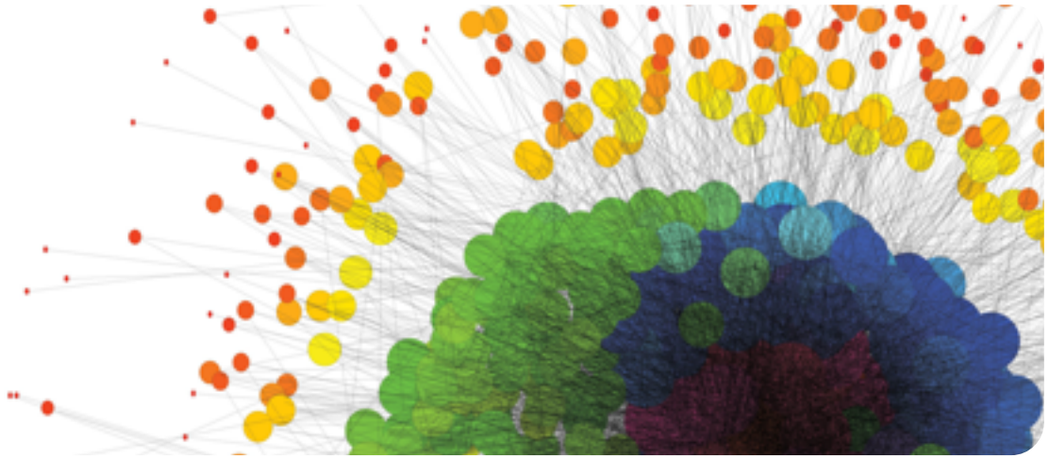
SAND2012-8649P

Exceptional service in the national interest

Sandia National Laboratories



Toward Translucent Heterogeneity with Qthreads

Kyle Wheeler

PGAS-X Workshop 2012

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

SAND: 2012-7850P

The Exascale Challenge

Sandia National Laboratories

- **What does an exascale machine look like?**

- What does system software for an excascale machine look like?
  - Collaborate with the level above and the level below
  - Leverage technology trends
  - Rethink application space (what will be important in a decade?)
  - *Be metric-focused!*
    - Picojoules, picojoules, picojoules ... and time too!

There's no THERE there!

Sandia National Laboratories

- **What does the future look like?**
  - 125 MW+ exascale platform
  - Cannot cool aggressive clockrates as feature size scales
  - All future performance gains through parallelism (3–5 orders of magnitude!)
    - DATA MOVEMENT and not FLOPS dominates energy performance
- **Apps increasingly unstructured**
  - DS&A
  - Physics
- **Given a limited investment portfolio, where do we invest?**
  - Data movement!
  - Parallelism!
    - Gains needed to match today's performance power efficiently
    - Plus 3 orders of magnitude to achieve exascale
- **The current commodity trajectory is insufficient to hurdle the energy wall**
  - Must influence more than the interconnect
  - The programming model is insufficient to meet the challenges

From where we sit...

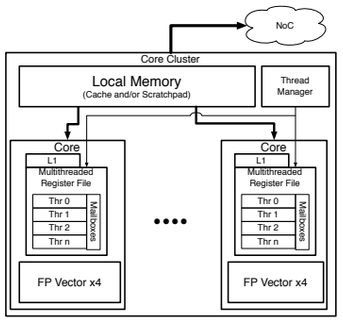
Sandia National Laboratories

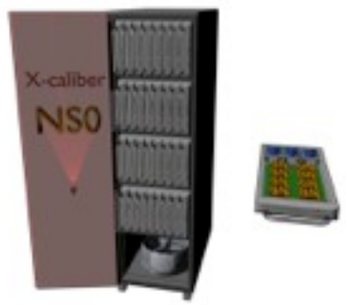
Safety and Security | **Reentry** | **Circuitry** | **Graph** | **Stream**

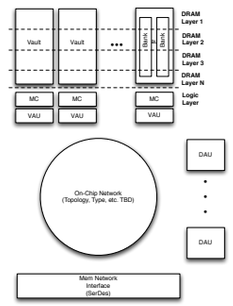
Application Drivers

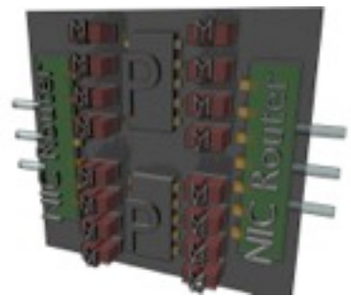
System Software - enabling a new model of computation

