

Mergesort Iterativo Distribuído

MAC0352 Redes de Computadores e Sistemas Distribuídos

Julio Kenji Ueda 9298281
Ricardo Akira Tanaka 9778856

Visão Geral

Propósito

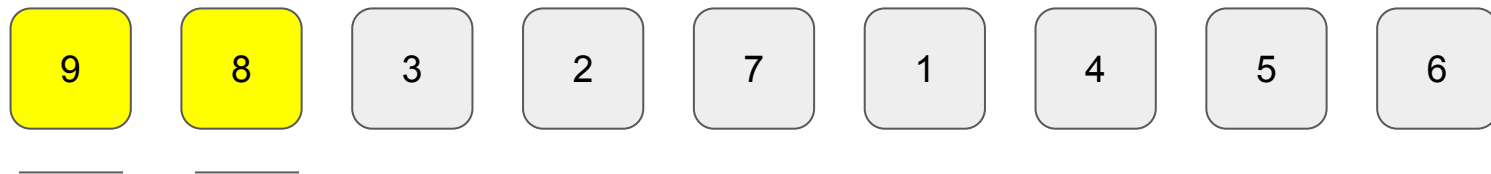
Sistema distribuído que faz a ordenação de uma lista de números inteiros utilizando o algoritmo Mergesort e um protocolo de comunicação próprio.

O que é Mergesort Iterativo:

Mergesort iterativo ordena um vetor intercalando dois blocos consecutivos de tamanho 2^i , $i = 0, 1, 2, 3, \dots$, onde i é o número da iteração.

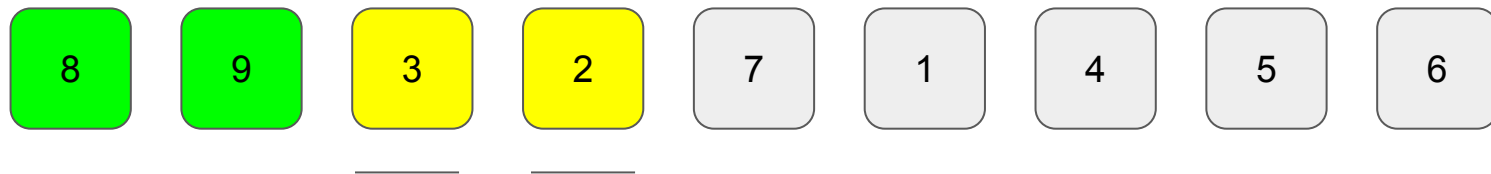
Quando o tamanho do bloco for maior do que o tamanho do vetor, temos o vetor ordenado.

Exemplo:



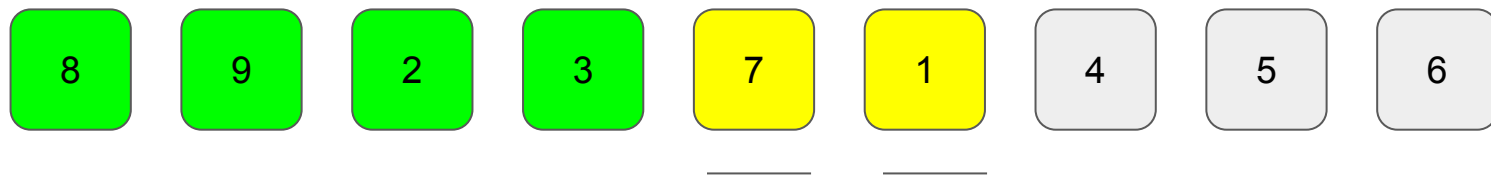
Iteração 0

Exemplo:



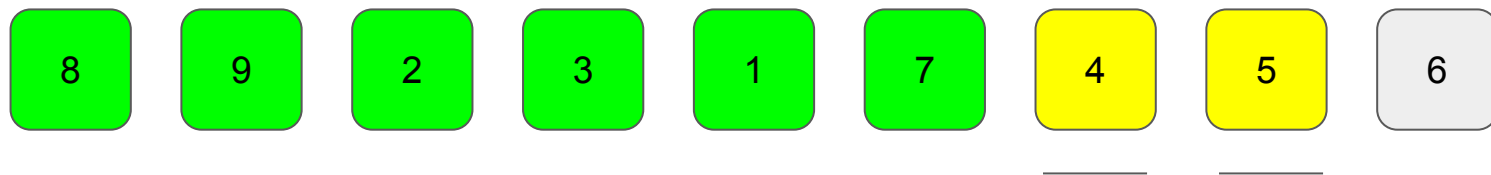
Iteração 0

Exemplo:



