Fault Oblivious eXascale Whitepaper

Eric Van Hensbergen, Ronald G. Minnich,
Curtis L. Janssen, Sriram Krishnamoorthy,
Andres Marquez, Maya Gokhale, P. Sadayappan, Jonathan Appavoo,
Jim McKie

Motivation

- Fault tolerance crucial at scale
- Applications burdened with this problem
- Disjoint efforts further the challenge

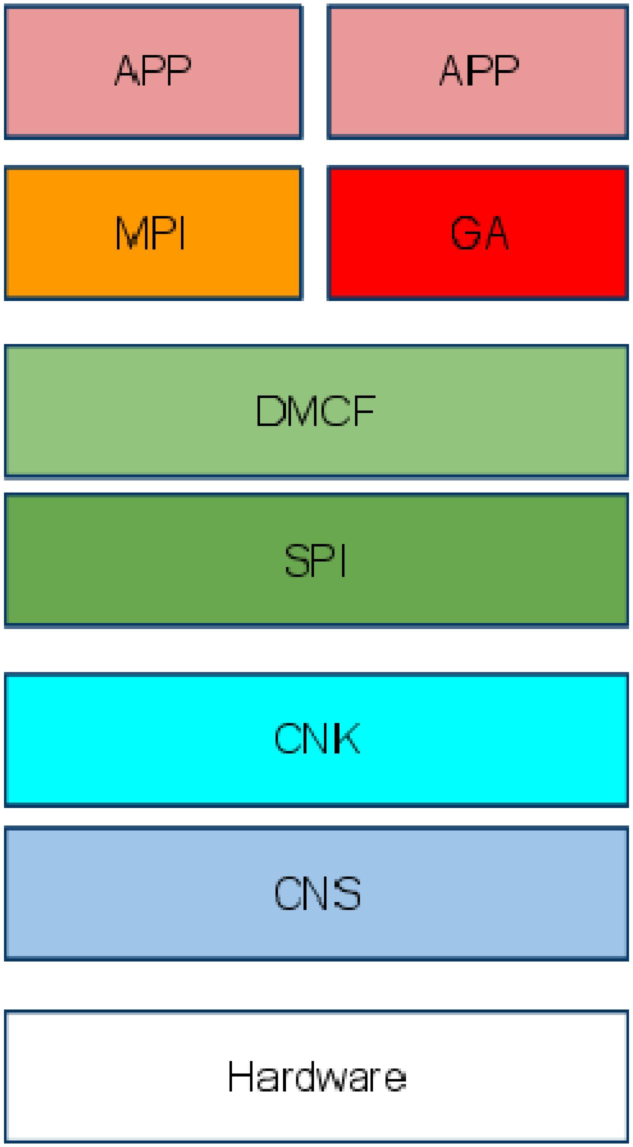
Approach

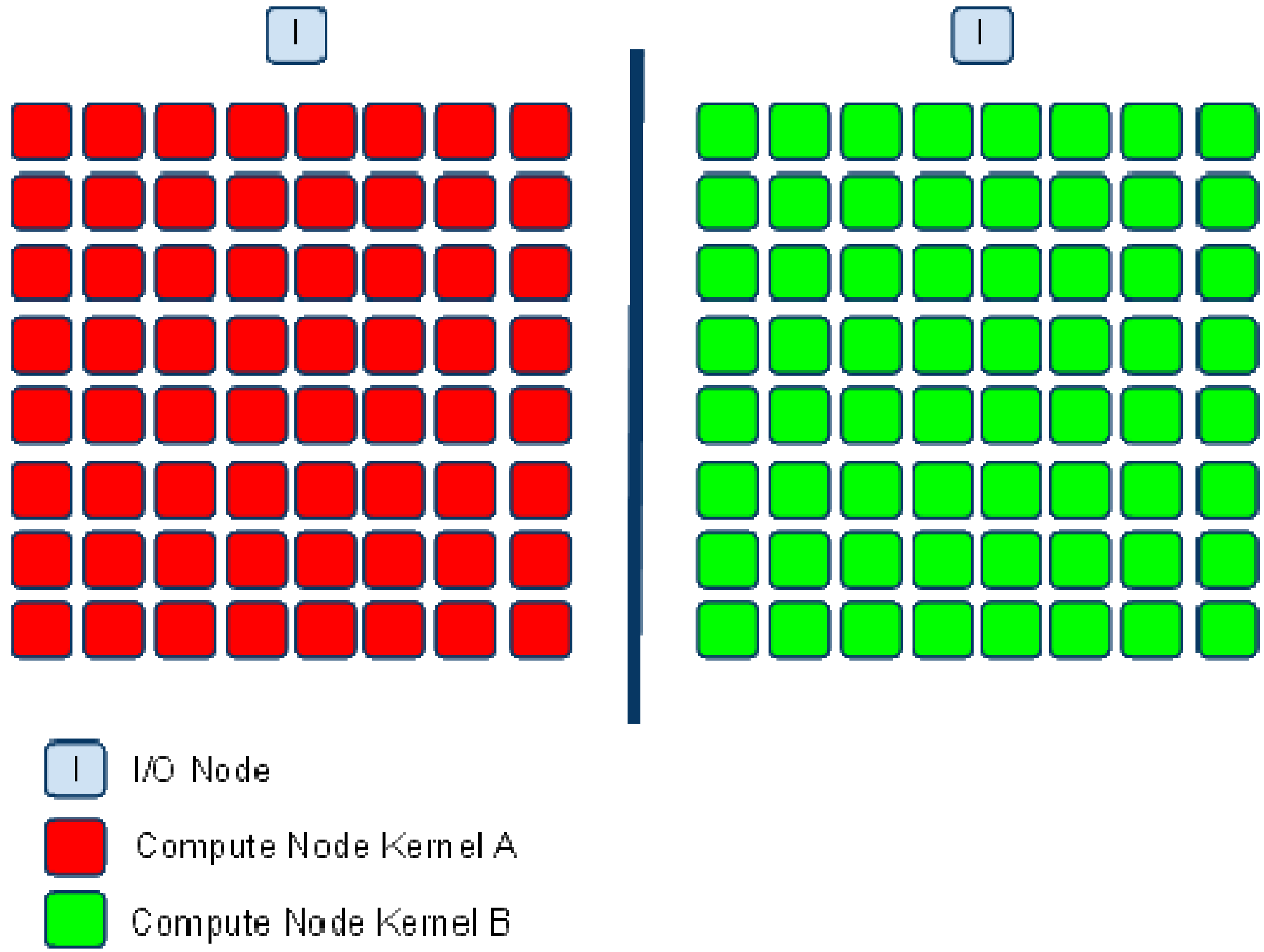
- Fault-Oblivious Execution Model
  - Everyone except the application work at tolerating faults
- Integrated approach
  - Simulators, system s/w, runtime, and application
- Answer these questions:
  - What can be done in an oblivious-fashion?
  - How?
  - Which layer does what?
- Impact: application arch.; runtime design; role of system s/w

