



Challenge and Solutions for { Peta | Exa }-scale Programming

WPSE09 panel discussion

Raymond Namyst

“Runtime” group
INRIA Bordeaux Research Center
University of Bordeaux 1
France



Legal Disclaimer

- This presentation represents **EXCLUSIVELY** personal views of the author and nobody else. Any relationship between the opinions expressed here and any official position of INRIA is purely coincidental.
- Software developed by the Runtime group are not intended for use in medical, life saving or nuclear simulation applications.



Overview of my answers

(I will develop them out-of-order!)

- **When will Exa-scale systems come?**
 - November 2018
- **What kind of programming style remains?**
 - All of them!
(I'm afraid we have to solve the hybrid programming problem...)
- **The most difficult challenge for {Peta|Exa}-scale software?**
 - Composability
- **What programming issues are solved/will remain at the end?**
 - Runtime issues will be solved ☺
- **Your free opinion on {Peta|Exa}-scale application/system software?**
 - I'm scared!



What kind of programming style remains?

- **I wish it would be OpenMP 5.0 or XcalableMP 3.0 or UPC 4.0!**
 - We need fine-grain, structured parallelism!
- **However:**
 - We are totally surrounded by (almost) natural born MPI programmers
 - MPI has proved to be very efficient on clusters
- **But I don't believe in the "pure, flat MPI" model**
 - How to develop portable topology-aware applications?
 - How to balance load when using 100 cores and 4 GPGPU with MPI?
- **So I think the number of hybrid applications will increase in the future**
 - Note that hybridization may often be indirect
 - [Parallel Kernels](#)



The most difficult challenges for Peta|Exa-scale software?

- Fighting against Amdahl...
 - Exascale computers = embarrassingly parallel machines?
- Solving the “hybrid programming model” problem
 - Compatibility
 - Ever tried to mix MPI + OpenMP with more than one MPI process per node?
 - Technical issues (can be solved by the runtime)
 - Composability
 - Ever tried to mix OpenMP and MKL? OpenMP and Intel TBB?
 - Semantics issues (cannot be solved by runtime only)
- Providing performance feedback to the programmer
 - Can we still understand performance?
- Allowing the user to give scheduling hints
 - Composability of hints? ☺



What programming issues are solved/will remain at the end?

- Technical problems can be solved rather easily
 - However, they may require more inter-domain coordination than today
- Some problems (composability, feedback) can only be solved by a common agreement on
 - New parallel language extensions
 - New APIs
 - And new ideas!



Your free opinion on Peta- and Exa-scale application/system software?

- **Software problems need to be solved at multiple stages**
 - We need to strengthen cooperation between
 - Numerical kernel libraries
 - Parallel languages compilers
 - Runtime systems
 - Usually we try to put it all together *_after_* the APIs are (independently) fixed
 - ➔ **More collaborations between research groups**
- **Complex parallel architectures will require powerful runtime systems**
 - Communication, I/O and thread scheduling are highly linked
 - Don't focus on a single aspect
 - This is something we have to anticipate!
 - Remember how long it took before getting a thread-aware MPI implementation?
 - ➔ **Again: more collaborations between research groups!**



When will Exa-scale systems come?

- **Very difficult question!**
- **An easiest question would be to guess the vendor that will release the first one...**
 - The machine may have deep blue stickers on it!
 - More seriously, it will be a vendor "MPP" machine
- **But what's more important is to think about the landscape of exa-scale systems**
 - Probably a large part of commodity clusters (as today)
 - With all associated software development intricacies
- **The date?**
 - First Exascale computer will be unveiled at SC 2018 ☺