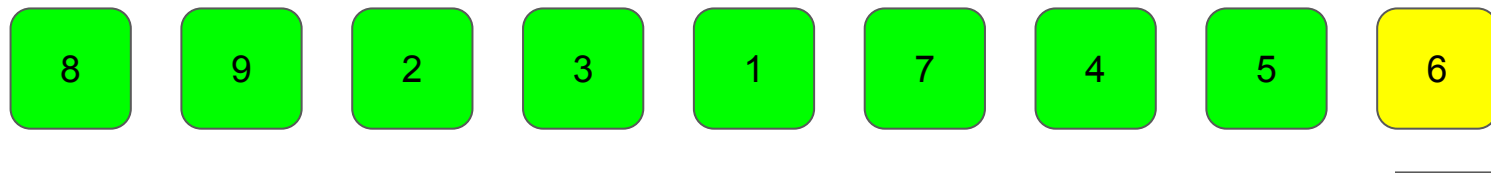
Iteração 0

Exemplo:



Iteração 0

Exemplo:



Iteração 0

Exemplo:



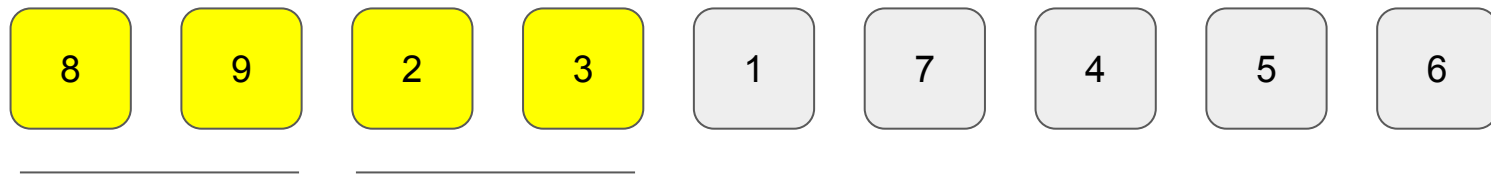
Iteração 0

Exemplo:



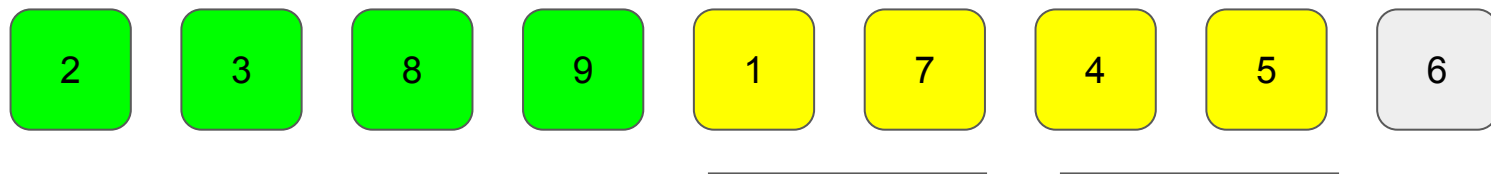
Iteração 1

Exemplo:



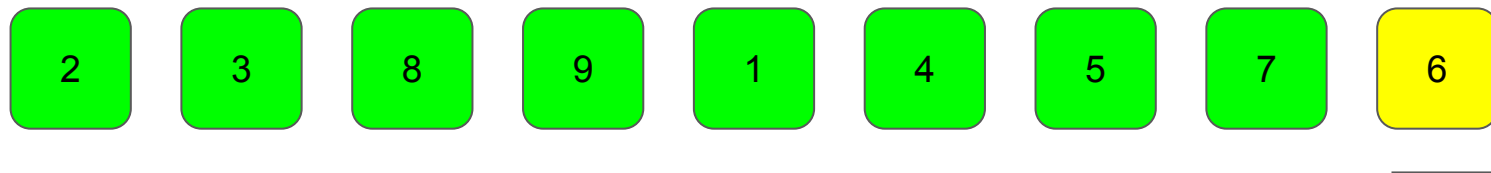
Iteração 1

Exemplo:



Iteração 1

Exemplo:



Iteração 1

Exemplo:



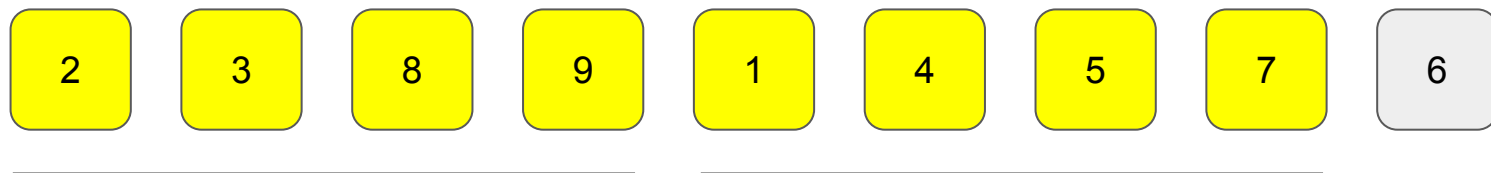
Iteração 1

Exemplo:



Iteração 2

Exemplo:



Iteração 2

Exemplo:



Iteração 2

Exemplo:



Iteração 2

Exemplo:



Iteração 3

Exemplo:



Iteração 3

Exemplo:



Iteração 3

Papéis

Há três papéis importantes para realizar a ordenação da lista:

First: papel realizado apenas pelo primeiro computador que inicia a operação e é responsável pela entrega e recebimento das tarefas apenas com o Leader.

Leader: papel único realizado por qualquer computador e é responsável pela divisão das tarefas com os *Workers*.

Worker: papel realizado por todos os computadores e é responsável pela execução da tarefa.

Os três papéis são desempenhados de forma paralela utilizando *Threads*.

First

Características

- Possui comunicação apenas com o *Leader*.
- Entrega tarefas e recebe resultados quando requisitado.
- Decide o momento da troca de líder.
- Procura 5 vezes pelo *Leader* se perder a conexão.
- Se o *Leader* não existir, ele se torna um.

Leader

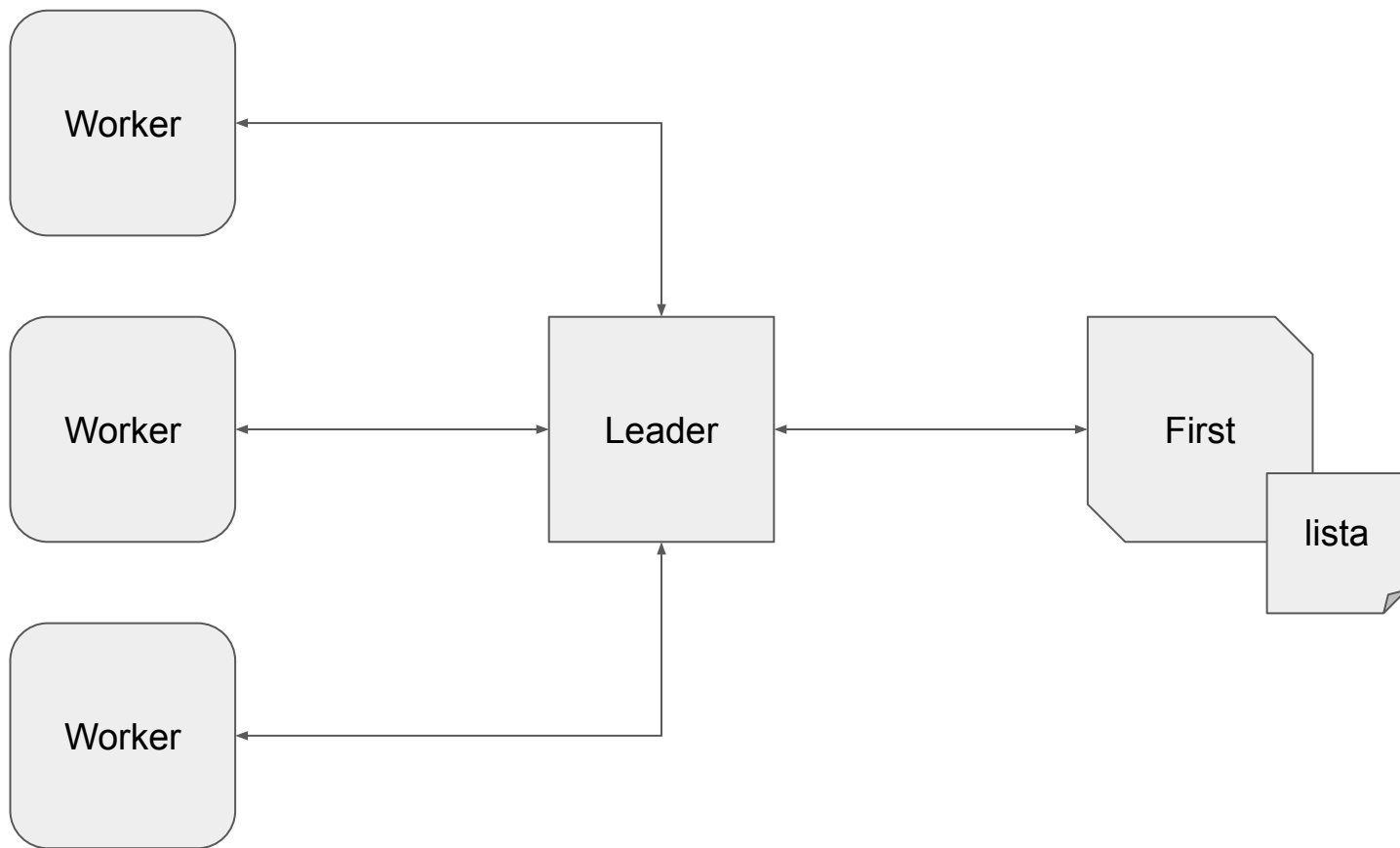
Características:

- Possui comunicação com o *First* e os *Workers*.
- Responde pedido de tarefa dos *Workers*.
- Responde pedido de envio de resultados dos *Workers*.
- Pede tarefa ao *First*.
- Entrega resultado ao *First*.
- Ordena os pedidos dos *Workers* em fila (justiça)
- Verifica e adiciona novas conexões (periodicamente).
- Verifica e remove conexões perdidas (periodicamente).

Workers

Características:

- Possui comunicação apenas com o *Leader*.
- Pede tarefas e entrega resultados.
- Procura 10 vezes pelo *Leader* se perder a conexão.
- Torna-se líder quando escolhido.



Protocolo de Mensagens

Pedido de envio de endereço de escuta

Mensagem enviada pelo Worker e First para o Leader:

Mensagem: 100 <comentário>

Resposta: 200 <comentário>

O endereço IP do Leader vem incluso com a resposta.

Pedido de tarefa

Mensagem enviada pelo Worker para o Leader e o Leader para o First:

Mensagem: 600 <comentário>

Resposta: 700 <comentário> (send_size)

Onde <send_size> é o tamanho do dado a ser transferido.

Início da transferência de dados

Mensagem enviada pelo lado que vai enviar os dados:

Mensagem: 700 <comentário> (send_size)

Resposta: 900 <comentário>

A resposta indica que o lado que vai receber está com buffer pronto para receber os dados de tamanho <send_size>.

