Performance, Portability and Productivity: Definitions & Metrics

Lead:
John Pennycook (Intel Corporation)

Co-Leads:
David Beckingsale (Lawrence Livermore National Laboratory)
Bob Bird (Los Alamos National Laboratory)
Doug Jacobsen (Intel Corporation)

DOE COE Performance Portability Meeting 2017, Denver, CO

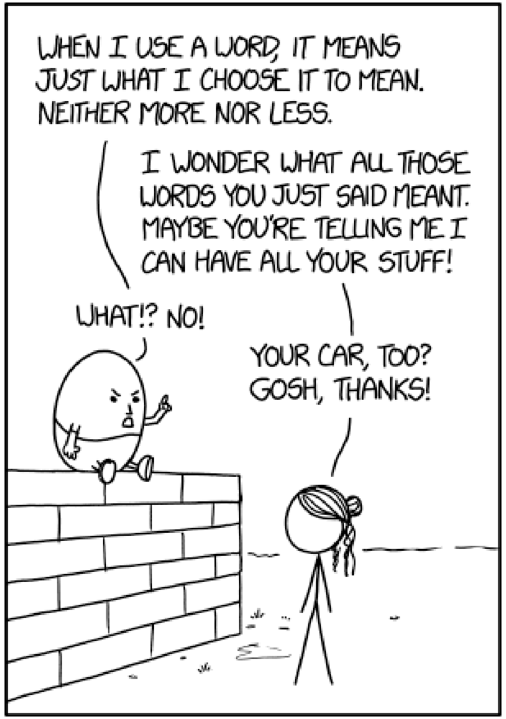
Motivation

"Let us not get tied up in definitions. ***Performance portability means different things to different people and we need to accept that****. Both performance and portability are poorly defined and depend on the applications. Every app has different constraints and there is no way to get around it."*

- Unnamed Participant

DOE COE Meeting 2016

(Emphasis mine)


WHEN I USE A WORD, IT MEANS JUST WHAT I CHOOSE IT TO MEAN. NEITHER MORE NOR LESS.
I WONDER WHAT ALL THOSE WORDS YOU JUST SAID MEANT. MAYBE YOU'RE TELLING ME I CAN HAVE ALL YOUR STUFF!
WHAT!? NO!
YOUR CAR, TOO? GOSH, THANKS!

Agenda

- 5 questions to tackle, with a maximum of 20 minutes each:
  1. Are shared definitions and metrics useful to the community? Why (or why not)?
  2. What do performance, portability and/or productivity mean to you?
     What about combined terms (e.g. performance portability)?
  3. Are you/your site currently tracking performance, portability and productivity of codes? How?
  4. How do metrics and definitions inform solutions to these problems in a software sense?
  5. How do metrics and definitions inform solutions to these problems in a hardware sense?

Question 1

Are shared definitions and metrics useful to the community?

Why (or why not)?

Generally – Yes. They help folks compare. Frameworks, hardware, etc…

It'd be helpful to push some of the definitions back onto the vendors, via standardization. A benchmark suite to check for performance portability would be useful during procurements.

Question 2

What do performance, portability and/or productivity mean to you?

What about combined terms (e.g. performance portability)?

Most of the discussion was focused on the definitions of portability and productivity.

Generally, people consider productivity to be part of portability (i.e. how much effort does a code base require to port to a new architecture, **can** a code be ported to a new architecture). (**Code Portability**)

Alternate definitions: At an instant in time, can you build / run on all of your architectures of interest. (**Application Portability**).
NOTE: This is also used by code developers, via regression tests.

Some definitions can hurt developers, for example if you are deemed not-PP, but are portable as far as the machines you care about.

Question 3

Are you/your site currently tracking performance, portability and productivity of codes?

How?

Performance is tracked via regressions against a “known” benchmark.

Portability is tracked via the previous “Application Portability” definition, or by users telling developers that the code no longer runs when given a new architecture.

Some people said yes, via issue trackers (on github for example) or scrum trackers (like JIRA) that track Story points for tasks.

Question 4

How do metrics and definitions inform solutions to these problems in a **software** sense?

Ran out of time before this question, but some of the discussions implied the following:

Sometimes, they tell people they need to refactor their code to be more agile when new architectures come out.

The software seems to be the **hard** part, while the hardware seems to be the **soft** part. As in, most people are stuck with the hardware they are given, and need to adapt their software to work with it.

Code specialization seems necessary to help with performance portability.

How do third party libraries (i.e. PETsc, Trilinos) and programming models (Kokkos, Raja) factor into these definitions of Performance Portability? If they do not support a new architecture, can you consider the model that uses it portable?

Question 5

How do metrics and definitions inform solutions to these problems in a **hardware** sense?

See previous slide…

Next Steps

Now what should we do??

- Work with developers to test out definitions on real codes
- Play with alternate definitions
- Explore different methods of measuring productivity within codes

Legal Notices and Disclaimers

Thanks to Tina Macaluso for taking notes

intel®