SC18 BoF 107: Pros and Cons of HPCx benchmarks

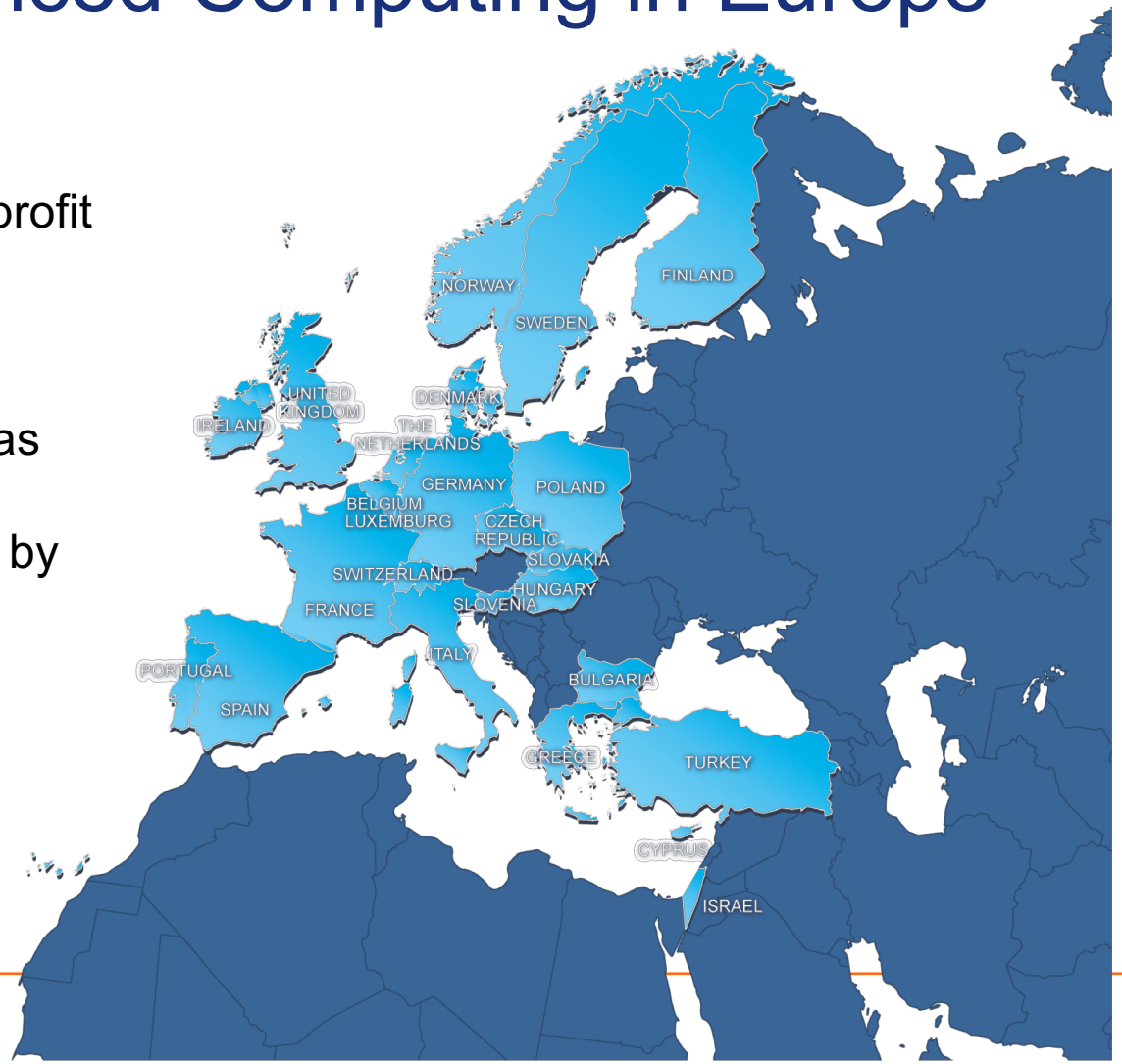
Walter Lioen <walter.lioen@surfsara.nl>

Unit Manager Compute Services @ SURFsara, NL
Benchmark Activity Leader @ PRACE, EU

SURF SARA

Partnership for Advanced Computing in Europe

- **PRACE** is an international not-for-profit association
- **PRACE** has 25 members and 2 observers
- **PRACE** is funded by its members as well as through a series of implementation projects supported by the European Commission
- Computing resources are made available by a group of members (Hosting Members)



PRACE Unified European Application Benchmark Suite | UEABS

The UEABS is a set of currently 14 application codes taken from the pre-existing PRACE and DEISA application benchmark suites, and extended with the PRACE Accelerator Benchmark Suite.

The UEABS has been and will be actively updated and maintained by the subsequent PRACE-IP projects.

The current list of UEABS codes is: ALYA, Code_Saturne, CP2K, GADGET, GENE, GPAW, GROMACS, NAMD, NEMO, PFARM, QCD, Quantum Espresso, SHOC, SPECFEM3D

Release 2.0 (May 31, 2018) is available via PRACE GitLab:
https://repository.prace-ri.eu/git/UEABS/ueabs/tree/master

Purpose of the unified benchmark suite

- The objective is providing a single benchmark suite of scalable, currently relevant and publicly available application codes and datasets, of a size which can realistically be run on large systems, and maintained into the future.
  - Note: there is no obligation on PRACE partners to use these codes in any procurement
- To provide performance data on existing PRACE systems to assist users when choosing which system to apply for time on.
- To provide data for “currency conversion” of CPU hours between PRACE systems.

Datasets

- Originally, each code comes with one or two test cases
- Where the codes have two test cases:
  - Test case A is suitable for Tier-1 systems (scales to ~1000 x86 cores)
  - Test case B is suitable for Tier-0 systems (scales to ~10,000 x86 cores)
- Where there is one dataset, it is suitable for both Tier-1 and Tier-0

- For “accelerated” benchmark subset extended with smaller test cases
- For some benchmarks extended with test cases that scale to ~100,000 x86 cores

Dutch National Supercomputer: Cartesius

- Used by academic users
  - Dutch Universities
  - Dutch Academic Research Institutes
- Also available for industry and SMEs
- General purpose, many science fields, many applications
- Funded by:
  - Ministry of Education, Culture and Science
  - Ministry of Economic Affairs
  - Netherlands Organization for Scientific Research (NWO)
  - SURF / SURFsara
- On demand growth:
  - June 2013 (Phase 1): 271 Tflop/s
  - November 2016 – end of 2019 (expected): 1.8 Pflop/s
- Built by
  - Atos/Bull

Procurement Process

- System requirements based on detailed usage analysis
  - which user applications
  - actual memory usage
  - I/O profiles
- Optimized price/performance
  - TCO: total budget = investment + energy + cooling + housing + ups (storage only)
  - performance: application throughput using the 7 most relevant applications (# jobs / lifetime)
  - maximization of application throughput / TCO
    (optimization of power related costs vs. investment costs)
    left as an “exercise” for the vendor during the procurement
  - result: using “slower” processors (lower clock frequency)

SURFsara Application Benchmark Suite (SABS 2012)

Benchmark Code	Scientific area	Scaling (MPI tasks)	Weight
ADF	Quantum chemistry	384	10%
GROMACS	MD	2048, 1024, 4096	20%
POP	Ocean circulation	1280, 640, 2560	15%
SPARKLE	CFD	1024	15%
SPO-DVR	Molecular QD	512, 256, 1024	10%
SUSHI	Cosmology	2048	15%
VASP	ab-initio QM-MD	128	15%

- Application benchmark codes selected based on
  - use
  - spread across scientific areas
  - scaling (potential)
- These 7 codes represent 50% of the work load on Huygens (2008 – 2012)
- Final application benchmark set selected in consultation with NWO-WGS

Throughput of System

- System:
  - total number of cores: size
  - expected lifetime: period
- Benchmark result per application:
  - wall clock time
  - number of cores
- Throughput of an application on a system:
  - number of jobs = (size × period) / (cores × time)
- Normalization per application:
  - over all systems: best = max(jobs)
- Overall result for a system:
  - over all applications: result = Σ weight × jobs / best
- System with best overall throughput:
  - over all systems: winner = max(result)

Discussion Topics (after the presentations)

- TOP500 relevance
- Overall application throughput vs. peak speed
- Can we do with synthetic benchmarking only using application modeling?
- Actual throughput benchmarking

THANK YOU FOR YOUR ATTENTION

Walter Lioen

walter.lioen@surfsara.nl

www.surfsara.nl

http://nl.linkedin.com/in/walterlioen

SURF SARA