Shades of Grey

William Gropp

www.cs.illinois.edu/~wgropp
What is Productivity?

- Time and cost to a science result
- It is not just the time to write a program
- Many components
  - Time to write code
  - Time to rewrite the code to make it fast enough
  - Time to run code
  - Time to fix bugs
  - Time to run code
  - Time to analyze the results
Are Heterogeneous Systems Hurting Productivity?

- Yes and no
- Yes (and obvious): More to learn, code specialization, risk of focusing on the details
- No: Time to science may decrease
  - Yes it might be more work
  - Yes it will be a problem if we don’t support those writing these codes
  - But if the time to science decreases, it is worth it.
Is Nested Programming Bad?

- If blindly implemented, Yes
- E.g., if no attention is paid to “+” in MPI+X, then nested programming will be bad
- Otherwise,
  - Nested programming is essential
  - Realistic execution models won’t be simple; different elements of the machine will be easier to program with programming systems that reflect that part of the execution model
  - This is the way we engineer everything else!
Will overheads in emerging runtimes be worth their benefit?

• Yes

• Systems will have less predictable performance properties (they already do!)

• See “Time to make code fast”
  ♦ Emerging runtimes will shorten this time and help with correctness (runtime absorbs some of code complexity)

• However, must focus on latency, not just bandwidth
  ♦ E.g., need a spectrum of methods to handle very fine to very coarse grain work
Will Exascale Hardware and Software be too hard to use?

- No.
- Yes – if you insist on a non-nested, single general purpose language solution.
- Plausible approaches:
For the Developers

• MPI-4 between nodes
  • Includes rich RMA, nonblocking operations
  • Fully adaptive dynamic model (even MPI-1 not just BSP)

• OpenMP+{C11/C++11/Fortran}

• Annotation/EDSL/etc. for performance transportability
  • This is where the real productivity tools are needed
For the Users

• None of that stuff. They want to solve their problems.

• No single solution, but
  ♦ Frameworks, community codes, ...
  ♦ “PETSc saves my students over a year”

• Interoperability with other systems
  ♦ Key is to support the data structures that match the algorithms