Architecture - Coping with Concurrency and Data Movement

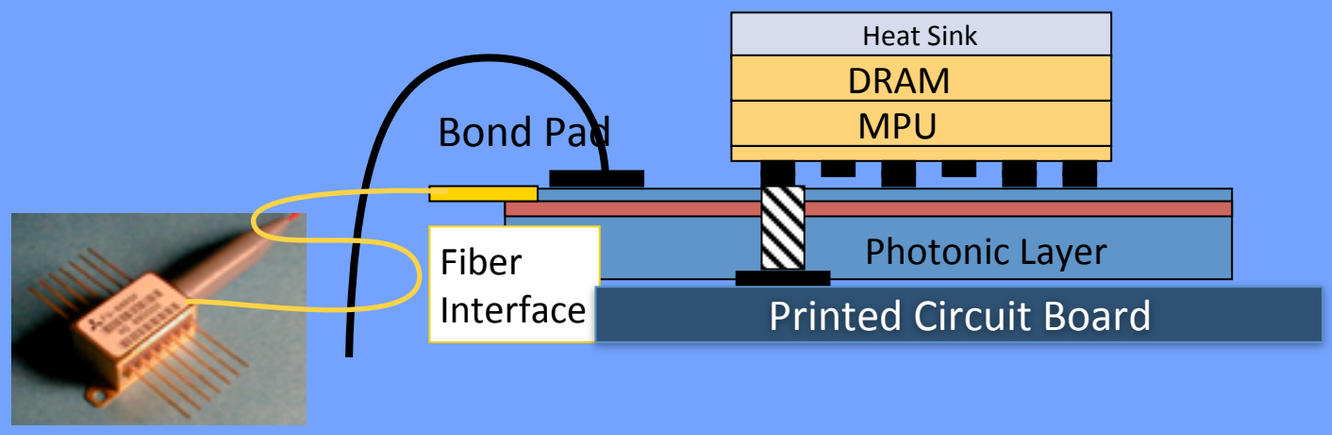


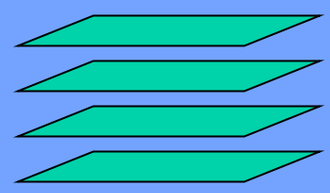






Microsystems - Key Data Movement Enabling Technologies

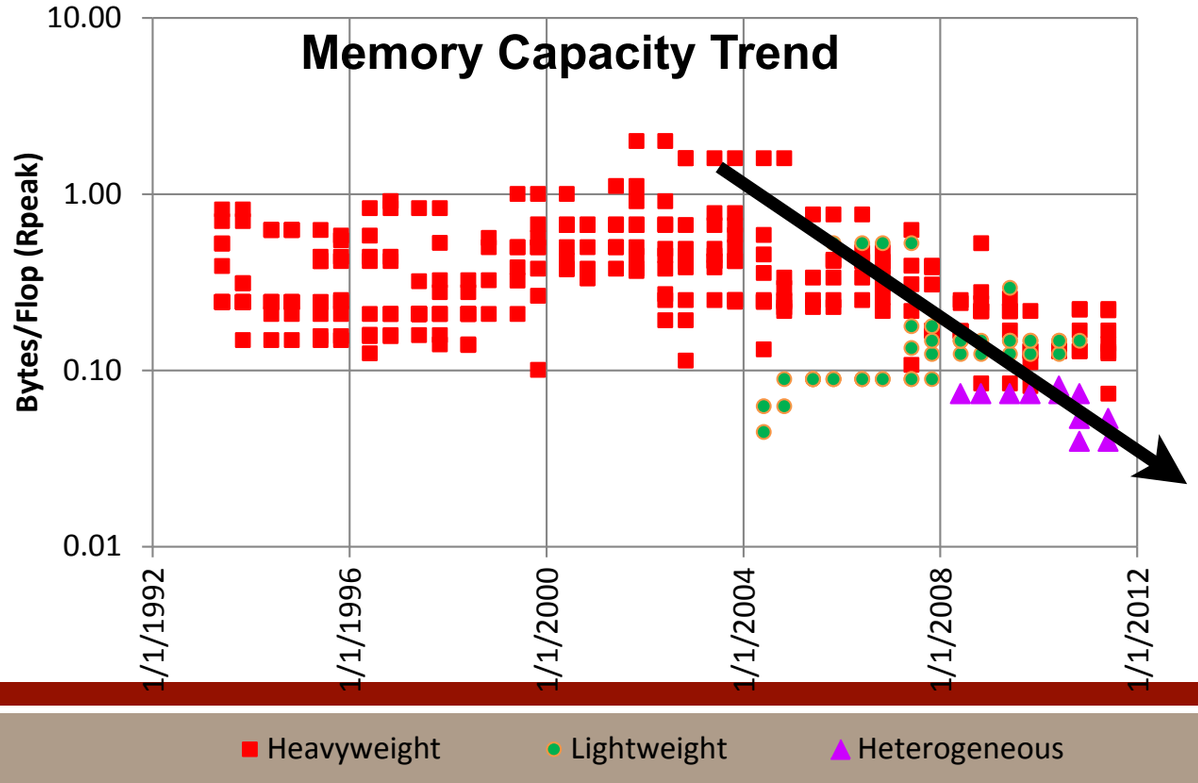


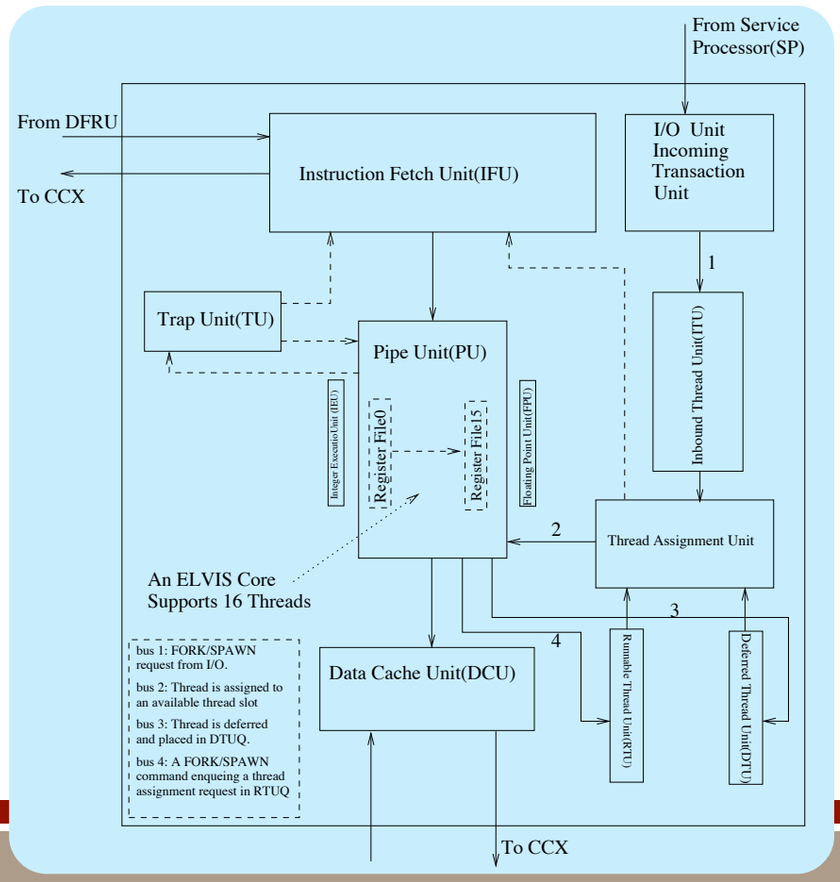


Trouble Down Below: Hardware

Sandia National Laboratories

- Exponential increase in node-level parallelism
- Lower memory capacity per core
  - Weak scaling will be insufficient!
- Heterogeneity at node-level
  - Not necessarily ISA-level
- Need for system software to have finer-grained control of hardware resources

