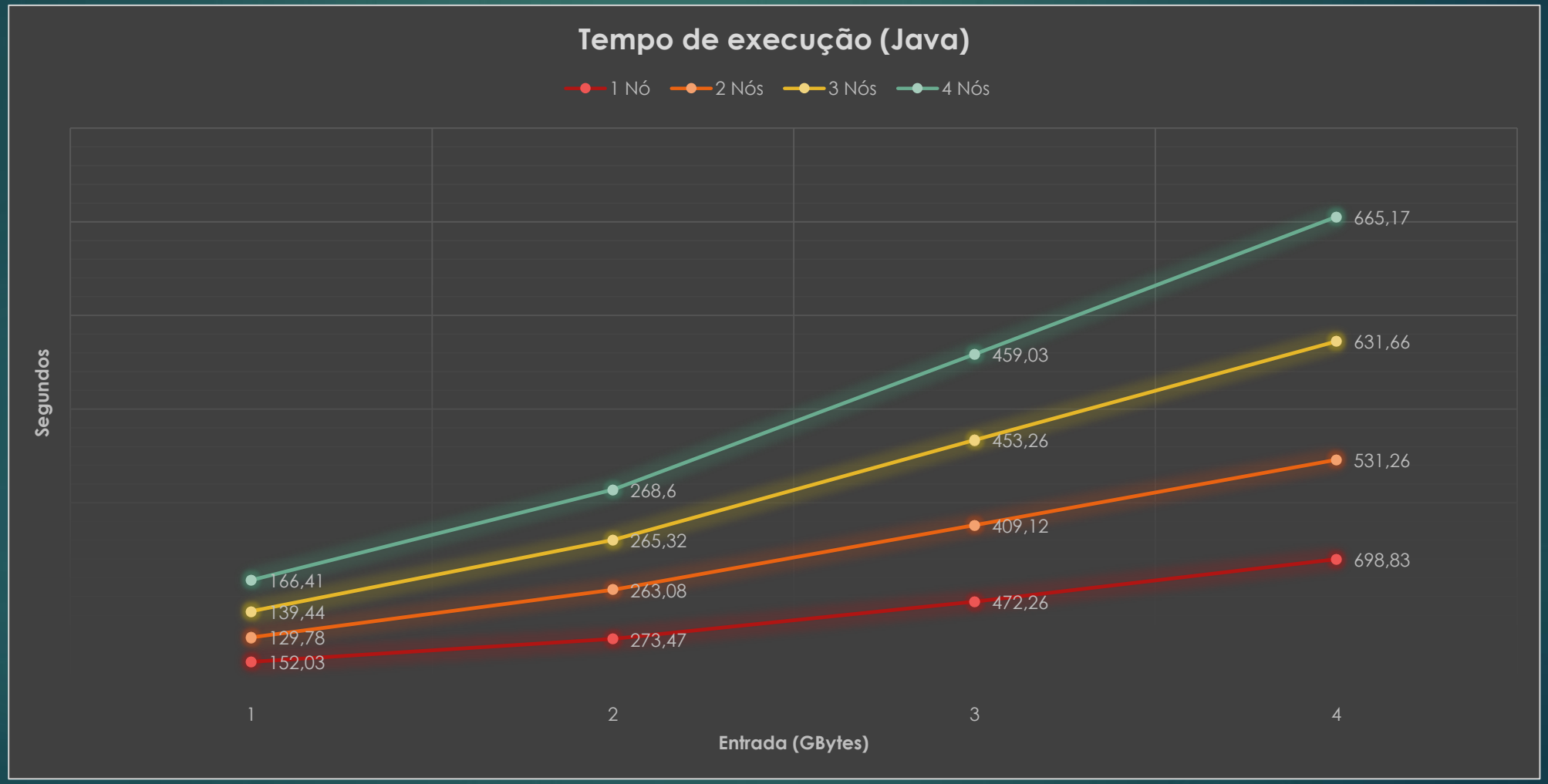
Resultados. (Network Down)



