Integrating Performance Modeling (PM) concepts into performance tools

• Automatic construction of performance models based on direct measurements (without code instrumentation) seems feasible

• A “performance model” in this context must be something simple (resource/bottleneck-based).
  • restrictions compared to fully analytic model
  • Roofline (from measured flop rate & memory traffic) or ECM (ditto)
  • IB/network traffic may be incorporated into the model

• HPCToolkit already solves most of the problems?
  • phase detection ← counter-based call stack sampling
  • steady state problem ← associating samples to code regions/loop bodies)
Integrating Performance Modeling (PM) concepts into performance tools

- Phase detection may be implemented by / based on object detection algorithms (e.g., Llort et al. SC13)
  - Possible metrics: IPC, flop/s, arith. intensity, or a combination
  - IPC often misleading

- “Low-hanging fruits” are “easy” if metrics are available (scalar loads? Flops?)
  - load imbalance
  - non-SIMD
  - NUMA traffic

- So, “data acquisition” is not the problem, but how about automatic interpretation of the data?
Integrating Performance Modeling (PM) concepts into performance tools

• **PerfExpert** provides “simple” resource-based advice
  • “What, based on our in-situ measurements, seems to be the bottleneck in your code?”
  • this is the **essence of Roofline**

• Could be augmented by “beyond Roofline” thinking providing, e.g., ranges (best vs. worst case)
  • ECM model, $T_{\text{comm}}/T_{\text{comp}}$ overlap/non-overlap etc.

• Phases must be exposed to user
Integrating Performance Modeling (PM) concepts into performance tools

• “Performance patterns” may be helpful in categorizing “issues”
  • max resource utilization / hazards / work-related
  • Some patterns require more than in-situ measurements → not directly accessible by automatic tools

• Patterns provide another refinement (besides ECM, etc.) of the bottleneck thinking

• In summary, the measurement infrastructure and simple resource-based modeling based on the data seems to be there; accurate phase detection seems possible but not really ready for prime time (one bottleneck record per phase).
• Link to “Patterns” slides: 
  https://www.dropbox.com/s/la0b91s2tj6cdje/Patterns.pdf?dl=0

• “Patterns” paper: http://dx.doi.org/10.1007/978-3-642-36949-0_50
Thank you!

georg.hager@fau.de