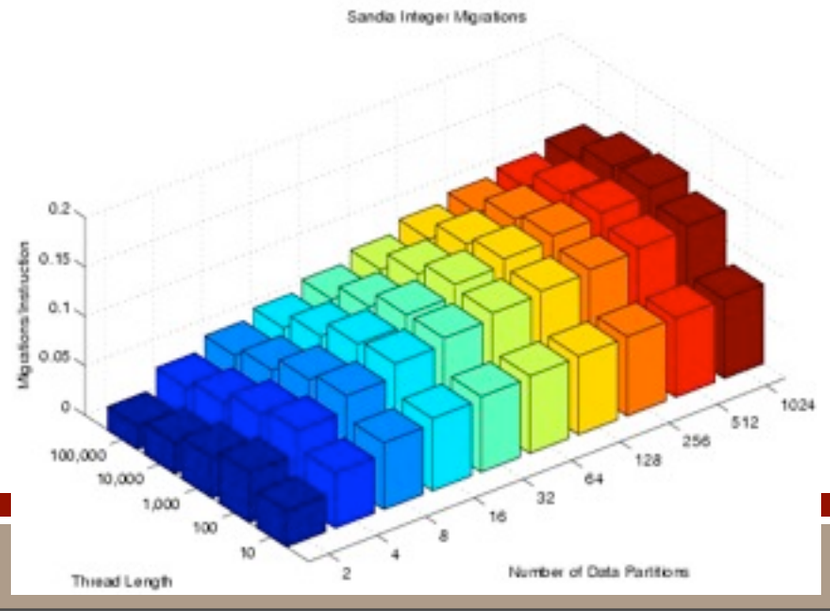


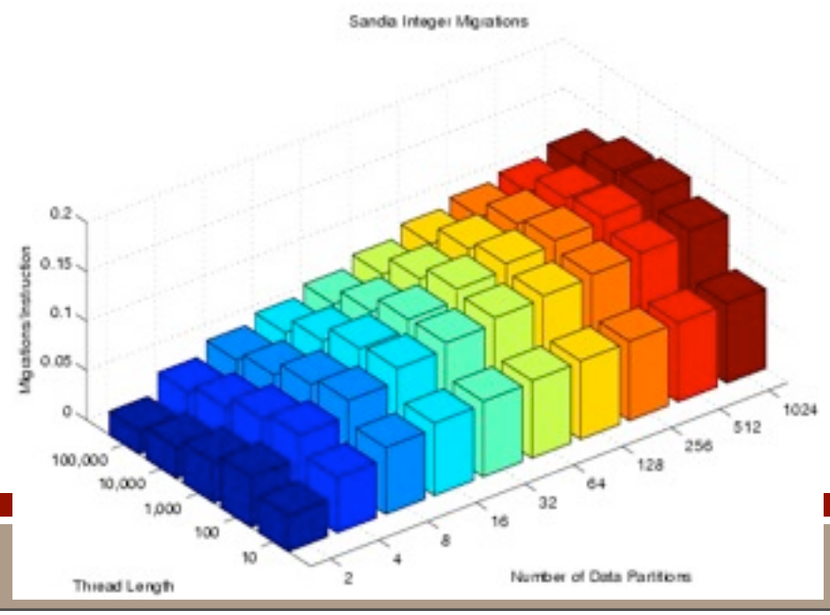


Programming Model Falls Down

Sandia National Laboratories

- **Every core in the system has to know about every other core in the system**
  - We can't afford the energy for today's loose coupling
  - We can't manage the concurrency (even locally) with today's model and today's coupling — the synchronization problem is too hard
- **New models are required**
  - 5–400x improvements in data movement over the application suite
    - **THIS IS WHERE YOUR ENERGY GOES!**
  - Reduced thread state size (15% of a modern register file)

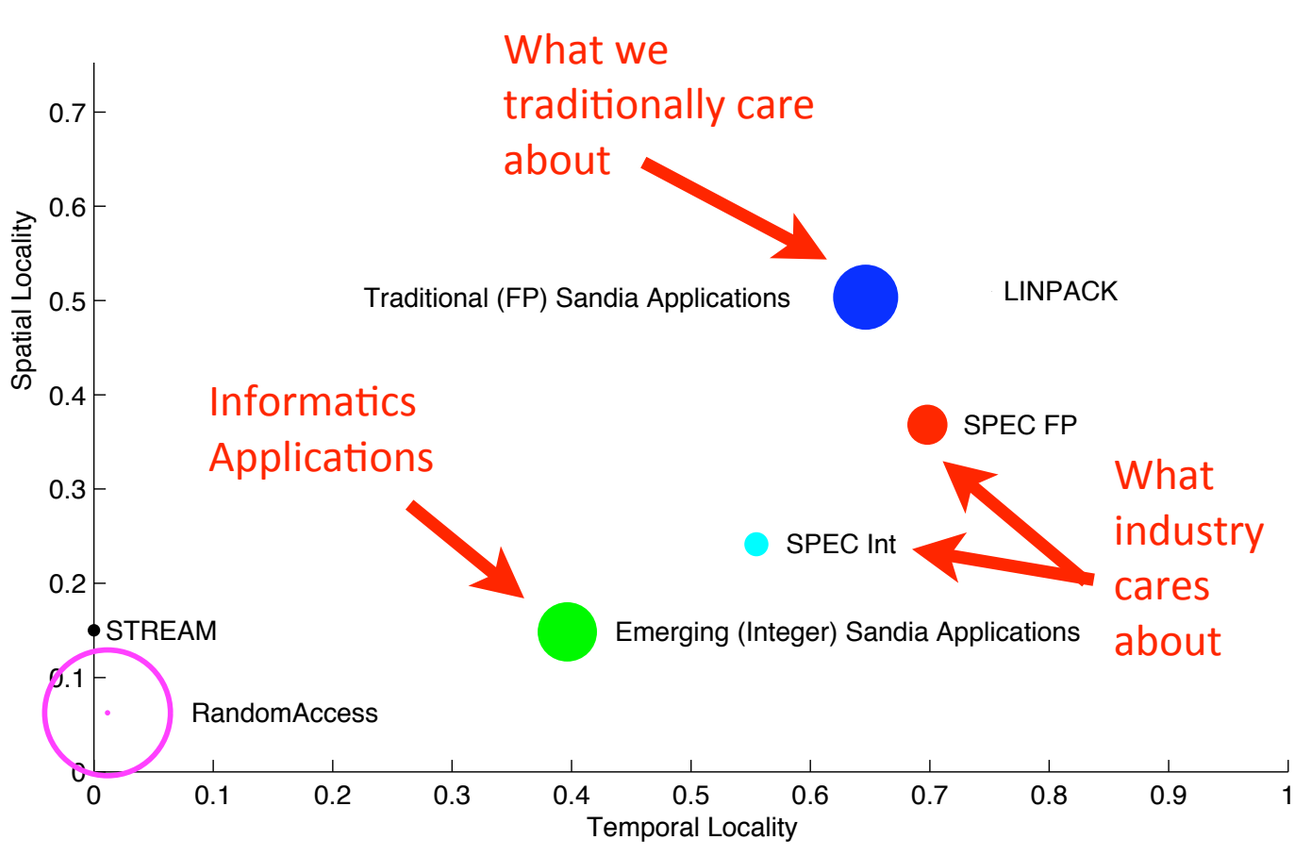




Death from Above: Applications

Sandia National Laboratories

- **Huge variety in programming models and run-times emerging**
  - Evolutionary BSP-originated applications
  - Revolutionary programming models
    - Everyone's got one, and they're all the best
    - 101+ actively developed parallel programming languages
  - Lots of new application varieties
  - Flexibility is key
- **Multiple optimization points**
  - Time to solution
  - Energy to solution
  - Money to solution
  - Total system efficiency