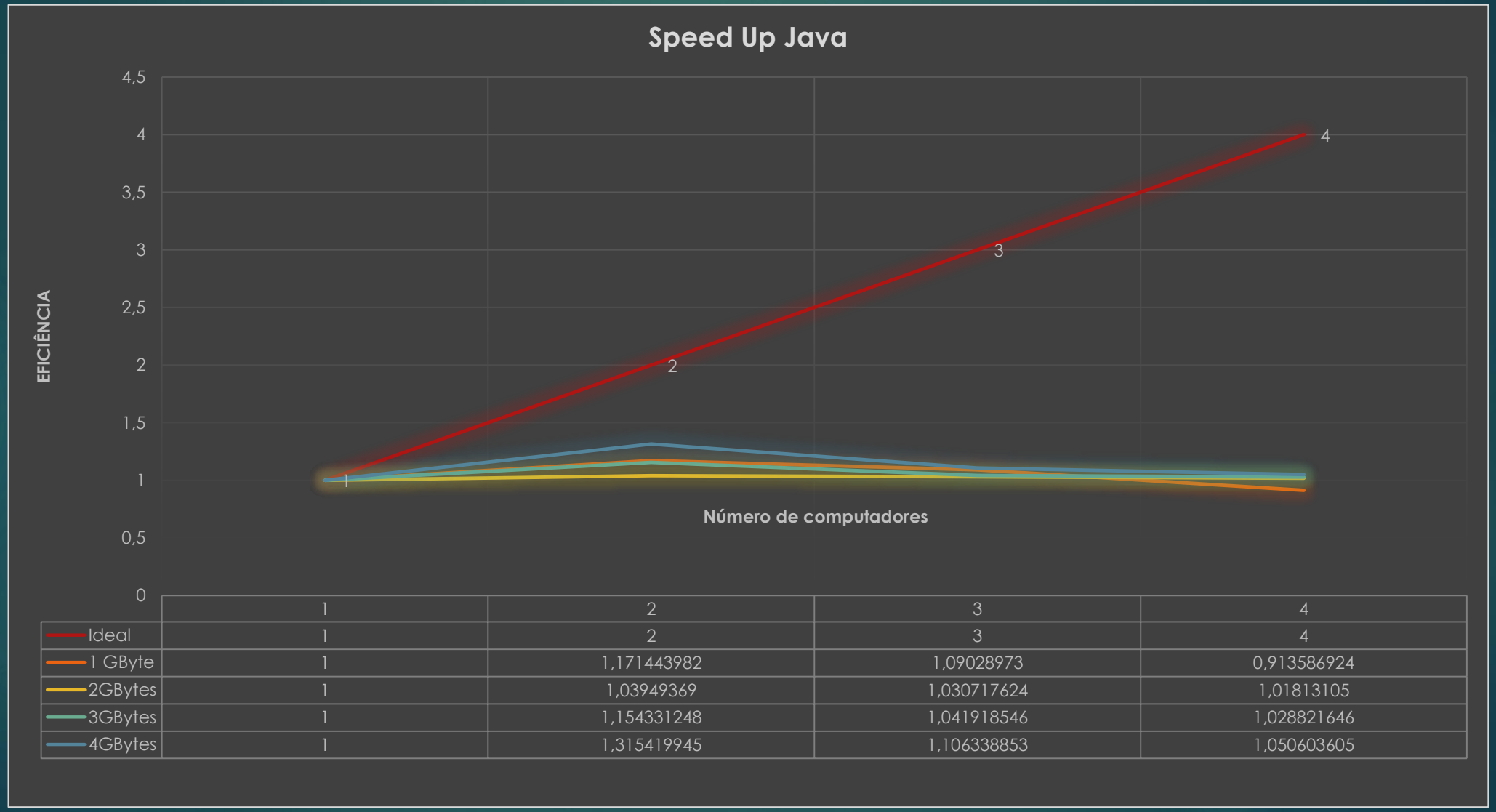
Resultados. (Network Up)



Resultados. (Tempo de execução)

