Stay Tuned

How NeSI Optimises the Usage of Shared HPC Resources
Computational Science Team @ NeSI

Jordi Blasco
(jordi.blasco@nesi.org.nz)

NeSI
New Zealand eScience Infrastructure

Outline

1. About NeSI CS Team
   - Who we are?
   - Who we are?
2. Identify the Bottlenecks
   - Identify the Most Popular Apps
3. Tuning
   - Profile and Debug
   - Increase the performance
   - Discover the scalability limits
   - Increase the efficiency
4. Back to the future
   - Moore's Law
   - Sequencing cost

About NeSI CS Team

Computational Science Team

Basically, it means that we are the Researchers best friends :-)

About NeSI CS Team

Which is the main goal of a shared HPC facility

- Run few jobs really fast? - no matter what their efficiency are.
- Run the maximum number of jobs? - in favour of short and small jobs.
- NeSI provides a wise combination by optimising the usage of computational resources.

Identify the Most Popular Apps

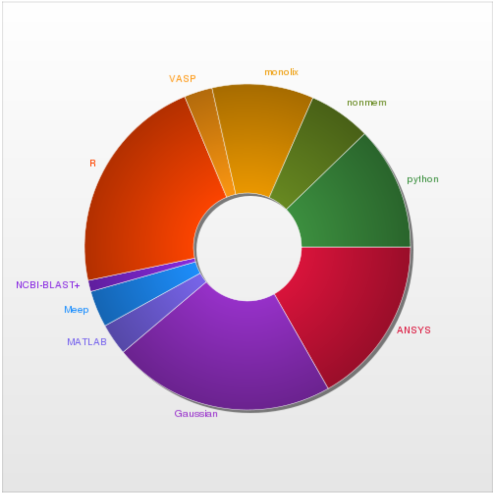
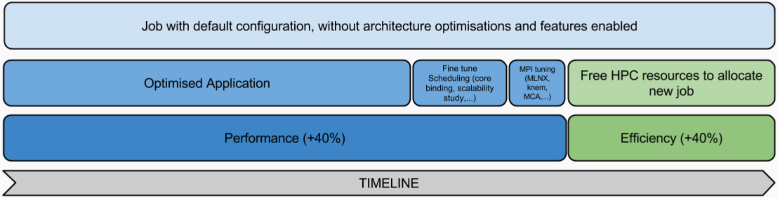


Figure : Most popular applications in Pan cluster. We are using snoopy to

Tuning

Tuning the most popular applications provides a major impact in

- the walltime of the users of this code.
- the global cluster efficiency.
- the availability of the computational resources.
- the waiting time.



Job with default configuration, without architecture optimisations and features enabled

Optimised Application

Fine tune Scheduling (core binding, scalability study,...)

MPI tuning (MLNX, knem, MCA,...)

Free HPC resources to allocate new job

Performance (+40%)

Efficiency (+40%)

TIMELINE

Tuning

There are several ways to tune an HPC Application

- Most obvious : tune the algorithm.
- Choose the right Libraries + Compilers + MPI "Flavour".
- Choose the right Options and Environment.
- Work in the work-flow.
- Explore the scalability (how well it scales).
- Check if benchmark results are good enough.

Profile and Debug

Available software

- Intel Vtune Amplifier & Intel Trace Analyzer
- DDT
- Score-P
- HPC Toolkit
- Scalasca
- Cube
- PAPI
- TAU
- Parallel Profile Visualization (ParaProf)
- Native Slurm profiling tools

Increase the performance

PhyML Case Study - Dr. Stéphane Guindon (UoA)