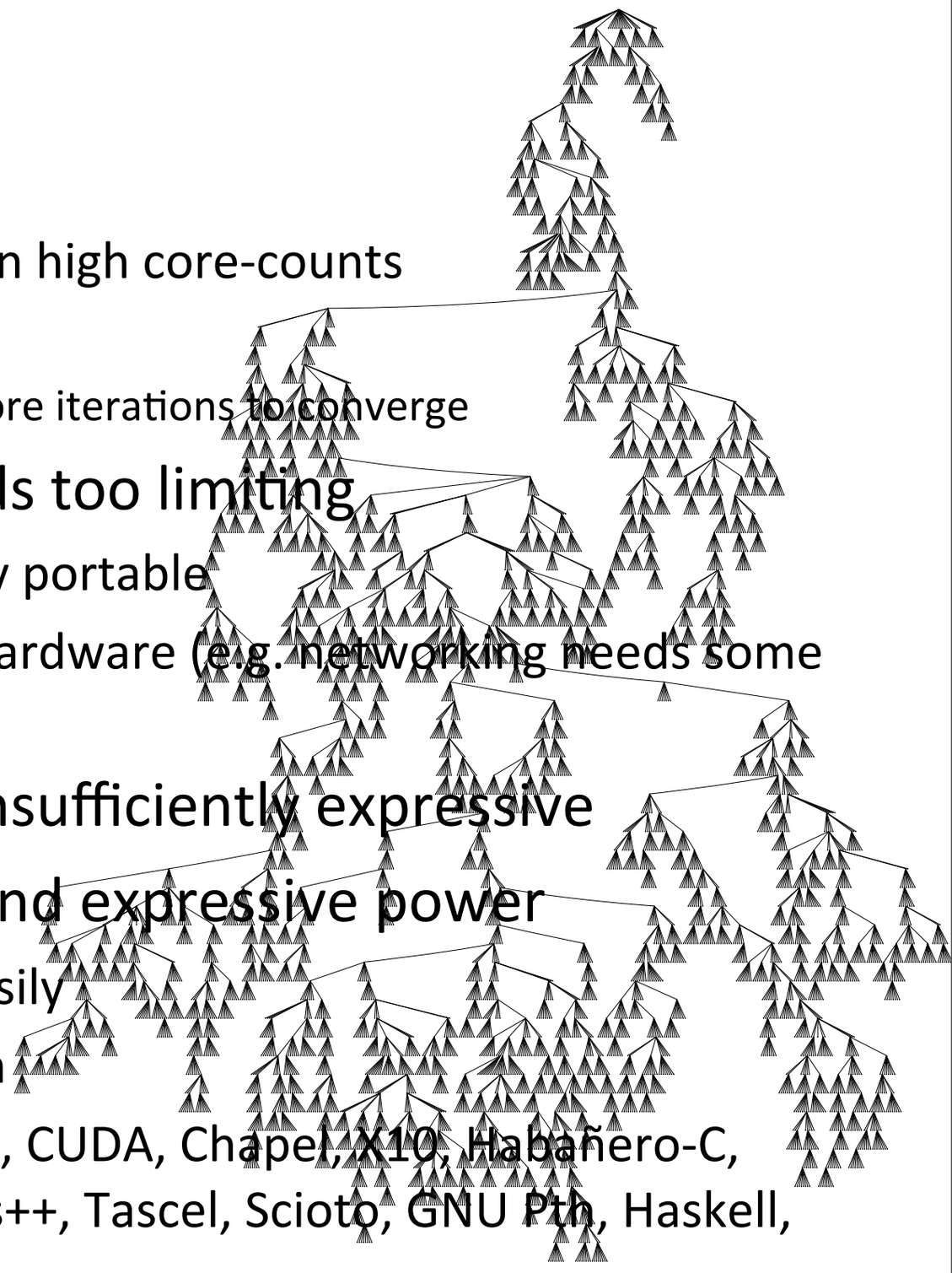


From: Murphy and Kogge, *On The Memory Access Patterns of Supercomputer Applications: Benchmark Selection and Its Implications*, IEEE T. on Computers, July 2007

Tasks to the rescue?

Sandia National Laboratories

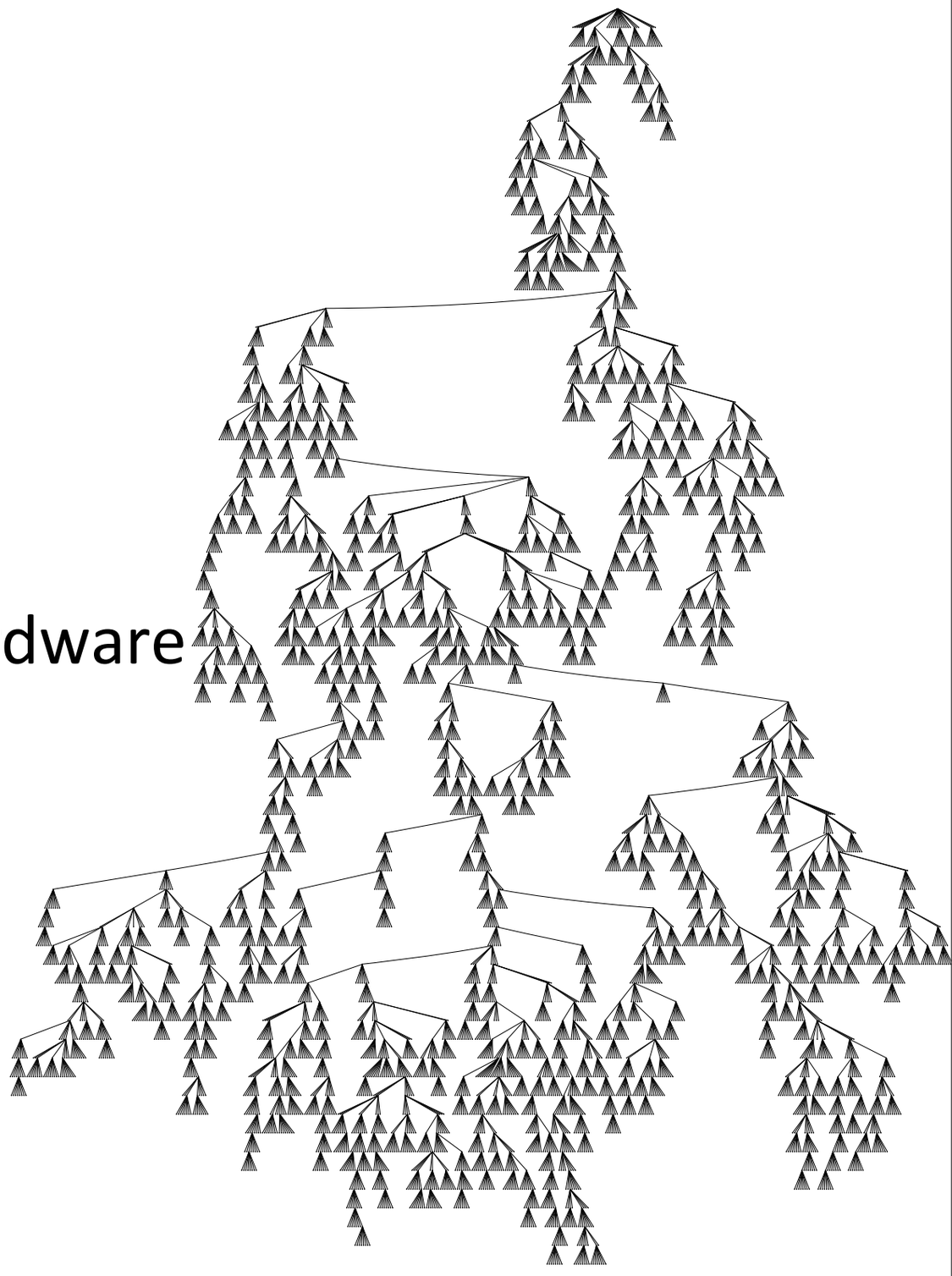
- Shared-memory node-level parallelism
  - Important for avoiding over-decomposition in high core-counts
    - CTH time-step length vs block size
    - Preconditioner effectiveness drops, requiring more iterations to converge
- Designing to specific numbers of threads too limiting
  - Core counts change too quickly, insufficiently portable
  - Does not react well to changes in available hardware (e.g. networking needs some CPU time)
- Loop-based parallelism powerful, but insufficiently expressive
- Task programming provides flexibility and expressive power
  - Handles loops and other parallel contexts easily
  - Scales with the level of expressed parallelism
  - Popular mechanism: OpenMP, Cilk, Intel TBB, CUDA, Chapel, X10, Habañero-C, Concurrent Collections, SWARM, HPX, Nanos++, Tascel, Scioto, GNU Pth, Haskell, MassiveThreads, etc.