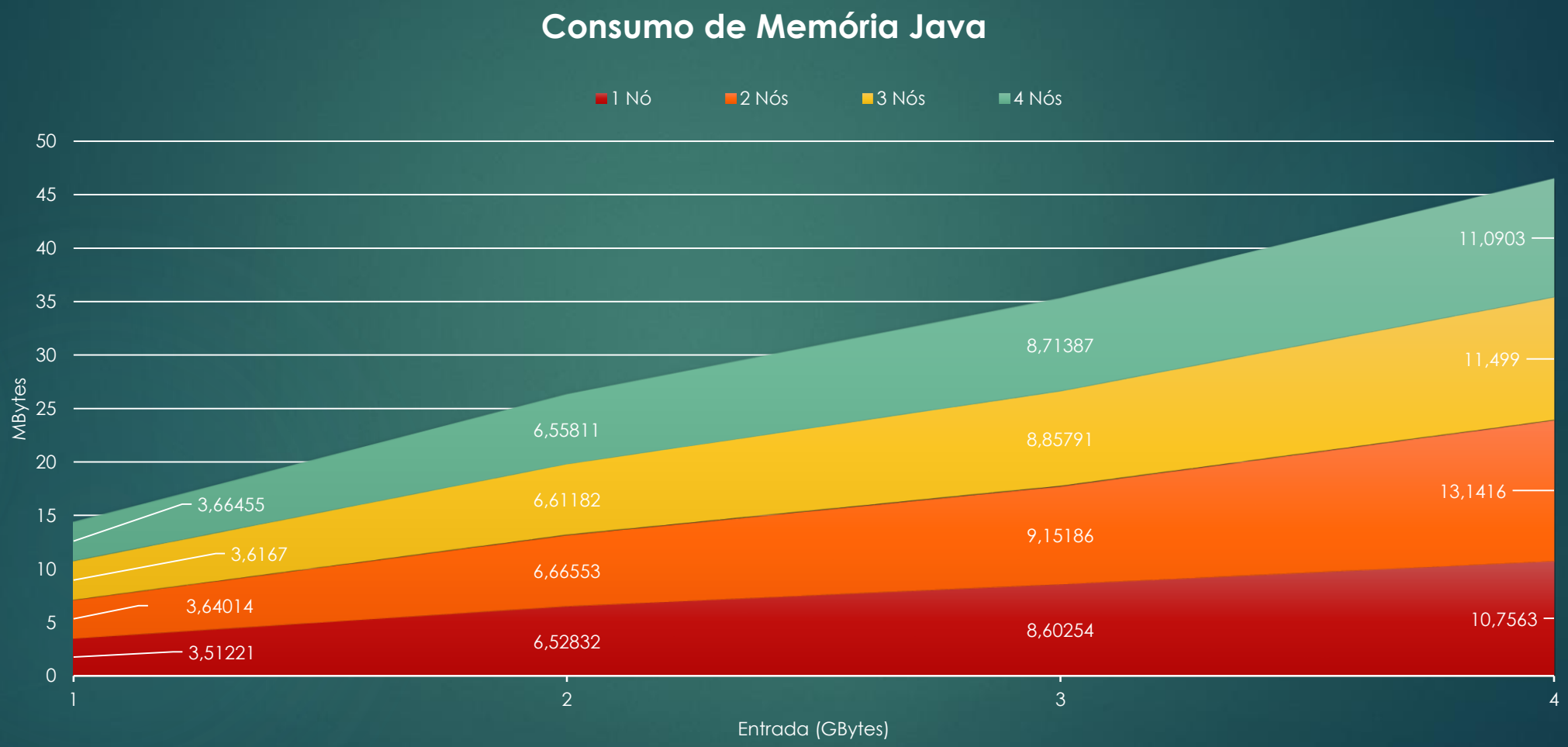


Resultados. (Speed Up)