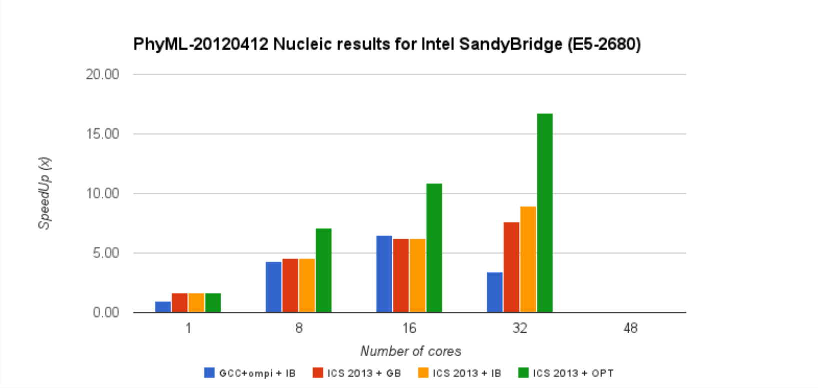
PhyML is a software that estimates maximum likelihood phylogenies from alignments of nucleotide or amino acid sequences. The main strength of PhyML lies in the large number of substitution models coupled to various options to search the space of phylogenetic tree topologies, going from very fast and efficient methods to slower but generally more accurate approaches.

Using the right tools

In this case, the right compilers and optimization options for an specific architecture increased the performance up to x6.

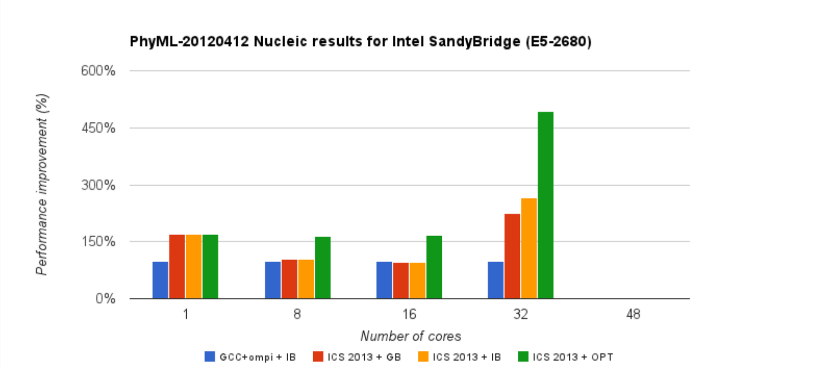
Increase the performance

PhyML Case Study : Speed Up



Increase the performance

PhyML Case Study : Performance



Discover the scalability limits

How well the code scales?

- There are theoretical limits but the reality can be really surprising!
- The scalability/quality of a code is measured in terms of efficiency (USL).
- The benchmarks can help to discover the real scalability limits.
- With this information you can get even faster results and save computational resources for other jobs.

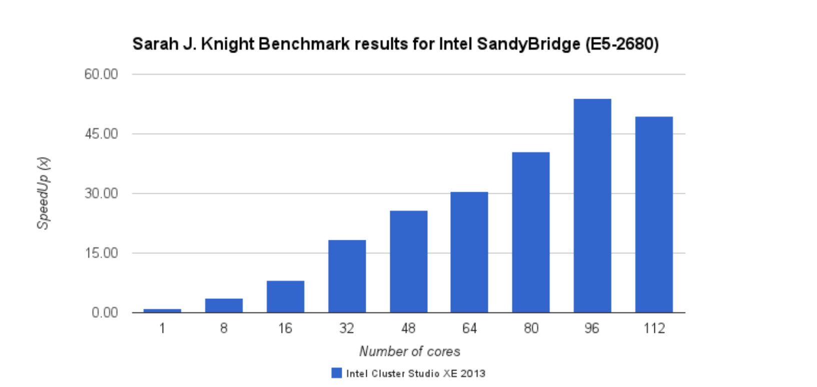
Discover the scalability limits

Migrate-n Case Study - Dr. Sarah Knight (UoA)

Migrate estimates effective population sizes and past migration rates between n population assuming a migration matrix model with asymmetric migration rates and different subpopulation sizes.