Hetero Hardware to the Rescue!

Sandia National Laboratories

- Often task-based
  - NVidia GeForce GTX
  - AMD HSA
  - Intel Xeon Phi
  - Convey CHOMP
- Directly address overheads with hardware
  - Cheap synchronization
  - Cheap task creation
  - Cheap context swap
  - Cheap scheduling
  - Inherent load-balancing



Issues in Task Parallelism

Sandia National Laboratories

- Overhead
  - Time spent creating, scheduling, synchronizing, load-balancing rather than computational work
  - ***SERIOUS*** barrier to adoption
- Programming Interface
  - Every platform has a unique one
  - Programmer usually encouraged to over-constrain execution
- Locality
  - Task execution time depends on data access time AND HARDWARE
  - Data placements often depends on previous thread placement

Data Movement Dominates Costs!

Commodity HW/SW Locality

Sandia National Laboratories

- Shared Memory a double-edged sword
  - Data can be accessed without explicit operation
  - Cache coherency expensive and contention-dependent
  - Uneven costs & implicit communication lead to unexpected performance characteristics
- Locality considered a third-order optimization
  - OpenMP agnostic toward affinity of data and computation
    - Many vendors provide non-portable extensions
    - Many developers use third-party libraries and make assumptions about OpenMP implementation details
    - First-touch memory distribution typical on Linux
      - Requires data "conditioning"
      - Cannot adapt to changing load
  - OpenACC too explicit about data placement
  - Chapel agnostic toward intra-node data/computation affinity

The Challenge of Heterogeneity

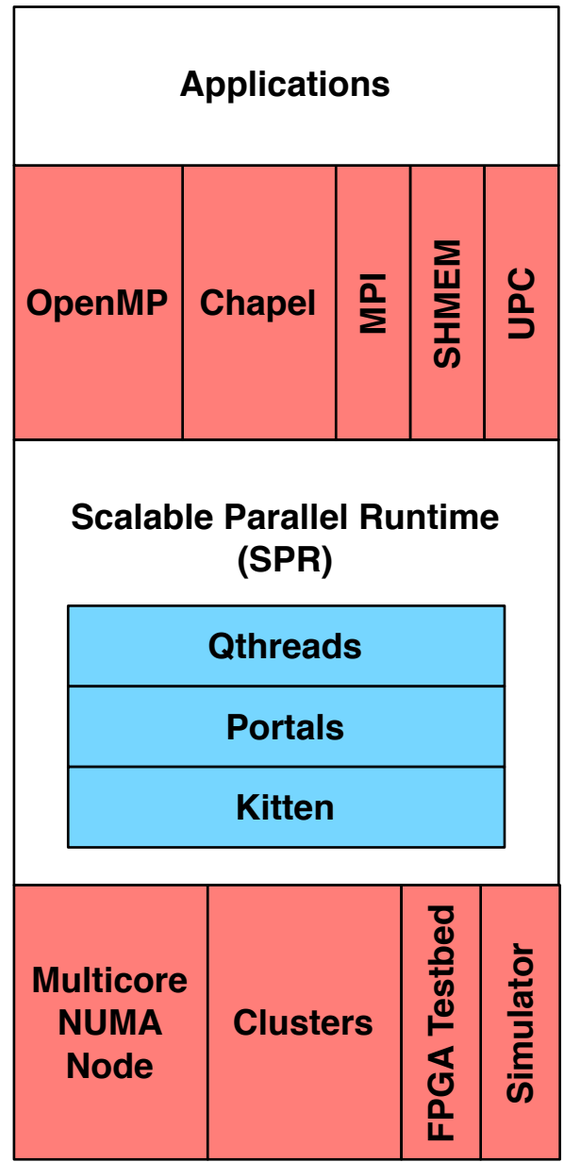
Sandia National Laboratories

- Creates more locality challenges
- Where is the data?
  - What format? Endianness? Bit-width?
  - Participation in cache coherency?
  - State?
  - Synchronization?
- Where is the computation?
  - “Best” location vs “available” locations
  - Granularity?
    - How to load-balance?
  - Portability?

Scalable Parallel Runtime (SPR)

Sandia National Laboratories

- **Kitten: Lightweight OS kernel**
  - Builds on lessons from ASCI Red, Cplant, Red Storm
  - Utilizes scalable parts of Linux environment
  - Primarily supports direct hardware mapping
- **Portals 4: Lightweight communication interface**
  - Semantics for supporting both one-sided and tagged message passing
  - Small set of primitives, allows offload from main CPU
  - Supports direct hardware mapping
- **Qthreads: Lightweight threading interface**
  - Scalable, lightweight scheduling on NUMA platforms
  - Supports a variety of synchronization mechanisms, including full/empty bits and atomic operations
  - Potential for direct hardware mapping