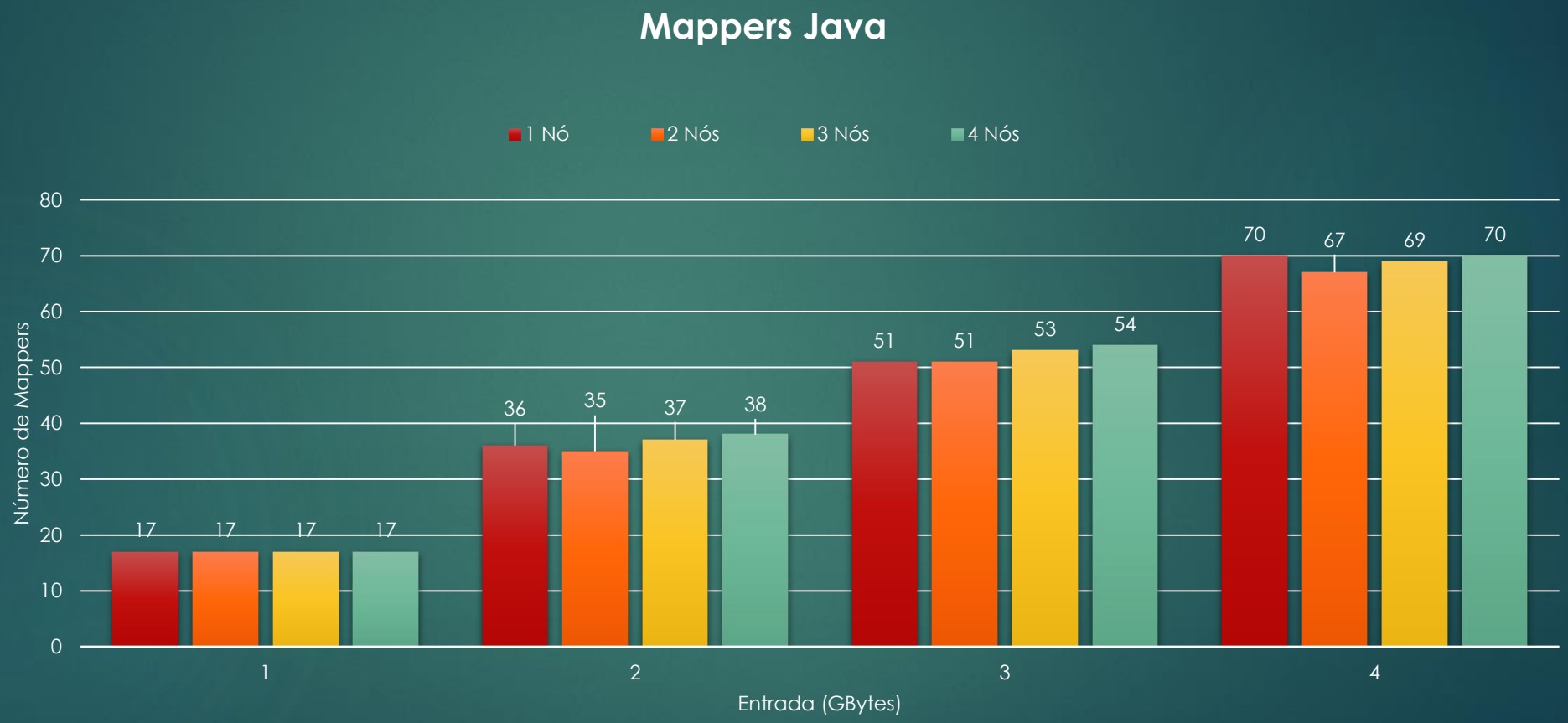


	1	2	3	4
Ideal	1	2	3	4
1 GByte	1	1,171443982	1,09028973	0,913586924
2GBytes	1	1,03949369	1,030717624	1,01813105
3GBytes	1	1,154331248	1,041918546	1,028821646
4GBytes	1	1,315419945	1,106338853	1,050603605