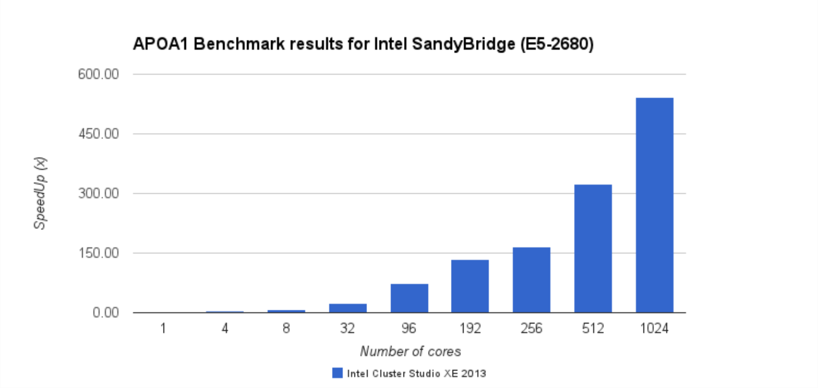
Discover the scalability limits

Migrate-n Case Study



Discover the scalability limits

Is the scalability as good as you expect?



Improve the efficiency

The workflow can save several HPC resources

- Thanks to the nature of some codes the problem can be transformed into a embarrassing parallel problem.
- In some cases it's possible to split large input file into several small files, allowing to run several copies of the same code using completely independent input files.
- In scenarios like these you can achieve a linear scalability.
- Several problems related with genomics have this approach.

Improve the efficiency

NCBI-BLAST+ Case Study

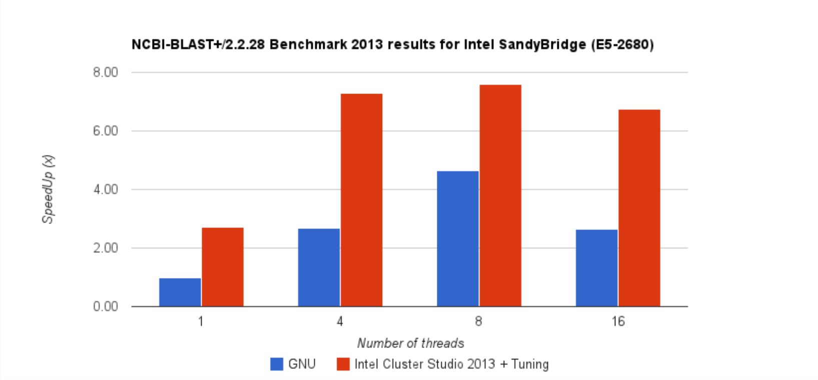
BLAST (Basic Local Alignment Search Tool) command line applications developed at the National Center for Biotechnology Information (NCBI).

Benchmark provided by Dr. Daniel White (Landcare Research)

It describes the dataset as a pathogen discovery dataset using de novo metagenomics.

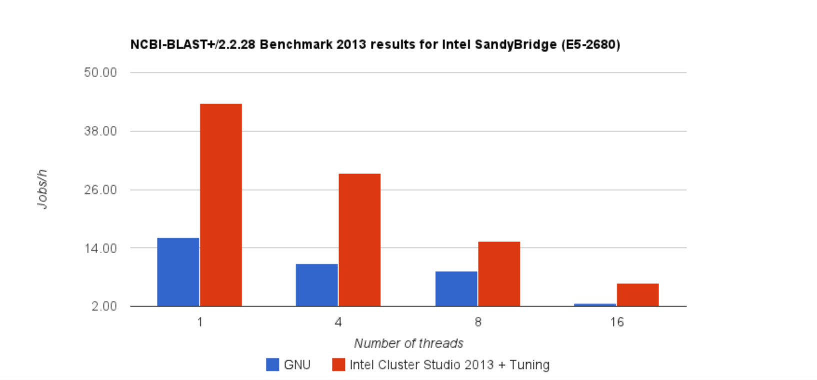
Improve the efficiency

NCBI-BLAST+ Case Study : Speed Up



Improve the efficiency

NCBI-BLAST+ Case Study : Efficiency



Improve the efficiency

NCBI-BLAST+ Case Study : Speed Up compared with original code

threads	split	tuned binaries	SHM	Speed Up	Efficiency
1	1000	2.71	●	●	●
4	1000	7.27	●	●	●
8	1000	7.61	●	●	●
16	1000	6.74	●	●	●

Back to the future

Moore's Law

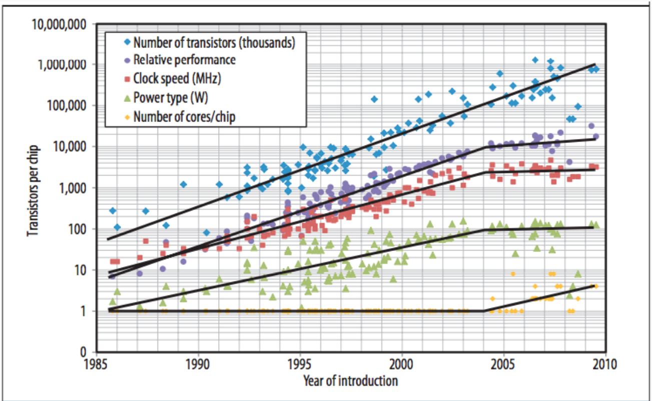
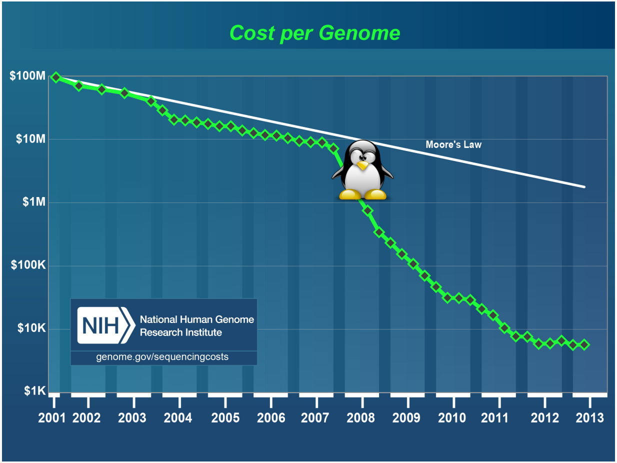


Figure 1. Transistors, frequency, power, performance, and processor cores over time. The original Moore's law projection of increasing transistors per chip remains unabated even as performance has stalled.

Back to the future

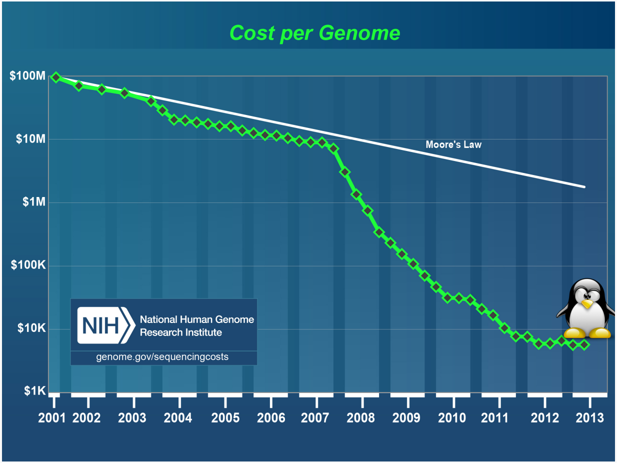
Sequencing Cost per Genome



Back to the future

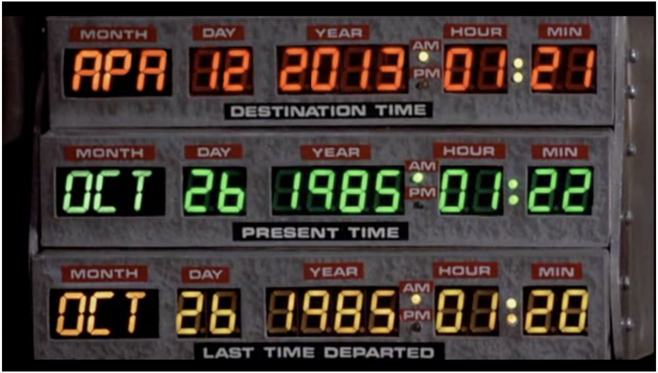
Back to the future

Sequencing Cost per Genome



Back to the future

Are you still living in the 80's?



Stay Tuned
FOR something
Awesome

Questions & Answers

NeSI

New Zealand eScience
Infrastructure