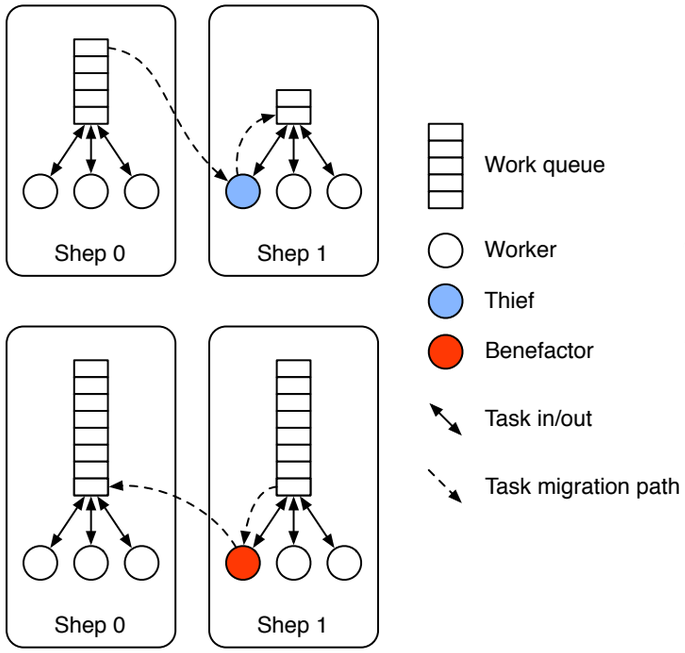


Open, Optimized Component Technologies

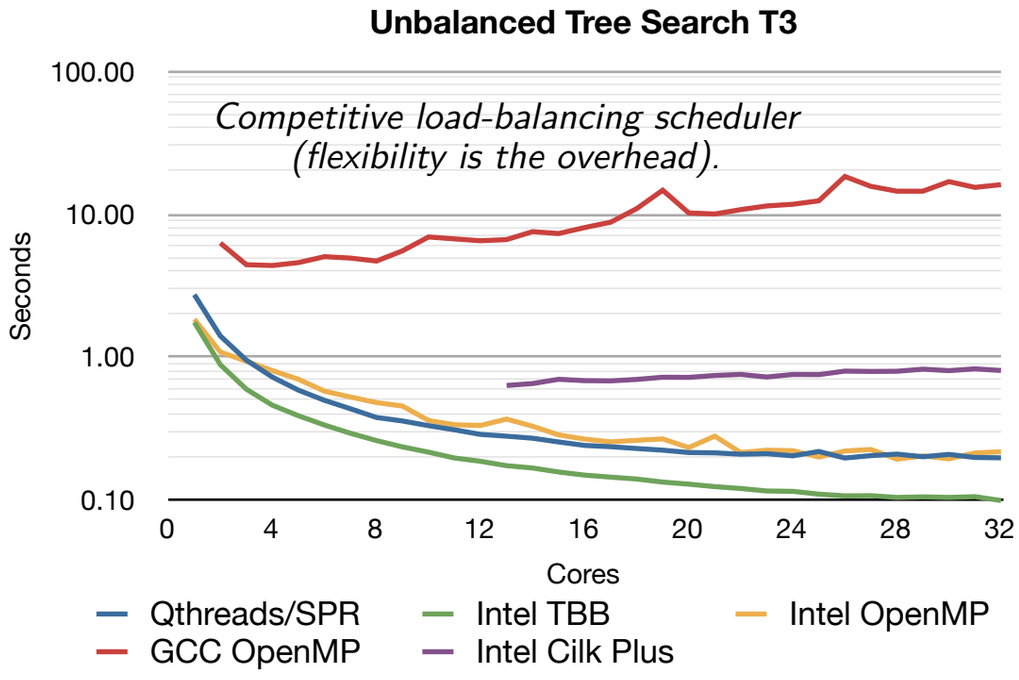
Qthreads Lightweight Threading

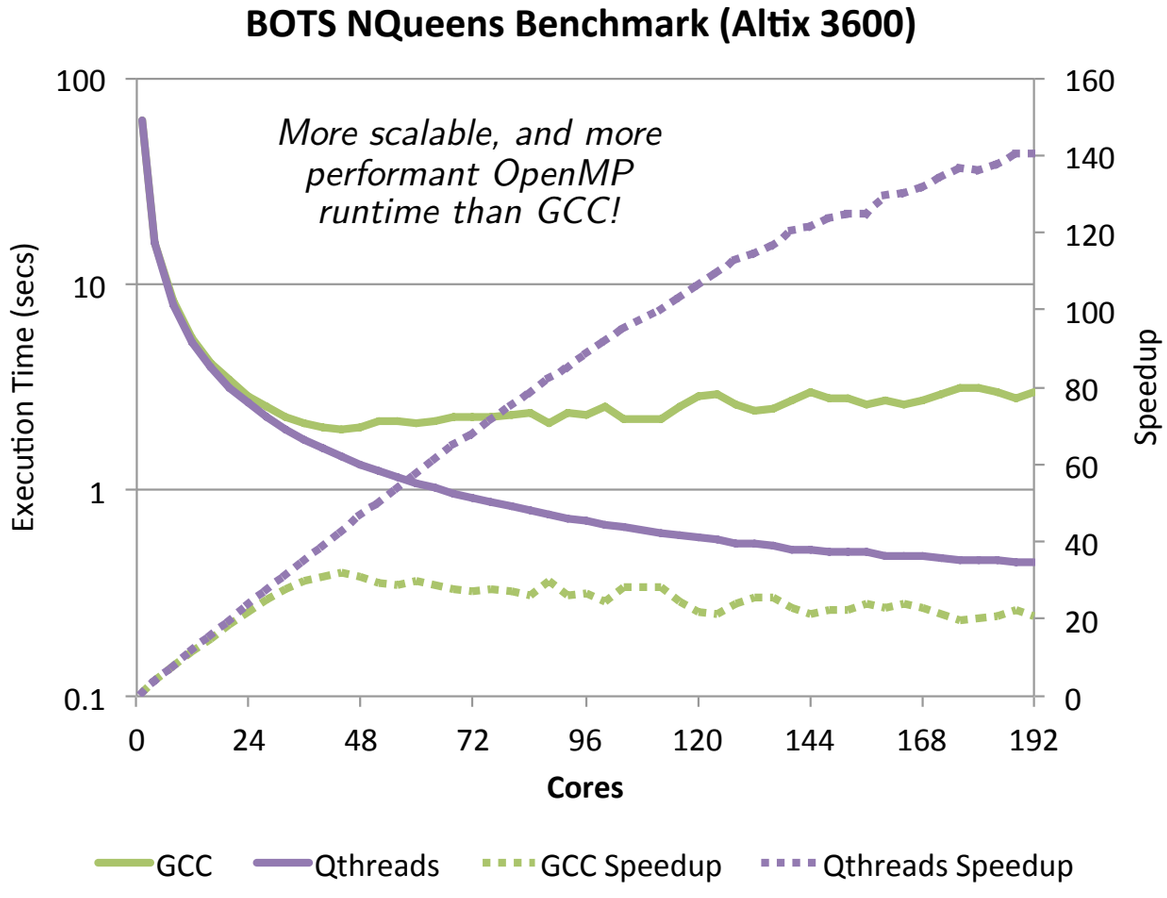
Sandia National Laboratories

- **Advanced task-based runtime**
  - Tool for programming model research
  - Supports both **OpenMP**-like models and more complex **Chapel**-like models
  - Presents simplified model of system to the application
  - High-performance scheduler
- **Current R&D**
  - Priority scheduling
  - Remote task launch efficiency
  - Efficient, flexible collective operations



