Numbench

A benchmarking tool for Gentoo

Google Summer of Code 2011 project

Andrea “spiros” Arteaga

Numbench

- Tool targeted for Gentoo
- Modular structure, written in Python
- Can test:
  - Implementations of standard library interfaces
  - Libraries
  - Programs
- Generates HTML reports with plots

Outline

1. Motivation
2. Implementation
3. Results
4. Outlook

Motivation: BLAS

- Standard fortran API for basic linear algebra subroutines

Example:

void dgemv_(char*, int*, int*, double*, double*, int*, double*, int*, double*, double*, int*)

$$y \leftarrow \alpha A x + \beta y$$

- Widely used in scientific applications (and not only)
- Many implementations exist. Usually they are targeted for specific systems, CPUs, OSs,...
- Gentoo offers many choices: reference, atlas, openblas, eigen, mkl, acml
- Every package installs one or more implementations

Motivation: BLAS

Find the best solution for the local machine by:

- comparing different implementations
- comparing different versions of the implementation
- compiling the packages with specific flags (CFLAGS)
- using different compilers of compiler versions
- using different run-time environments (e.g. OMP_NUM_THREADS)

Motivation: FFTW

“Just” a library: only one implementation. However, we can:

- compare different versions (does it improve?)
- compile it with different CFLAGS (how relevant are they)
- compile it with different compilers (is icc better than gcc?)

Motivation: Metis

It is an application for graph and partitioning

- Same questions as above

In general:

