MARCIN COPIK <MARCIN.COPIK@INF.ETHZ.CH>

DPHPC

Recitation session, 5.12.2019

Few comments…

1. Make sure to describe statistical correctness of your experiments!

Few comments…

1. Make sure to describe statistical

Scientific Benchmarking of Parallel Computing Systems

Twelve ways to tell the masses when reporting performance results

Torsten Hoefler
Dept. of Computer Science
ETH Zurich
Zurich, Switzerland
htor@inf.ethz.ch

Roberto Belli
Dept. of Computer Science
ETH Zurich
Zurich, Switzerland
bellir@inf.ethz.ch

ABSTRACT

Measuring and reporting performance of parallel computers constitutes the basis for scientific advancement of high-performance computing (HPC). Most scientific reports show performance improvements of new techniques and are thus obliged to ensure reproducibility or at least interpretability. Our investigation of a stratified sample of 120 papers across three top conferences in the field shows that the state of the practice is lacking. For example, it is often unclear if reported improvements are deterministic or observed by chance. In addition to distilling best practices from existing work, we propose statistically sound analysis and reporting techniques and simple guidelines for experimental design in parallel computing and codify them in a portable benchmarking library. We aim to improve the standards of reporting research results and initiate a discussion in the HPC field. A wide adoption of our minimal set of rules will lead to better interpretability of performance results and improve the scientific culture in HPC.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Reproducing experiments is one of the main principles of the scientific method. It is well known that the performance of a computer program depends on the application, the input, the compiler, the runtime environment, the machine, and the measurement methodology [20, 43]. If a single one of these aspects of *experimental design* is not appropriately motivated and described, presented results can hardly be reproduced and may even be misleading or incorrect.

The complexity and uniqueness of many supercomputers makes reproducibility a hard task. For example, it is practically impossible to recreate most hero-runs that utilize the world's largest machines because these machines are often unique and their software configurations changes regularly. We introduce the notion of *interpretability*, which is weaker than reproducibility. *We call an experiment interpretable if it provides enough information to allow scientists to understand the experiment, draw own conclusions, assess their certainty, and possibly generalize results*. In other words, interpretable experiments support sound conclusions and convey precise information among scientists. Obviously, every scientific paper should be interpretable; unfortunately, many are not.

For example, reporting that an High-Performance Linpack (HPL) run on 64 nodes (N=314k) of the Piz Daint system during normal operation (cf. Section 4.1.2) achieved 77.38 Tflop/s is hard to interpret. If we add that the theoretical peak is 94.5 Tflop/s, it

Few comments…

1. Make sure to describe statistical correctness of your experiments!

2. Tell us how you get input data.

Few comments…

1. Make sure to describe statistical correctness of your experiments!

2. Tell us how you get input data.

3. Plots: titles, units, speedup.

Few comments…

1. Make sure to describe statistical correctness of your experiments!

2. Tell us how you get input data.

3. Plots: titles, units, speedup.

4. How many dimensions in your experiments? # of cores, problem size (multiple params?), algorithms.

Few comments…

1. Make sure to describe statistical correctness of your experiments!

2. Tell us how you get input data.

3. Plots: titles, units, speedup.

4. How many dimensions in your experiments? # of cores, problem size (multiple params?), algorithms.

5. Tell us where you run experiments.

Few comments…

1. Make sure to describe statistical correctness of your experiments!

2. Tell us how you get input data.

3. Plots: titles, units, speedup.

4. How many dimensions in your experiments? # of cores, problem size (multiple params?), algorithms.

5. Tell us where you run experiments.

Look at your report and ask yourself: do I have enough information to repeat the experiments?
Many details are crucial yet require adding only a single sentence!

Quiz!

kahoot.it

Volunteers wanted!

Serverless benchmarking

Serverless supercomputing

…and others!

Work on paper submission to upcoming, top-class conferences.

marcin.copik@inf.ethz.ch