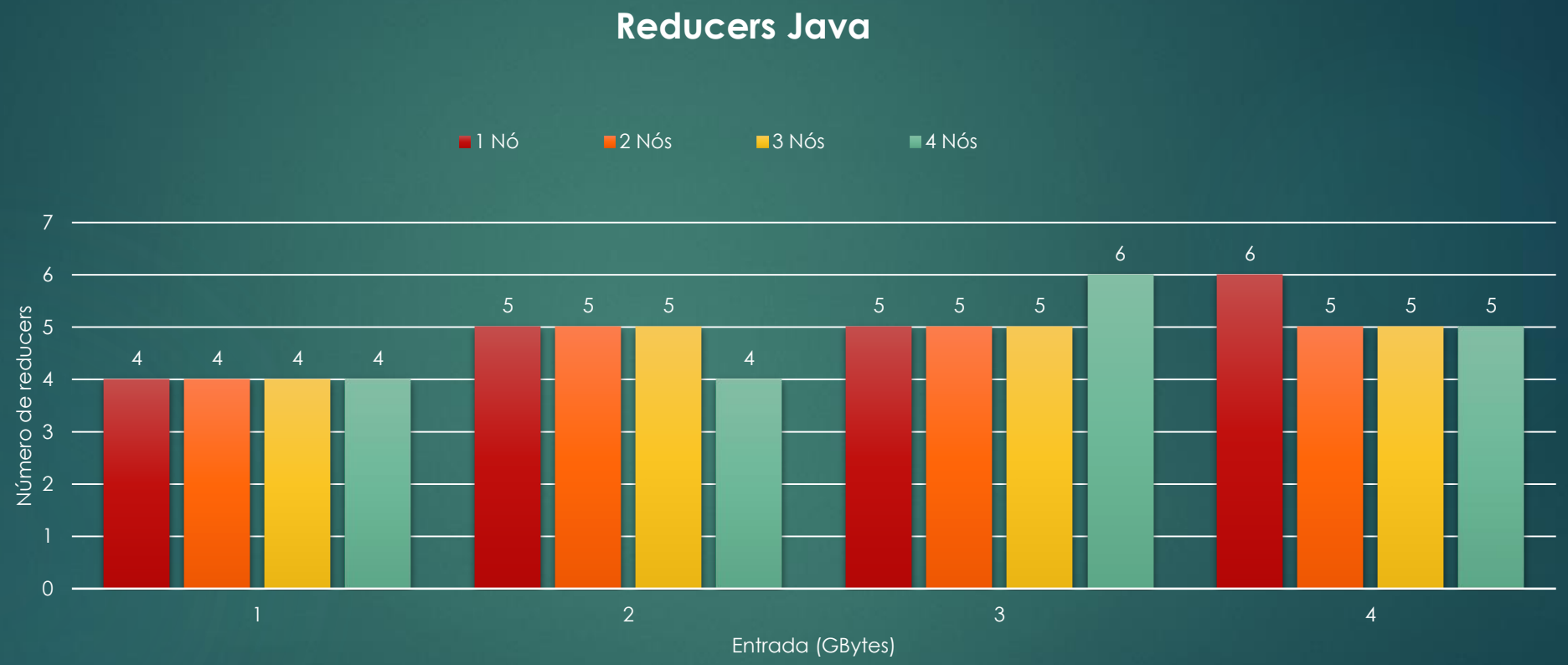
Resultados. (Consumo de Memória)