Transferência de dados

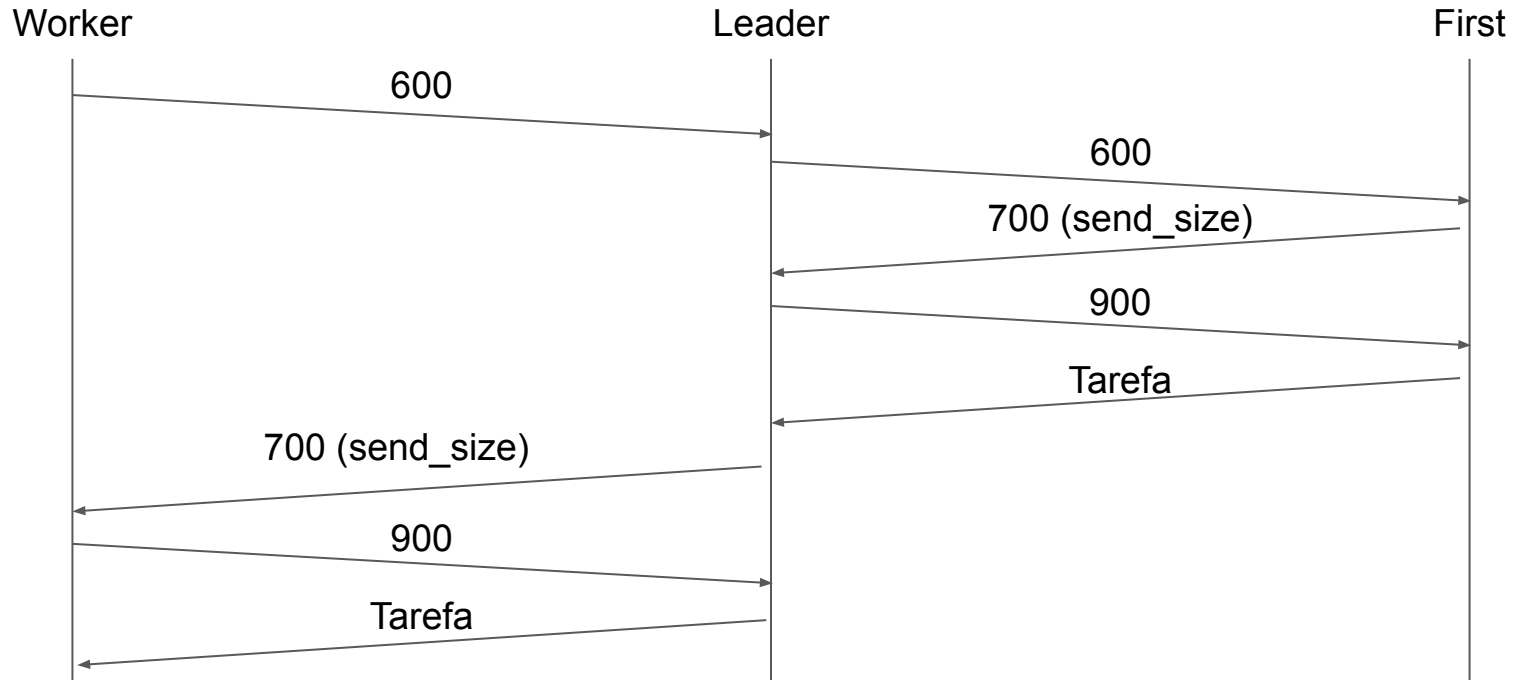
Mensagem enviada pelo lado que vai receber os dados:

Mensagem: 900 <comentário>

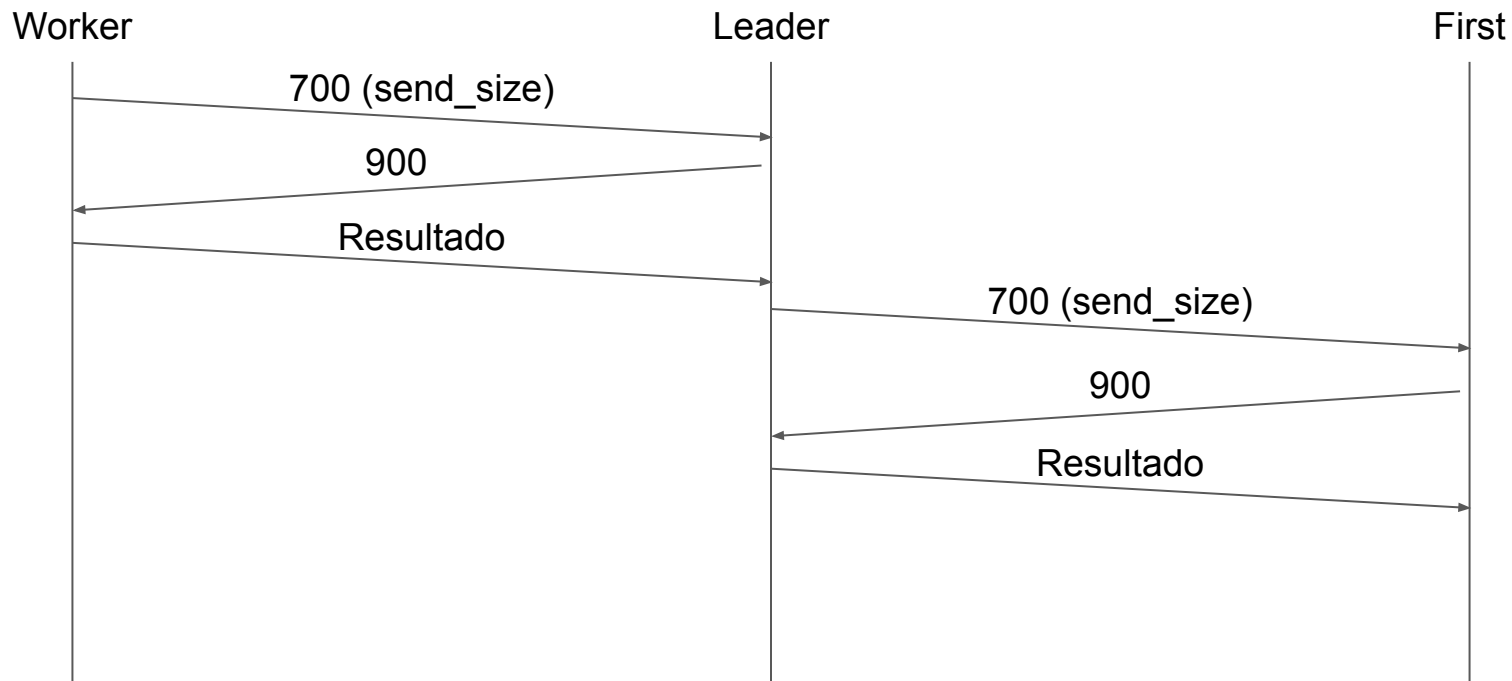
Resposta: O dado em si

Uma vez recebido o tamanho da transferência, o código 900 informa que o ambiente está pronto para a transferência de dados.

Exemplo de requisição de tarefa do Worker



Exemplo de requisição de envio do resultado



Início da eleição de líder

Mensagem enviada pelo First para o Leader:

Mensagem: 400 <comentário>

No fim de uma iteração, o First:

1. Envia um pedido para o início da eleição do novo líder para o Leader.
2. Fecha a conexão com o Leader.

Escolha do líder

Mensagem enviada pelo Leader para o Worker:

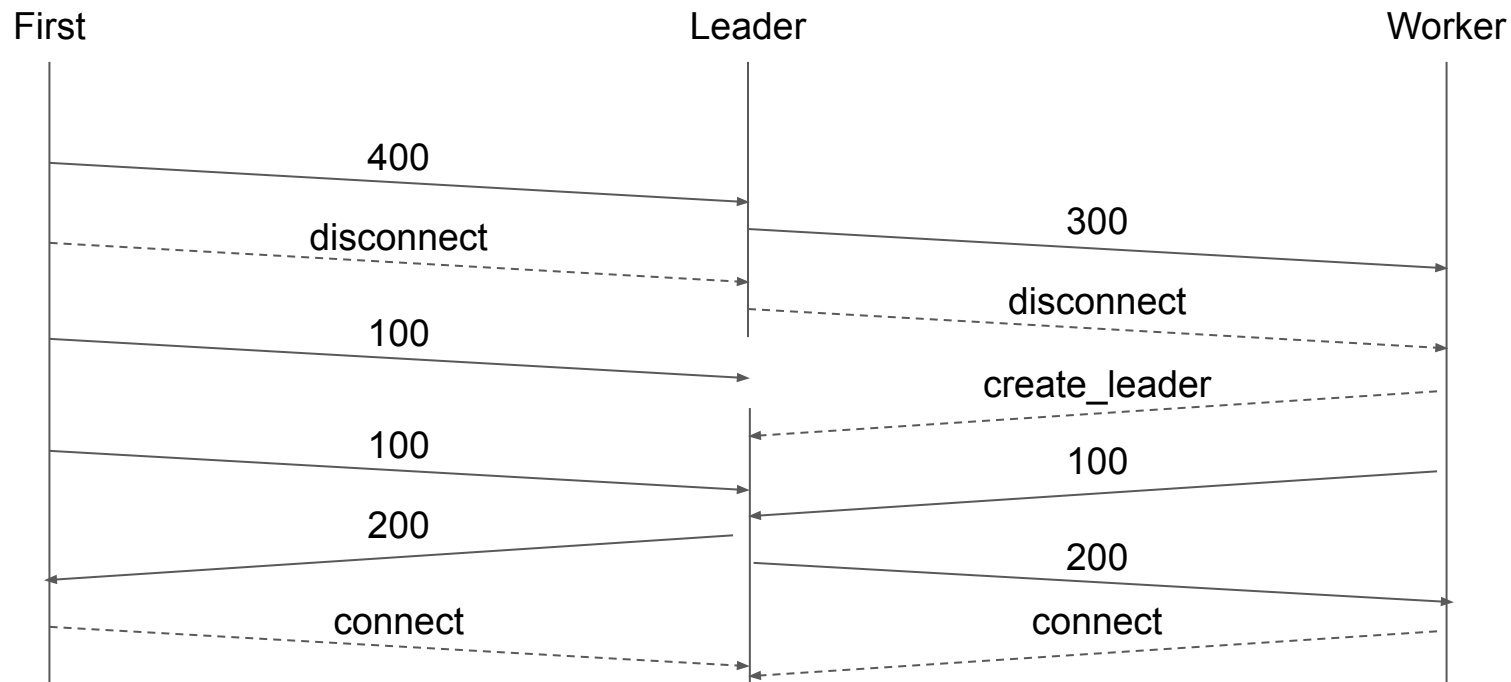
Mensagem: 300 <comentário>

Critério: número de tarefas realizadas por cada Worker. O maior se torna líder.

Após o envio da mensagem para o Worker escolhido:

1. Leader fecha todas as conexões
2. Worker escolhido cria um novo Leader
3. Workers e First realizam a conexão com o novo Leader

Exemplo de eleição do líder e reconexão



Trabalho Concluído

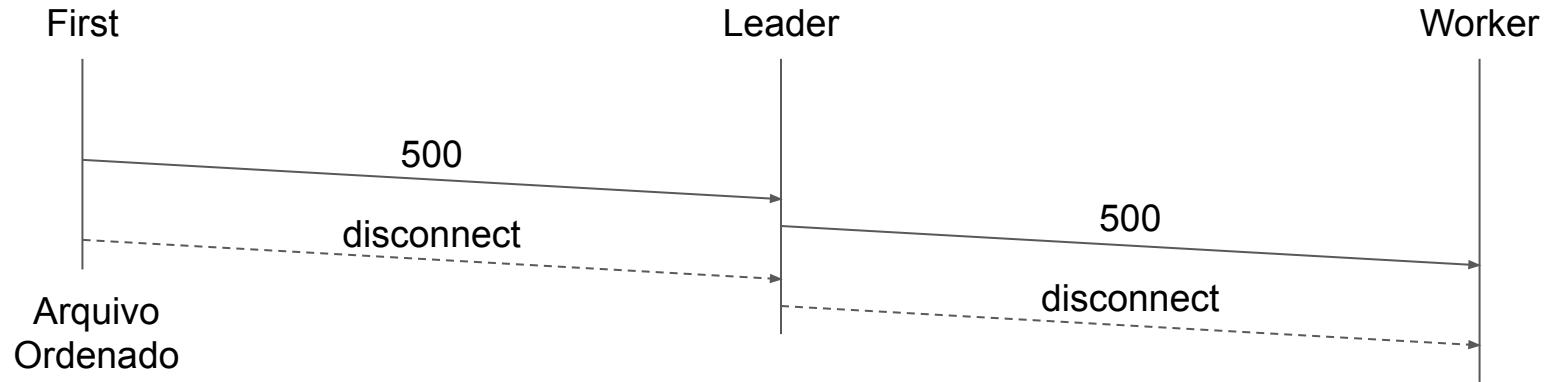
Mensagem enviada pelo First para o Leader e o Leader para os Workers:

Mensagem: 500 <comentário>

Após o término da ordenação da lista:

1. First envia a mensagem para o Leader.
2. First fecha a sua conexão e gera o arquivo com o conteúdo ordenado.
3. Leader envia a mensagem a todos os Workers, e fecha todas as conexões.
4. Workers terminam suas operações.

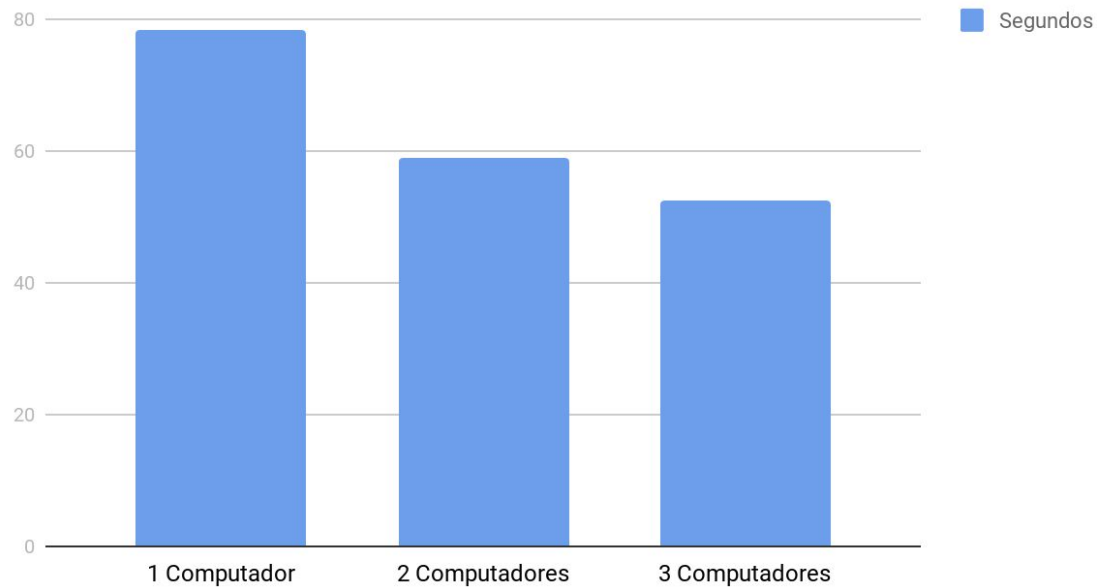
Exemplo de Trabalho Concluído



Análise de Desempenho

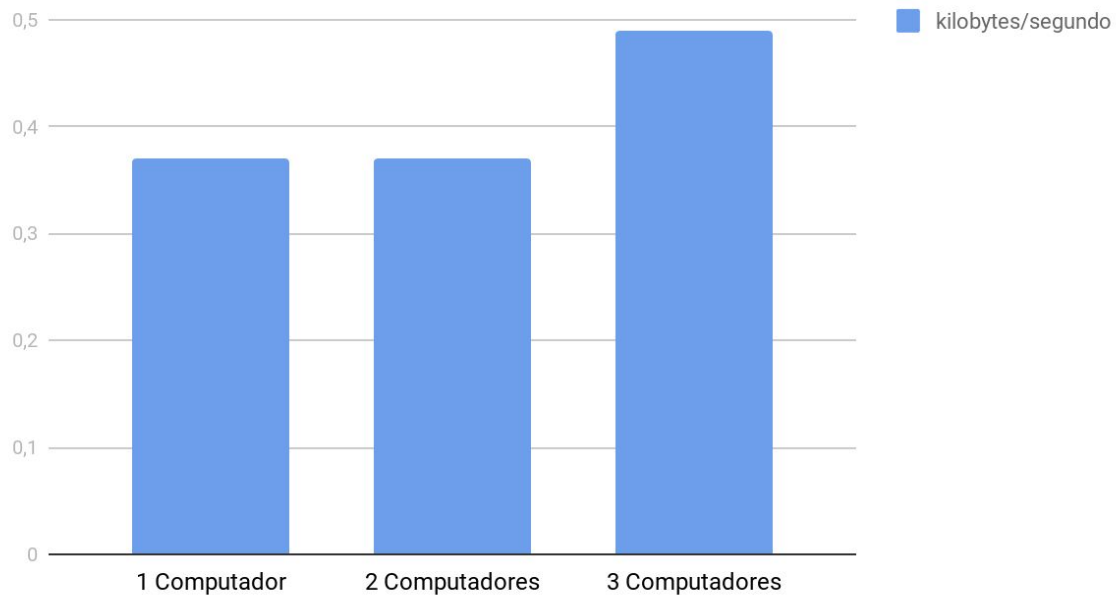
Uso da CPU

Tempo médio de ordenação de 1000 inteiros



Uso da Rede

Uso médio da rede na ordenação de 1000 inteiros



Uso da Rede

O uso médio da rede foi calculado utilizando:

- Programa **ifstat** para medir o uso da rede por segundo.
- Instância limpa do Google Compute Engine com uso mínimo de recursos.
- Containers Docker para simular uma rede de computadores distintos.

Referências

Feofilof, Paulo 2018, *Mergesort*

<<https://www.ime.usp.br/~pf/algoritmos/aulas/mrgsrt.html>>

McMillan, Gordon 2019, *Socket Programming HOWTO*

<<https://docs.python.org/3/howto/sockets.html>>

socket - Low-Level networking interface

<<https://docs.python.org/3/library/socket.html>>

select - Waiting I/O completion

<<https://docs.python.org/3/library/select.html>>