- **Does the performance depend on the compilation target?**

Motivation

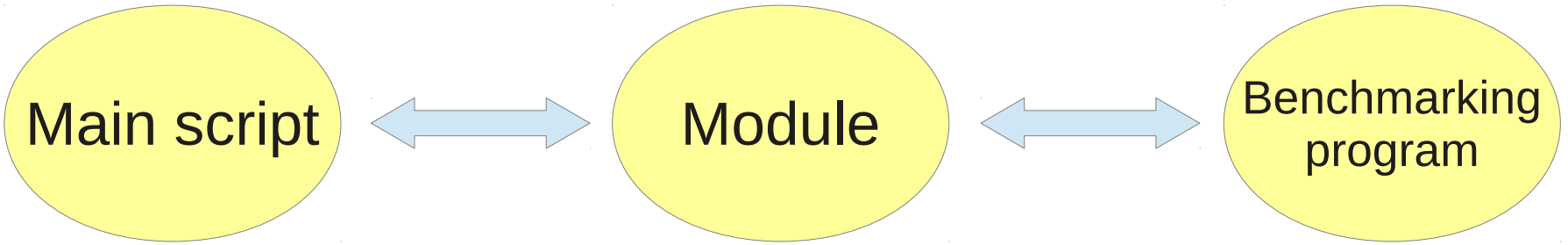
Let's answers these questions by

BENCHMARKING

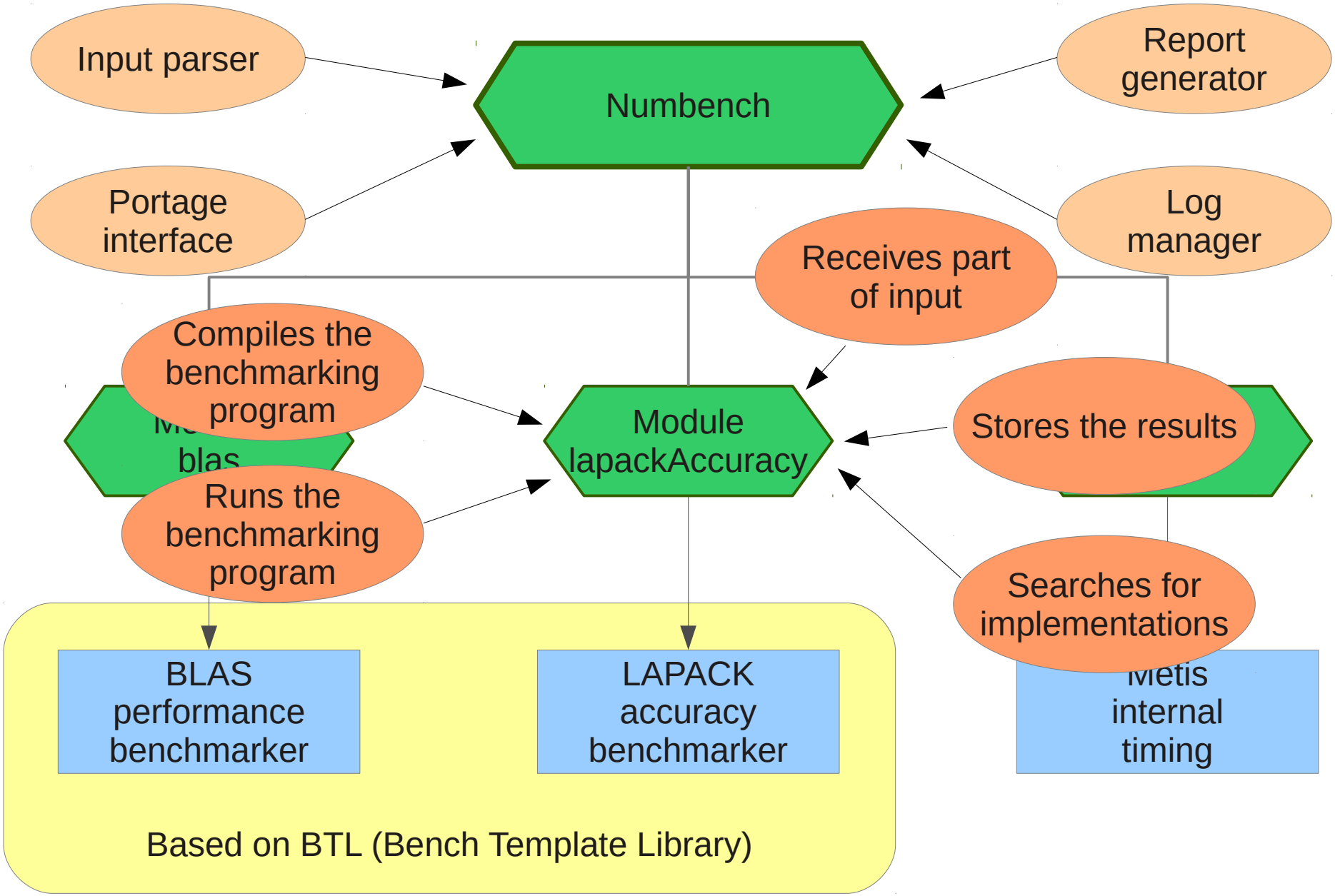
Implementation

Three components:

- Main script
- Modules
- Benchmarking programs



Implementation: scheme



Implementation: input

Input is given through an XML document. It specifies:

- Module that has to be run (with arguments)
- Test cases

A test case is basically an atom with some information:

- Emerge-time environment
- Run-time environment
- Implementations that should be skipped