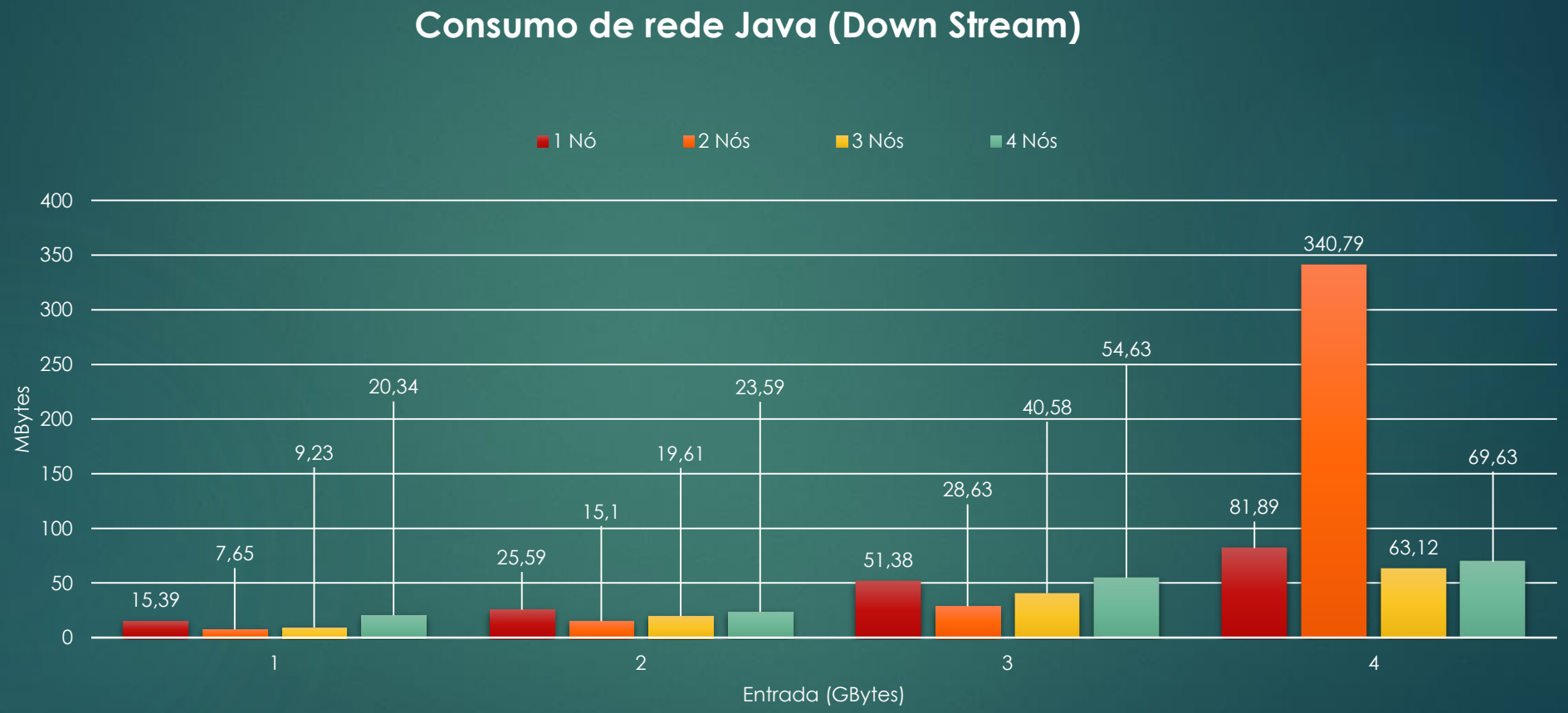
Resultados. (Mappers)



