CellHeap: Instâncias maiores podem não ser melhores

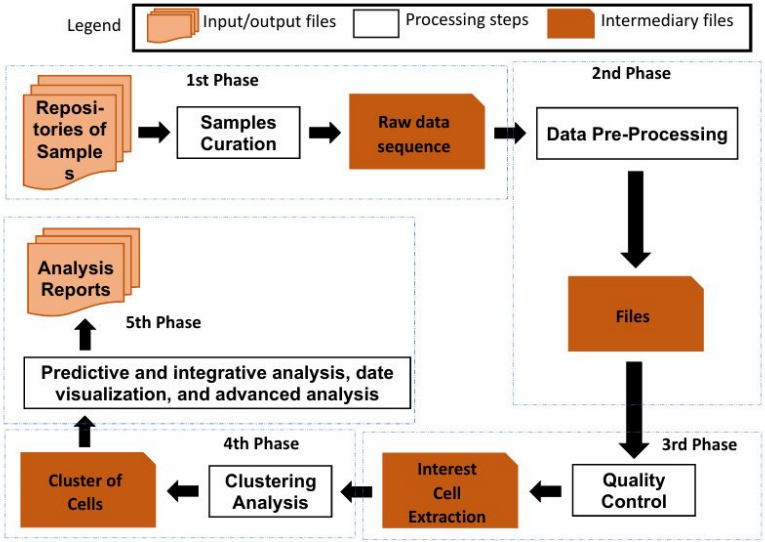
Alunos: Bernardo Mendes e Lucas Avelar
Orientador: Vinod Rebello

Curso de Ciência da Computação
Instituto de Computação, Universidade Federal Fluminense (UFF)

Introdução

Workflow CellHeap

- Proposto pela Fiocruz, CellHeap é um workflow flexível, portátil e robusto para analisar grandes conjuntos de dados scRNA-seq
    - Controle de qualidade em todas as fases de execução
    - Permite aproveitar plataformas que suportam dados em larga escala, como supercomputadores ou nuvens
- Composto por 5 fases



Maria Clicia S. Castro et al. CellHeap: A scRNA-seq workflow for large-scale bioinformatics data analysis. bioRxiv, p. 2023.04. 19.537508, 2023.

1ª fase CellHeap

Curadoria e framework

- Curadoria das amostras
  - Inúmeros repositórios públicos armazenam amostras
  - Uso de ferramentas, como *SRA toolkit* para baixar as amostras e validá-las
- Sobre implementação do framework
  - Realiza etapa de pré-processamento dos dados
  - Executa 3 comandos: *vdb-validate*, *fastq-dump*, *gzip*
  - É executada no ambiente de nuvem da AWS com o serviço **ParallelCluster**
  - A execução pode ser realizada com ou sem *work stealing*
    - Uma técnica adotada para balancear a carga de trabalho dos CPUs e dos nós do cluster

1ª fase CellHeap

Funcionamento do framework

- Recebe uma lista de amostras, geralmente obtidas do repositório de dados genéticos como o NCBI
- Aloca cada amostra a uma thread que executa uma sequência de comandos para validação, conversão e compressão dos arquivos *SRA*
- As execuções das threads ocorrem em múltiplos nós chamados *worker nodes*
- O trabalho (um conjunto de threads específicas) realizado por cada *worker node* é determinado inicialmente por um único nó.

Motivação

Análise e entendimento do framework (sem *work stealing*)

- Speedup abaixo do esperado: 2,8 com dois nós (4vCPUs)
- Observar a eficiência do framework
  - Quanto ao uso de diferentes instâncias do EC2
  - Quanto ao escalonamento
  - Quanto ao número de threads

Propostas e Objetivos

Propostas

- Verificar o tempo de execução das sequências com diferentes configurações
  - Diferentes números de threads
  - Instâncias menos custosas

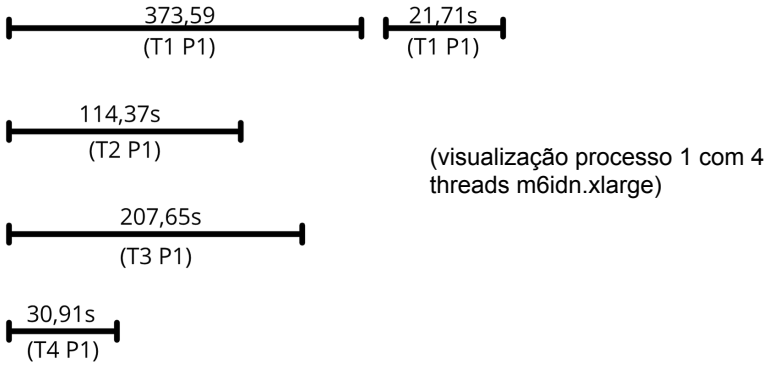
Objetivos

- Buscar uma configuração de threads e instância de forma a aproveitar o máximo do paralelismo
- Limitar o tempo de execução de uma thread com várias sequências ao tempo da sequência que é executada em maior tempo
- Analisar qual configuração poderia ter um custo mais baixo e que obtivesse um resultado próximo

Resultados Experimentais

Sequências				m6idn.xlarge		m6idn.large	
	SRR	Tamanho	Tempo individual	4 threads	2 threads	4 threads	2 threads
1	12570722	591,90 MiB	≈ 279,69s	373,59s (T1 P1)	312,64s (T1 P1)	382,66s (T1 P1)	312,21s (T1 P1)
2	12570159	98,40 MiB	≈ 48,08s	102,51s (T1 P2)	66,80s (T1 P2)	103,64s (T1 P2)	66,67s (T1 P2)
3	11783965	89,80 MiB	≈ 49,51s	114,37s (T2 P1)	70,45s (T2 P1)	110,07s (T2 P1)	70,30s (T2 P1)
4	11783964	131,40 MiB	≈ 74,19s	129,27s (T2 P2)	102,19s (T2 P2)	131,82s (T2 P2)	102,27s (T2 P2)
5	11783961	209,70 MiB	≈ 119,17s	207,65s (T3 P1)	118,49s (T1 P1)	207,20s (T3 P1)	118,52s (T1 P1)
6	1553580	114,00 MiB	≈ 37,05s	90,71s (T3 P2)	49,43s (T1 P2)	88,33s (T3 P2)	49,57s (T1 P2)
7	11409384	31,60 MiB	≈ 11,64s	30,91s (T4 P1)	15,53s (T2 P1)	31,93s (T4 P1)	15,45s (T2 P1)
8	19484251	43,10 MiB	≈ 16,17s	43,47s (T4 P2)	19,78s (T2 P2)	44,58s (T4 P2)	19,79s (T2 P2)
9	23922989	55,40 MiB	≈ 21,7s	21,71s (T1 P1)	21,81s (T1 P1)	21,92s (T1 P1)	21,84s (T1 P1)
		Tempo Total	**657,2s**	**398,27s**	**456,17s**	**405,67s**	**453,66s**

- Escalonamento foi feito de forma intercalada
- Leitura da tabela:



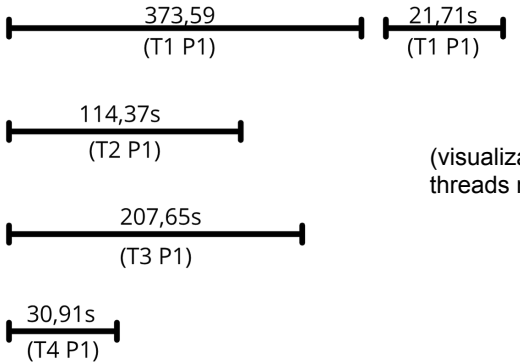
(visualização processo 1 com 4 threads m6idn.xlarge)

- Configurações das instâncias
  - m6idn.xlarge: 4 vCPUs / 16 GiB RAM / SSD 237 GB
  - m6idn.large: 2 vCPUs / 8 GiB RAM / SSD 118 GB

Resultados Experimentais

Sequências				m6idn.xlarge	
	SRR	Tamanho	Tempo individual	4 threads	
1	12570722	591,90 MiB	≈ 279,69s	373,59s (T1 P1)	
2	12570159	98,40 MiB	≈ 48,08s	102,51s (T1 P2)	
3	11783965	89,80 MiB	≈ 49,51s	114,37s (T2 P1)	
4	11783964	131,40 MiB	≈ 74,19s	129,27s (T2 P2)	
5	11783961	209,70 MiB	≈ 119,17s	207,65s (T3 P1)	
6	1553580	114,00 MiB	≈ 37,05s	90,71s (T3 P2)	
7	11409384	31,60 MiB	≈ 11,64s	30,91s (T4 P1)	
8	19484251	43,10 MiB	≈ 16,17s	43,47s (T4 P2)	
9	23922989	55,40 MiB	≈ 21,7s	21,71s (T1 P1)	
		Tempo Total	657,2s	398,27s	

- Escalonamento foi feito de forma intercalada
- Leitura da tabela:



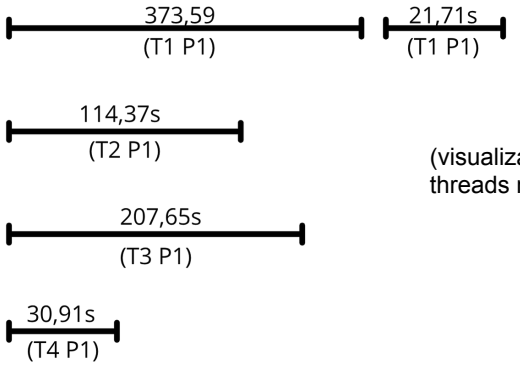
(visualização processo 1 com 4 threads m6idn.xlarge)