Implementation: results

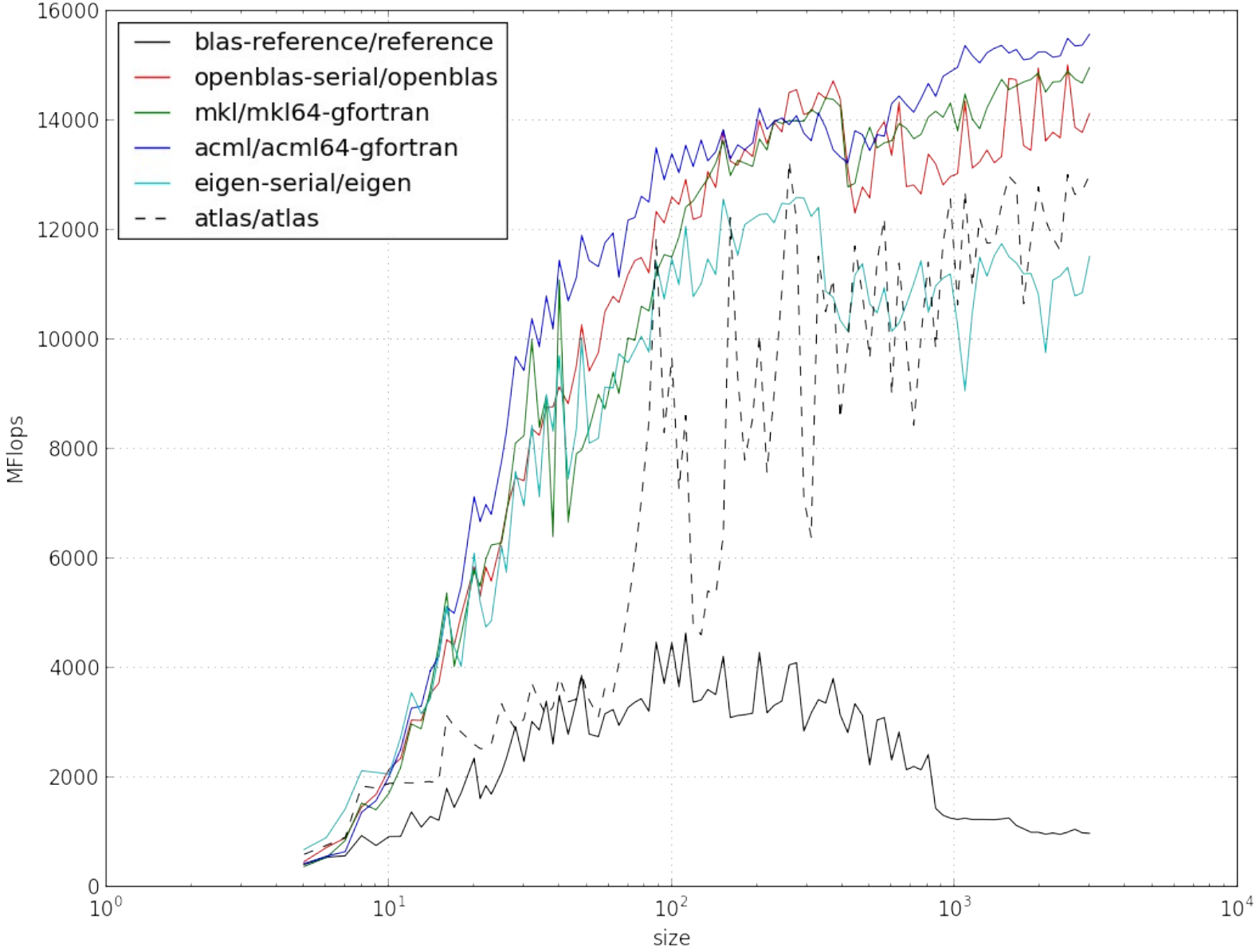
- Numbench expects benchmarking results as DAT files
- It generates plots
- And an HTML document, ready for publishing
- Report gives access to logs

Implementation: features

- Compeletely automatized benchmarking tool
- Benchmarking programs for numerics based on the high-quality Bench Template Library
- Sandboxed execution: it does not require superuser privileges
- In case of error, the intermediate results are re-usable
- Everything is logged, even the correct linkage of the benchmarking program