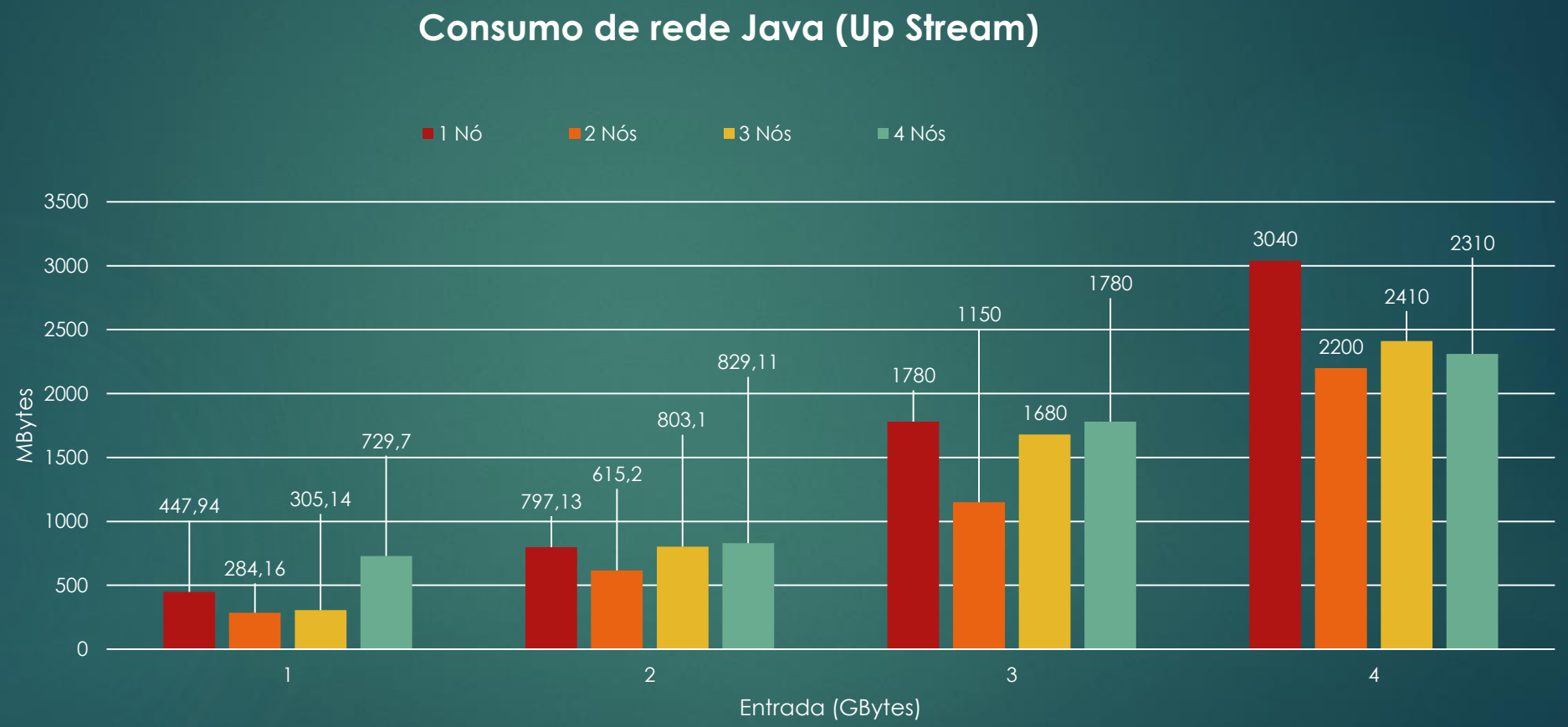
Resultados. (Reducers)



Resultados. (Network Down)



Resultados. (Network Up)



Conclusões e trabalhos futuros.

- Com relação a arquitetura distribuída:
  - Cluster pode não dar suporte a aplicação.
  - Volumes de entrada são custosos para executar/avaliar.
    - Dado o tempo de execução.
  - Configuração de um ambiente distribuído é custoso.
- Com relação ao uso de C++:
  - Resultados demostraram que C++ não apresenta competitividade comparado a Java, tanto no consumo de recursos como no tempo de execução.
  - C++ não apresenta a mesma facilidade de programação e uso do Hadoop quando comparado a Java.
  - C++ limita a portabilidade da aplicação para ambientes heterogêneos.

Conclusões e trabalhos futuros.

- Com relação a aplicação de sequenciamento de DNA:
  - É limitada as bases de referencia.
  - Não dá suporte a busca de outros tipos de genes.
  - Pelas avaliações pode não ser escalável.
  - Amostras sequenciadas podem vir sobre outros formados (BAM?) e não estarem alinhadas.
- Com relação a trabalhos futuros:
  - Detectar sobre quais aspectos o Map Reduce pode ser interessante para a aplicação proposta.
  - Execução deste trabalho no cluster gradeP.
  - Identificar outros aspectos da aplicação a serem utilizados pelo modelo como por exemplo:
    - Processo de alinhamento.
    - Busca nos arquivos de referencia pelos genes a serem sequenciados.
    - Busca nas bases de mutações por genes utilizados durante o sequenciamento.

Perguntas ?