- Configurações das instâncias
  - m6idn.xlarge: 4 vCPUs / 16 GiB RAM / SSD 237 GB
  - m6idn.large: 2 vCPUs / 8 GiB RAM / SSD 118 GB

Resultados Experimentais

Sequências				m6idn.xlarge		m6idn.large	
	SRR	Tamanho	Tempo individual	4 threads	2 threads	4 threads	2 threads
1	12570722	591,90 MiB	≈ 279,69s	373,59s (T1 P1)	312,64s (T1 P1)	382,66s (T1 P1)	312,21s (T1 P1)
2	12570159	98,40 MiB	≈ 48,08s	102,51s (T1 P2)	66,80s (T1 P2)	103,64s (T1 P2)	66,67s (T1 P2)
3	11783965	89,80 MiB	≈ 49,51s	114,37s (T2 P1)	70,45s (T2 P1)	110,07s (T2 P1)	70,30s (T2 P1)
4	11783964	131,40 MiB	≈ 74,19s	129,27s (T2 P2)	102,19s (T2 P2)	131,82s (T2 P2)	102,27s (T2 P2)
5	11783961	209,70 MiB	≈ 119,17s	207,65s (T3 P1)	118,49s (T1 P1)	207,20s (T3 P1)	118,52s (T1 P1)
6	1553580	114,00 MiB	≈ 37,05s	90,71s (T3 P2)	49,43s (T1 P2)	88,33s (T3 P2)	49,57s (T1 P2)
7	11409384	31,60 MiB	≈ 11,64s	30,91s (T4 P1)	15,53s (T2 P1)	31,93s (T4 P1)	15,45s (T2 P1)
8	19484251	43,10 MiB	≈ 16,17s	43,47s (T4 P2)	19,78s (T2 P2)	44,58s (T4 P2)	19,79s (T2 P2)
9	23922989	55,40 MiB	≈ 21,7s	21,71s (T1 P1)	21,81s (T1 P1)	21,92s (T1 P1)	21,84s (T1 P1)
		Tempo Total	657,2s	398,27s	456,17s	405,67s	453,66s

- Escalonamento foi feito de forma intercalada
- Leitura da tabela:



(visualização processo 1 com 4 threads m6idn.xlarge)

- Configurações das instâncias
  - m6idn.xlarge: 4 vCPUs / 16 GiB RAM / SSD 237 GB
  - m6idn.large: 2 vCPUs / 8 GiB RAM / SSD 118 GB

Resultados Experimentais

Análise da tabela

- A instância **m6idn.large** com um CPU físico obteve tempos próximos aos da **m6idn.xlarge** com dois CPUs físicos.
- Atraso no término de cada sequência em nós que possuem mais threads
- Neste experimento, CPUs adicionais não oferecem o nível de benefício esperado
- O escalonamento utilizado prejudica o tempo total da execução sem *work stealing*

Conclusões

- A instância do tipo **m6idn.large** demonstra vantagens sobre a **m6idn.xlarge**: além de obter um resultado próximo, custa metade do preço
- O número de CPUs pode não ser o único fator que influencia o tempo
- Limitação do algoritmo de escalonamento (*round robin*) usado

m6idn.large	USD 0,15912	2	8 GiB	1 x 118 SSD NVMe	Até 25.000 megabits
m6idn.xlarge	USD 0,31824	4	16 GiB	1 x 237 SSD NVMe	Até 30.000 megabits

Amazon Web Services (2025)

Trabalhos Futuros

- Análise mais aprofundada do uso de CPU e as taxas de leitura e escrita obtidas tanto para os arquivos finais no armazenamento de longo prazo quanto no armazenamento local para arquivos temporários
- Realização de mais testes com um número maior de sequências de tamanhos maiores
  - Uso de métricas estatísticas e nova visualização dos dados
- Utilizar diferentes algoritmos de escalonamento para evitar o uso de *work stealing*
- Otimizar a execução dos comandos da Fase 1 com versões paralelas

`fastq-dump` **VS** `fasterq-dump` `gzip` **VS** `pigz`

Fim da Apresentação

Referências Bibliográficas

- Amazon Web Services (2025). AWS Parallel Cluster (https://docs.aws.amazon.com/parallelcluster)
- Maria Clicia S. Castro et al. CellHeap: A scRNA-seq workflow for large-scale bioinformatics data analysis. **bioRxiv**, p. 2023.04. 19.537508, 2023.
- National Library of Medicine. The Sequence Read Archive (SRA). (https://www.ncbi.nlm.nih.gov/sra/docs/)
- Erik Izidoro, Cristina Boeres, Vinod Rebello. Oferta ou Roubo: avaliando abordagens de escalonamento dinâmico para Workflow scRNA-seq na nuvem AWS, **ERAD-SE 2025.**
- Helena Silva, et al. A framework for automated parallel execution of scientific multi-workflow applications in the cloud with work stealing. In: **European Conference on Parallel Processing**, 2024. p. 298-311.