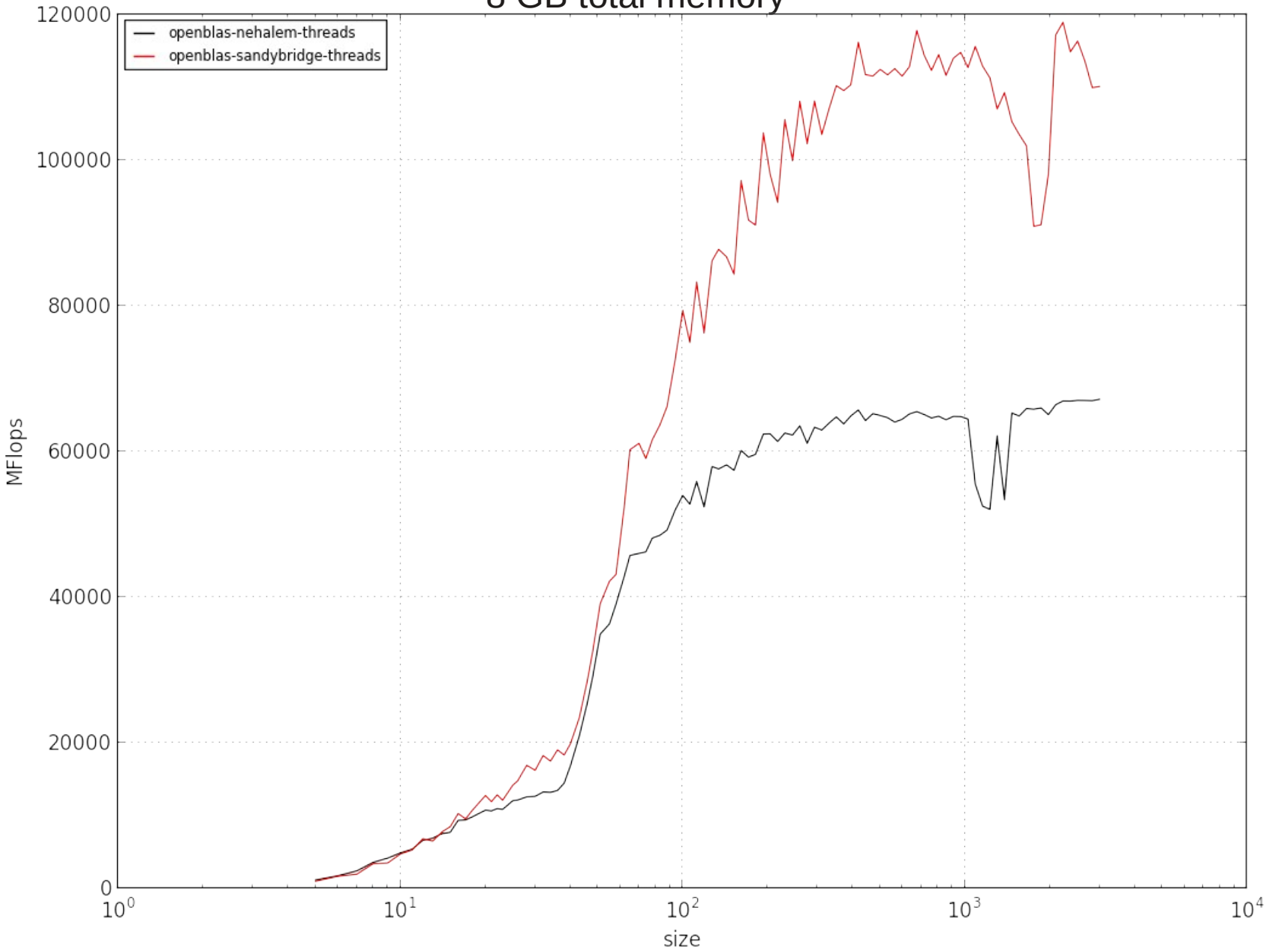
Results

BLAS: matrix-matrix multiply serial implementations on
Quad-Core AMD Opteron(tm) Processor 8354
132363512 kB total memory



