UL HPC School 2017

PS6: Bioinformatics Workflows and Applications

UL High Performance Computing (HPC) Team

V. Plugaru and S. Peter

University of Luxembourg (UL), Luxembourg

http://hpc.uni.lu

Latest versions available on Github:

UL HPC tutorials: https://github.com/ULHPC/tutorials

UL HPC School: http://hpc.uni.lu/hpc-school/

PS6 tutorial sources:

https://github.com/ULHPC/tutorials/tree/devel/advanced/Bioinformatics/

Summary

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain
- very many associated workflows, thus excellent examples

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain
- very many associated workflows, thus excellent examples

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain
- very many associated workflows, thus excellent examples

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain
- very many associated workflows, thus excellent examples

ABySS BOW TIE TopHat
A spliced read mapper for RNA-Seq

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain
- very many associated workflows, thus excellent examples

Objective of this Session

Better understand the usage of Bioinformatics packages on the UL HPC Platform.

Why Bioinformatics? 3Vs:

- very relevant in the context of the UL/LCSB
- very fast growing domain
- very many associated workflows, thus excellent examples

Summary

ABySS

ABySS: Assembly By Short Sequences
a de novo, parallel, paired-end sequence assembler designed for short reads

ABySS

ABySS: Assembly By Short Sequences
a de novo, parallel, paired-end sequence assembler designed for short reads

- several applications in the ABySS package
- only **ABYSS-P** is parallelized using MPI
  ↪ started with the **abyss-pe** launcher
- workflow (pipeline) of **abyss-pe** also includes:
  - ↪ OpenMP-parallel applications
  - ↪ serial applications
- Note: compared with other de novo assemblers, the per-node memory requirements are smaller due to ABySS' task distribution model

Gromacs

GROMACS: GROningen MAchine for Chemical Simulations
versatile package for molecular dynamics, primarily designed for biochemical molecules

Gromacs

GROMACS: GROningen MAchine for Chemical Simulations
versatile package for molecular dynamics, primarily designed for biochemical molecules

- very large codebase: 1.836.917 SLOC
- many applications in the package, several parallelization modes
- **mdrun**: computational chemistry engine, performing:
  ↪ molecular dynamics simulations
  ↪ Brownian Dynamics, Langevin Dynamics
  ↪ Conjugate Gradient
  ↪ L-BFGS
  ↪ Steepest Descents energy minimization
  ↪ Normal Mode Analysis
- **mdrun** - parallelized using MPI, OpenMP, pthreads and with support for GPU acceleration

Bowtie2/TopHat

Bowtie2: Fast and sensitive read alignment

ultrafast & memory-efficient alignment of sequencing reads to long ref. sequences

TopHat: A fast spliced read mapper for RNA-Seq

alignment of RNA-Seq reads to a genome, to identify exon-exon splice junctions

Bowtie2/TopHat

Bowtie2: Fast and sensitive read alignment

ultrafast & memory-efficient alignment of sequencing reads to long ref. sequences

TopHat: A fast spliced read mapper for RNA-Seq

alignment of RNA-Seq reads to a genome, to identify exon-exon splice junctions

- TopHat aligns reads to mammalian-sized genomes using Bowtie
- then analyzes the mapping results to identify splice junctions between exons
- **bowtie2** is OpenMP-parallel
- rest of workflow is sequential

mpiBLAST

mpiBLAST: Open-Source Parallel BLAST

parallel implementation of NCBI BLAST, scaling to hundreds of processors

mpiBLAST

mpiBLAST: Open-Source Parallel BLAST

parallel implementation of NCBI BLAST, scaling to hundreds of processors

- two main applications: **mpiblast mpiformatdb**
- requires (NCBI) substitution matrices and formatted BLAST databases
- the databases can be segmented
  - ↪ into as many segments as the # of cores that will be used when performing searches
  - ↪ or a multiple, in order to avoid load imbalance
- **mpiblast** requires >= 3 processes, 2 used for internal tasks
  ↪ mpirun -np 3 mpiblast [...] only gives you one searcher process!

Summary

Notes..

.. on real world applications (bioinfo or others):

- make sure you *understand the parallel capabilities* of your software
  ↪ pthreads/OpenMP vs MPI vs hybrid
  ↪ use of GPU acceleration

Notes..

.. on real world applications (bioinfo or others):

- make sure you *understand the parallel capabilities* of your software
  ↪ pthreads/OpenMP vs MPI vs hybrid
  ↪ use of GPU acceleration
- make sure you *request the appropriate resources* for the processing needs of your workflow
  - ↪ Does the software always take advantage of more than 1 core or node?
  - ↪ How does it scale? Many obstacles to perfect scalability!

Notes..

.. on real world applications (bioinfo or others):

- make sure you *understand the parallel capabilities* of your software
  ↪ pthreads/OpenMP vs MPI vs hybrid
  ↪ use of GPU acceleration
- make sure you *request the appropriate resources* for the processing needs of your workflow
  - ↪ Does the software always take advantage of more than 1 core or node?
  - ↪ How does it scale? Many obstacles to perfect scalability!

.. on data management:

- make sure you use *the appropriate storage place*
  ↪ $HOME vs $WORK vs $SCRATCH
- stage data in/out, archive your (many & unused) 'small' files

Summary

Exercises

- Read and understand the Bioinformatics tutorial

  https://github.com/ULHPC/tutorials/tree/devel/advanced/Bioinformatics/

- Run the examples

  ↪ all calculations should be fast

  ↪ you should attempt the exercises proposed in each section

- Try even more tests, e.g.:
  - ↪ on different node classes
  - ↪ with one core per node on $>=$ 2 nodes
  - ↪ vs $>=$ 2 cores on single node

Summary

Conclusion

- Bioinformatics applications execution on the UL HPC Platform
- Outlined:
  - ↪ different workflows
  - ↪ some of the concepts you should care about when running complex software

Perspectives

- Personalize the UL HPC launchers with the specific commands for ABySS, Gromacs, TopHat, Bowtie, mpiBLAST..

Thank you for your attention...

Questions?

http://hpc.uni.lu

The UL High Performance Computing (HPC) Team
University of Luxembourg, Belval Campus:
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu