CS Discussions

ExMatEx All-Hands

LA-UR-14-27909

Get to Prototype

- What's needed to get us to petascale prototype?
- Keep in mind the "blue sky" requirements
- Once we have the sample problem
  - Model and simulate it.
  - Program it.
  - Run it on some system.

Modeling & Simulation

- Given:
  - Problem (CoEVP, LULESH, fine-scale, queries, etc.)
  - Target platform (Sequoia? Trinity?)
  - Desired execution time (?)
- Answer…
  - Very basic questions (LULESH/fine-scale split)
  - Database requirements (drive testing)
  - Target problem size (LULESH)? Parallel? Elastic?

“Tools”

- Aspen
  - Models of LULESH, fine-scale (VPFFT, etc.)
  - Time-dependent random variables in development
- SST (O.B.H.F)
  - Parallel, full-system sim
    - calling to ASPEN for single-scale
    - Time-dependent random variables working (?)
    - First cut available by report due date (?)
- Optimization (McKerns ideas)
- Gremlins (with parts of prototype apps MPI/not)

Programming

- In addition to DSLs and MPI+X explore task-based programming models
  - Makes sense for dynamic apps in dynamic environment.
  - Make sense for "static" scales in "static" environment? Why? Over decomposition, etc?
  - Have explored as runtime system work and in summer schools—found lacking in some areas (?)
  - Common semantics possible? Common model?
  - Where do we go from here?

XStack, FF2, DF

- Opportunities to collaborate—leverage expertise
  - Refactor proxy into offeror-specific model
  - Tight collaboration and commitment (3/30?)
    - How to prioritize and choose?
  - ExMatEx provides physics/algorithms/methods
  - ExMatEx learns new model from expert (and the refactoring process)
  - Offeror gains proxy-driven development reality
  - Other than this what do we learn? Can we measure it?
    - Performance? Productivity? Portability?
  - XStack/DF/FF2 issues (multi-scale for XStack?)
  - CnC, OCR, DEGAS/UPC++, Chapel,… (reach out to others)

Running the Prototype

- No time to discuss this, but…
- Traditional target system (Sequoia? Trinity?)
  - MPI+X (and some things we can install as users)
    - Dynamics, etc. handled by application developer (lib?)
  - Database implementations
    - Local vs. distributed?
    - Feasibility on target systems?
    - "C0 problem", et. al.
- Additional work on non-traditional system to feed exascale specification
  - Databases, languages, runtime systems, services, etc.
  - Explore where we could go vs. where we had to go