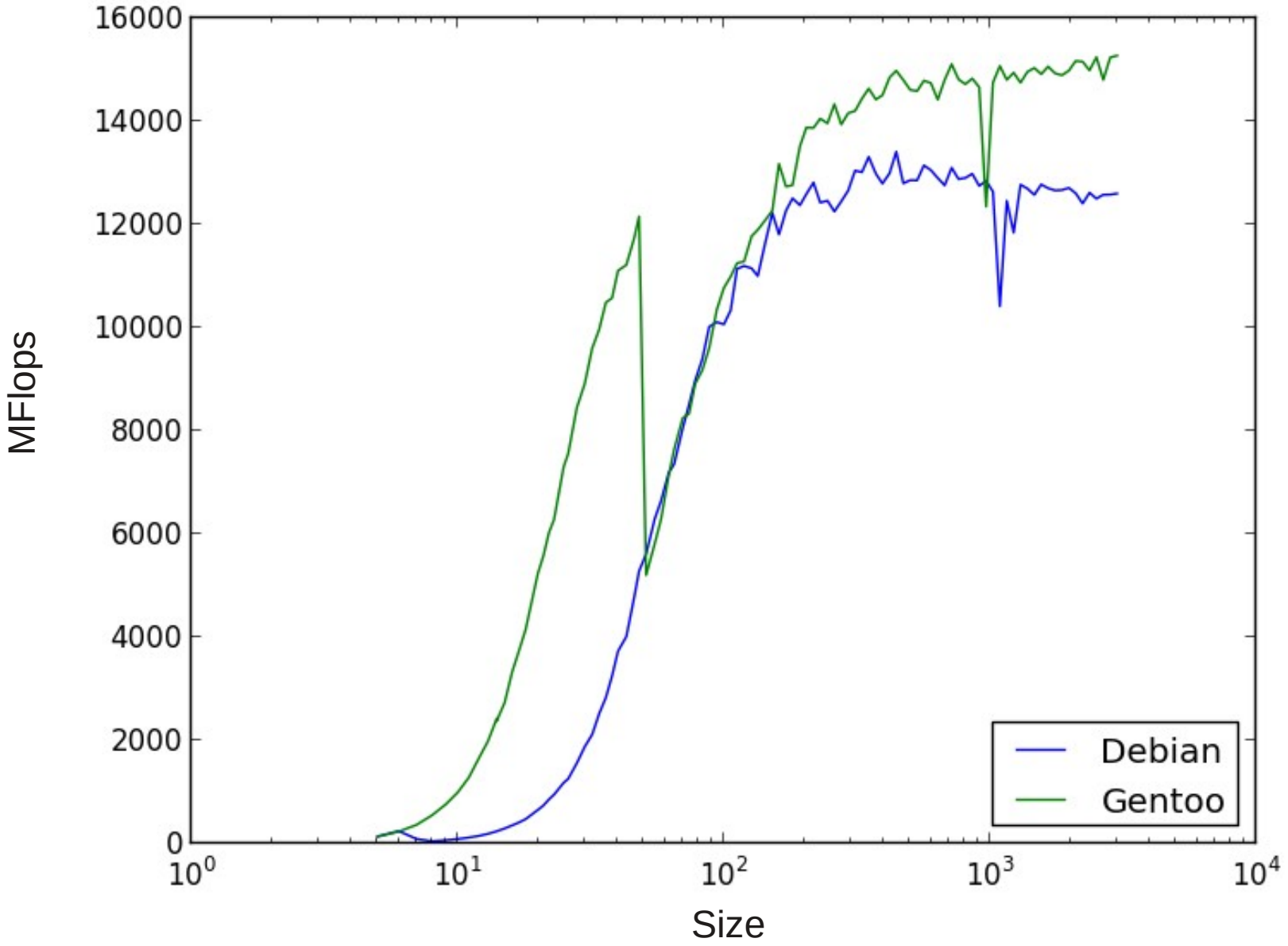
Results

BLAS: matrix-matrix multiply, openblas threaded implementation
Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
8 GB total memory