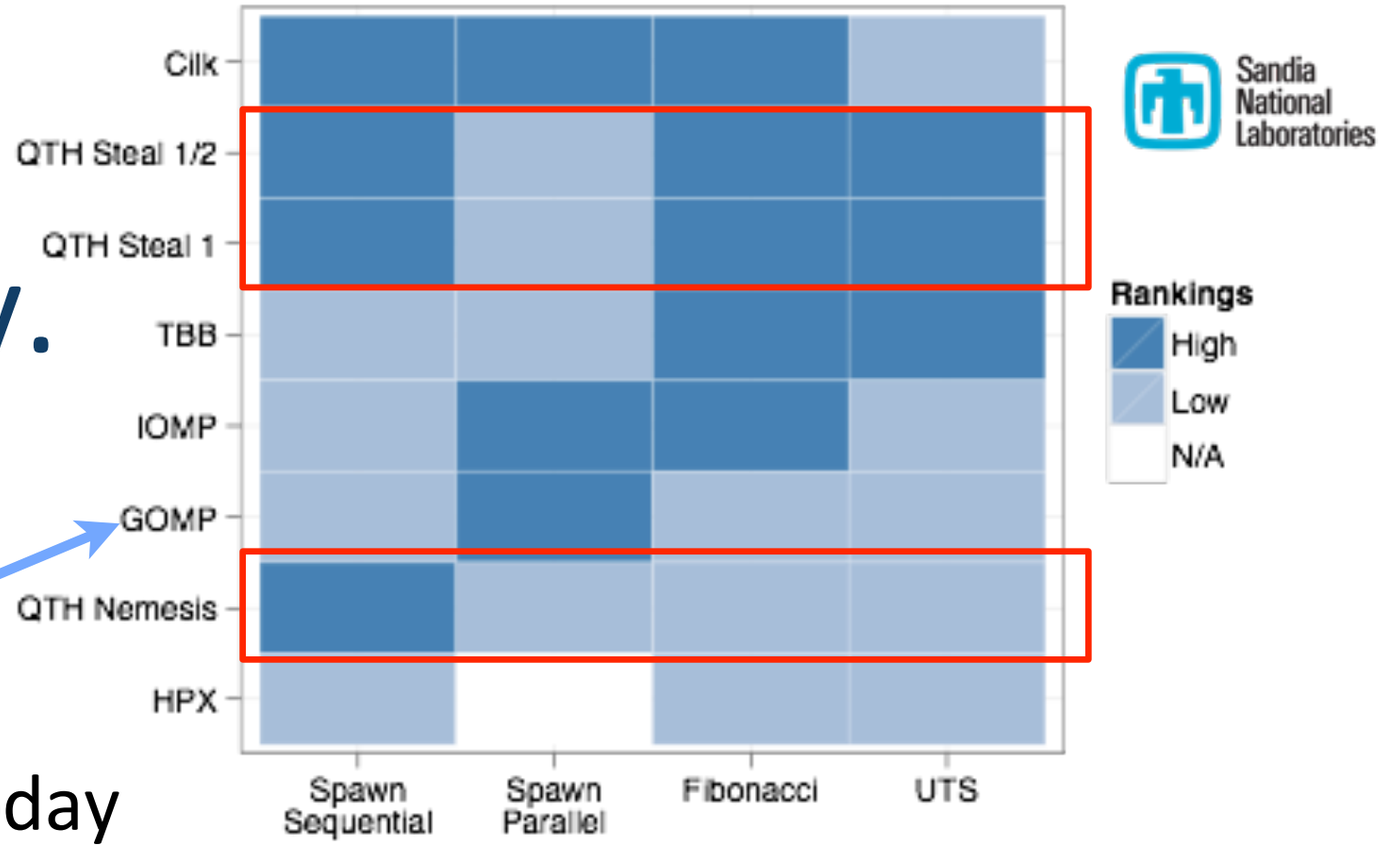
High performance "sherwood" work-stealing scheduler effectively balances cache efficiency with load balancing.

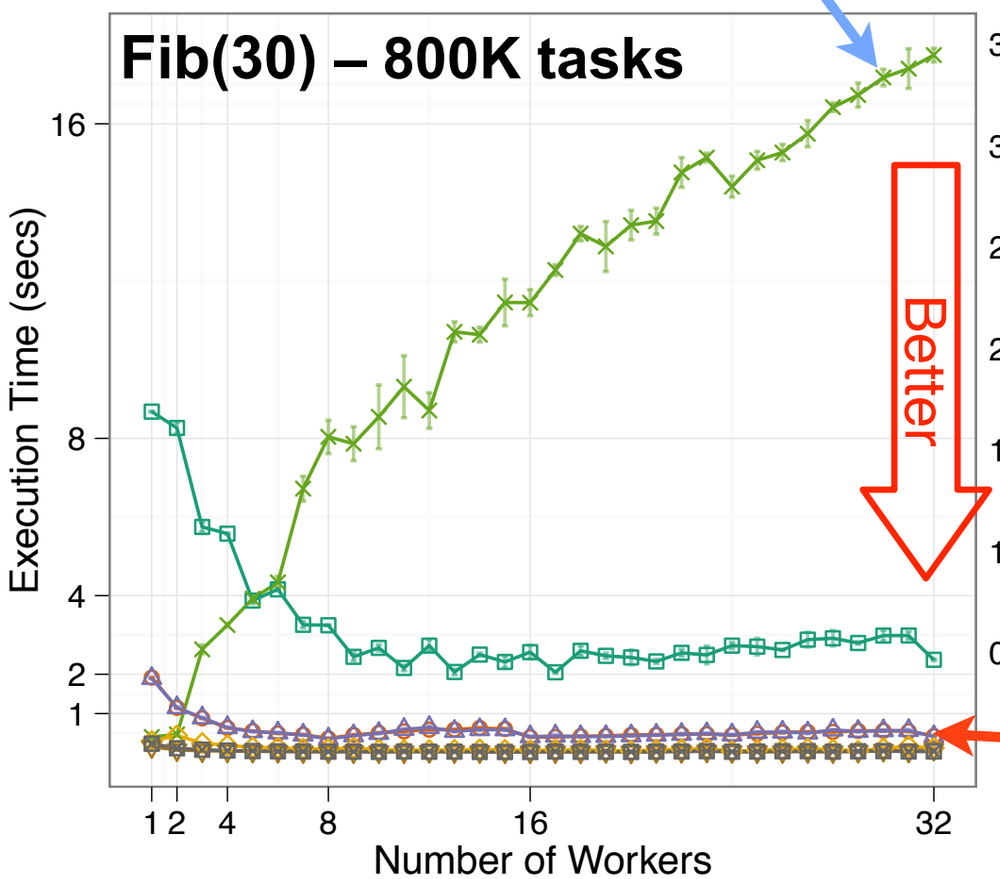


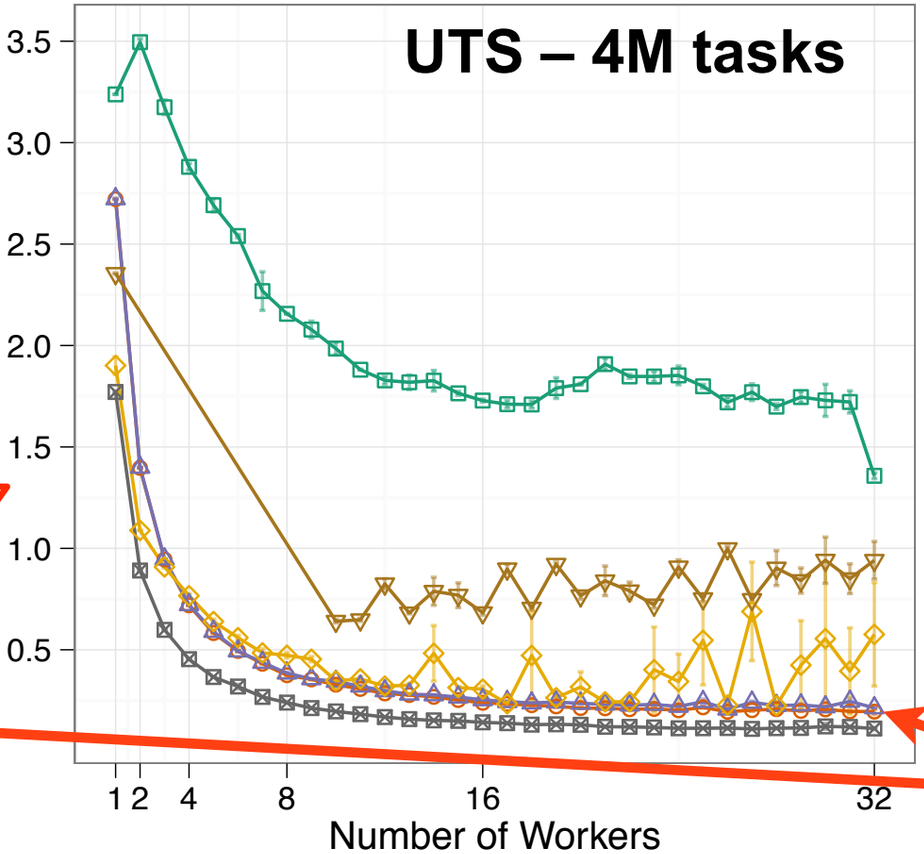


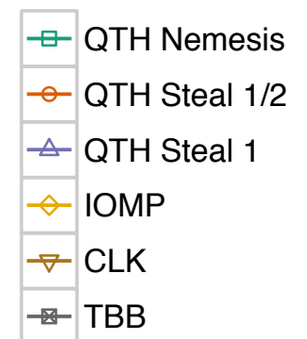
A ***Vehicle*** for Research *and* Dev.

- Ongoing runtime research
  - Highly configurable
  - Not one-size-fits-all
- Performance-competitive today









Qthreads/ SPR

Heterogeneous Features

Sandia National Laboratories

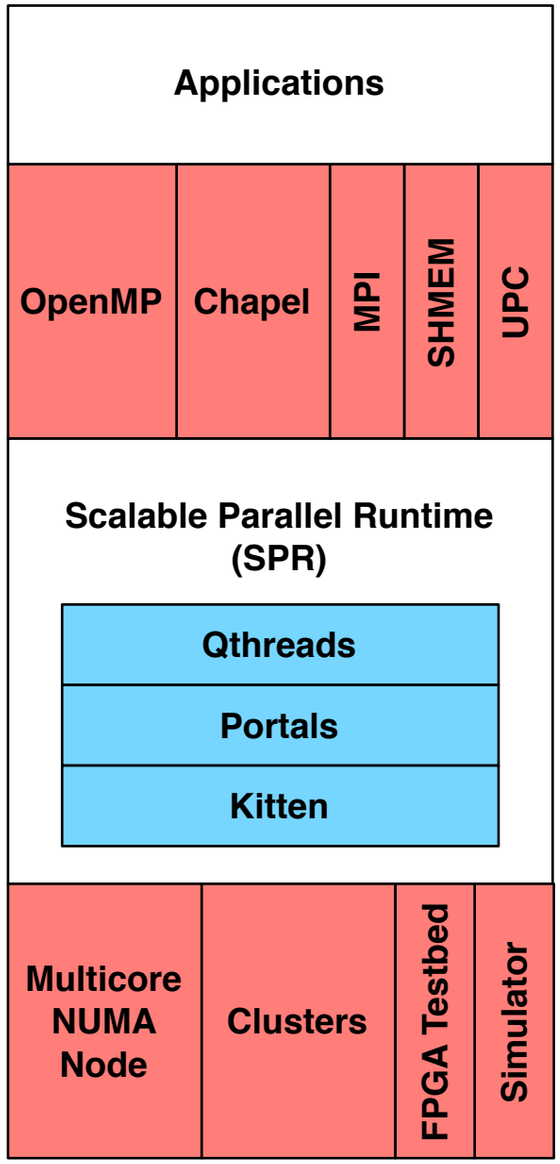
- Inter-node Heterogeneity
  - Communicate via Portals4
  - Function Registration
  - Supports both SPMD and MIMD
  - Communication and task-spawn explicit
  - Programmer's responsibility to ensure data-format portability

- Intra-node Heterogeneity
  - Task aggregation & offload
    - Aggregation has a cost (increased contention)
      - User-configurable
    - Not all tasks aggregatable
      - e.g. only identical tasks
  - Not yet finished

Open Research Questions!

Sandia National Laboratories

- **How will threads evolve to be more lightweight and match hardware semantics?**
  - What will hardware threading semantics be?
- **What synchronization primitives are necessary for highly asynchronous applications?**
  - Fast, Free, Infinite
- **What memory consistency models are necessary?**
  - … or even useful?
- **What communication primitives are necessary for evolving applications?**
  - Probably not six-function MPI



SPR is an Experimental Platform for Exascale R&D

Sandia National Laboratories

http://code.google.com/p/qthreads

http://code.google.com/p/portals4

http://code.google.com/p/kitten

THANK YOU!

Runtime Architecture / Experimental Platform