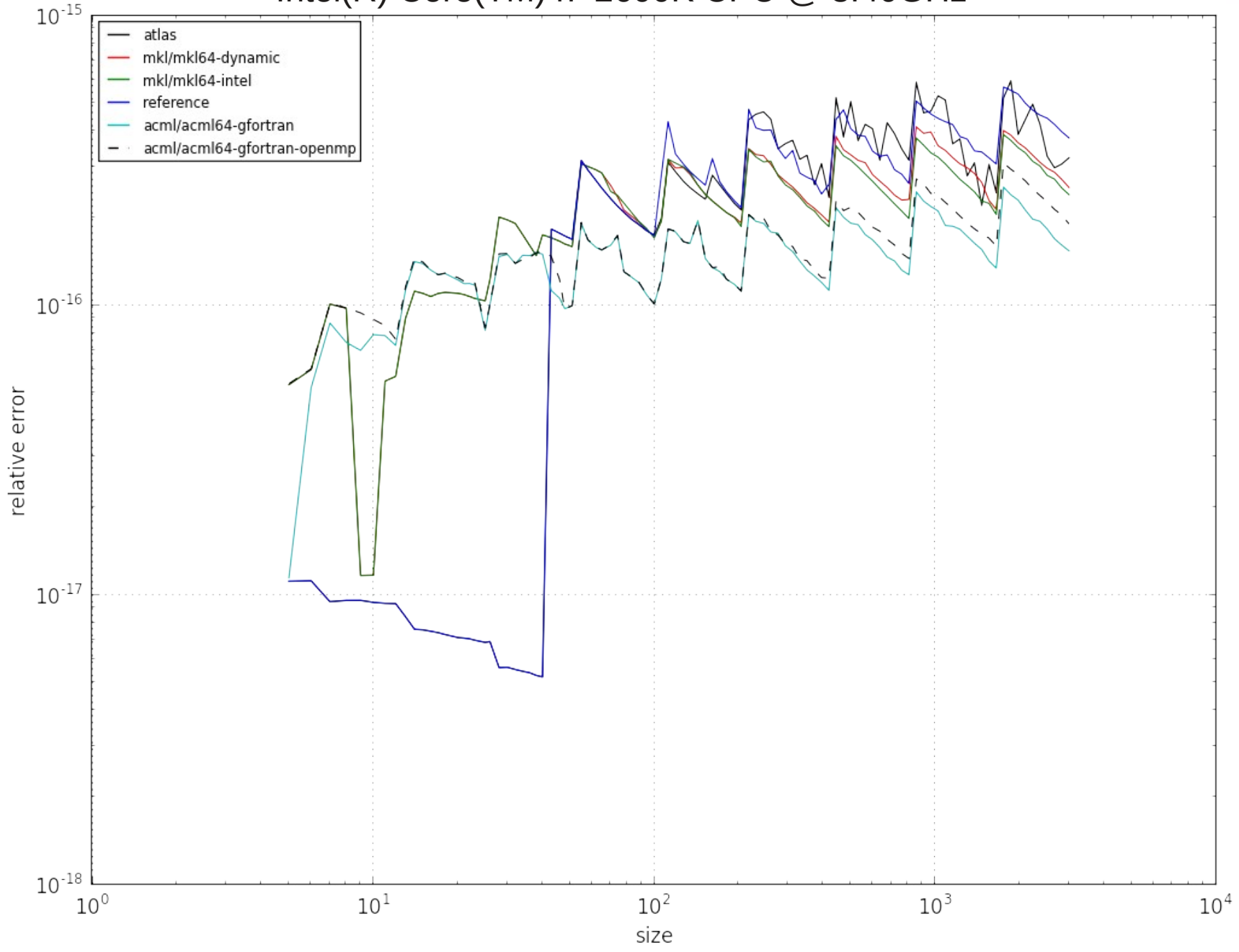


Results

BLAS: matrix-matrix multiply
Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
VirtualBox machine with 4 cores, 4 GB RAM



Results


LAPACK: LU decomposition, relative error
Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
atlas
mkl/mkl64-dynamic
mkl/mkl64-intel
reference
acml/acml64-gfortran
acml/acml64-gfortran-openmp
relative error
size
10^{-15}
10^{-16}
10^{-17}
10^{-18}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}

Outlook

Goals for the future:

- Complete the accuracy benchmark modules
- Add some more numerical modules
- Complete the web interface
- Enlarge the user base
- Enlarge the developers team

Outlook: other possible uses

- What numbench basically does:
  - Automatically install packages with specific environments
  - Perform some (highly customizable) tests with those packages
  - Displays results in form of graphs
- It can be used for many other tasks than benchmarking
- It is more a tool related to Gentoo than a tool devoted to numerics

References

- The project is hosted on Github: https://github.com/andyspiros/numbench
- The ebuild is available in the science overlay: app-benchmark/numbench
- Feel free to contact me directly: spiros@gentoo.org

Thanks for your attention