Research Elements

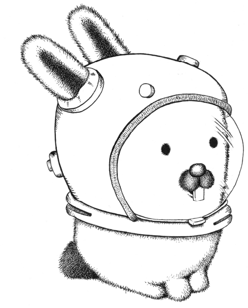
- Elastic System software
- Data-Driven Execution and Recovery
- Application Validation

Current Systems



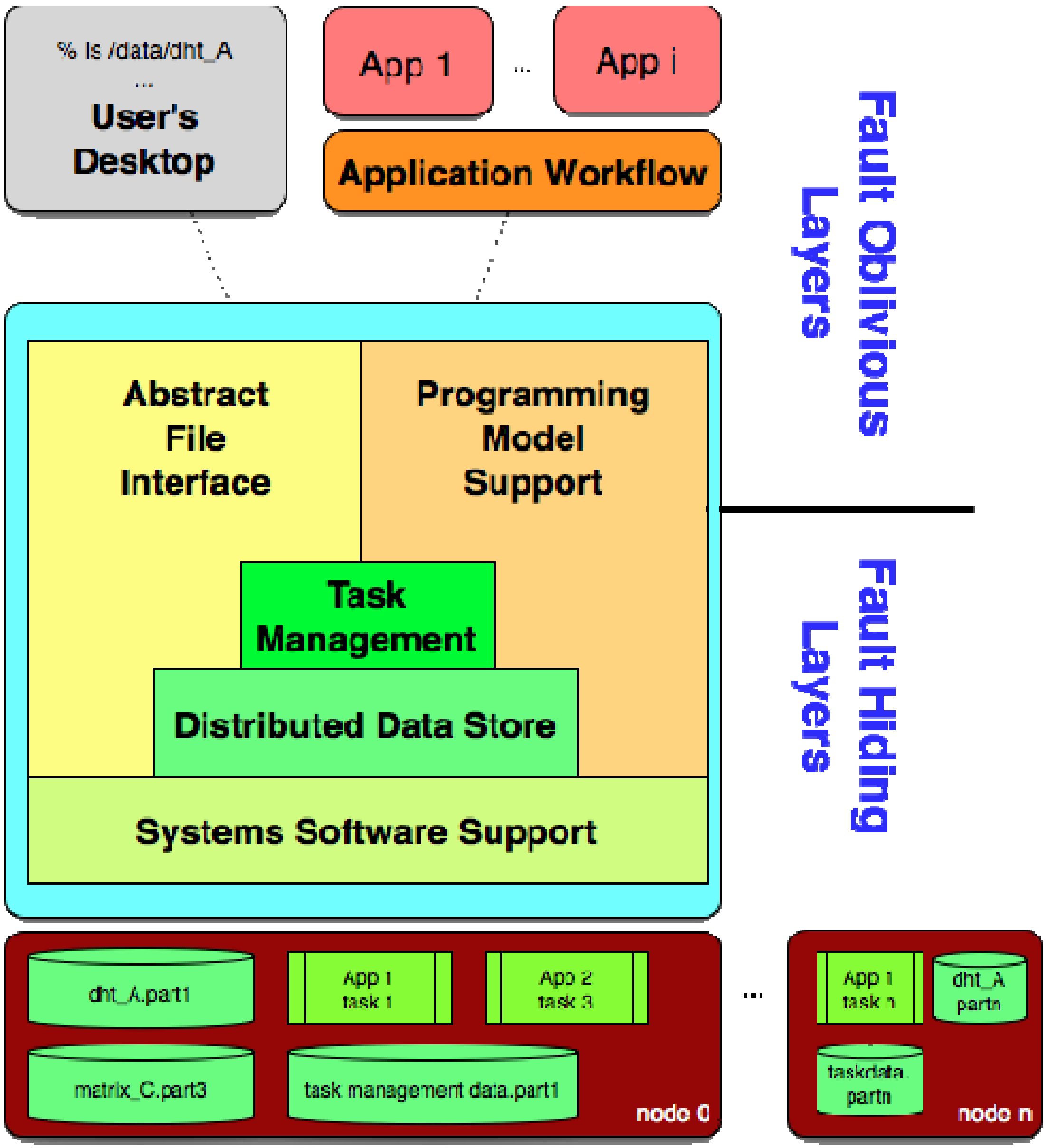


Prior Research: HARE & Right Weight Kernels

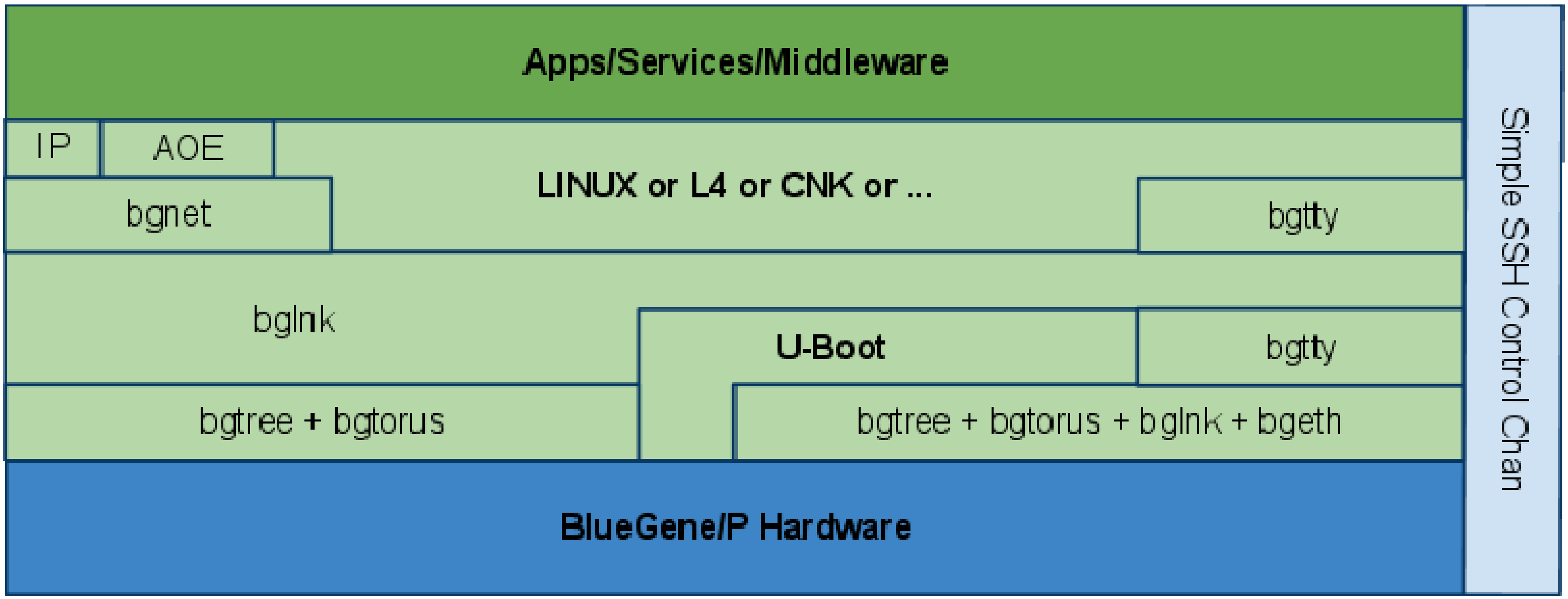


- Look for a middle ground between light-weight kernels and general purpose kernels
- Start with a small, simple general purpose kernel and adapt it to the performance characteristics of HPC
- Choose an operating system that is already a distributed system to match the distributed nature of HPC systems and clusters
- Challenge the notion that latency requirements require OS bypass on HPC systems
- Look into architected models of providing high-performance operations to userspace
- Investigate alternative runtimes to MPI by looking at resillient models from the dataflow applications community
- More information: http://goo.gl/WgDGa

FOX Operating System Framework



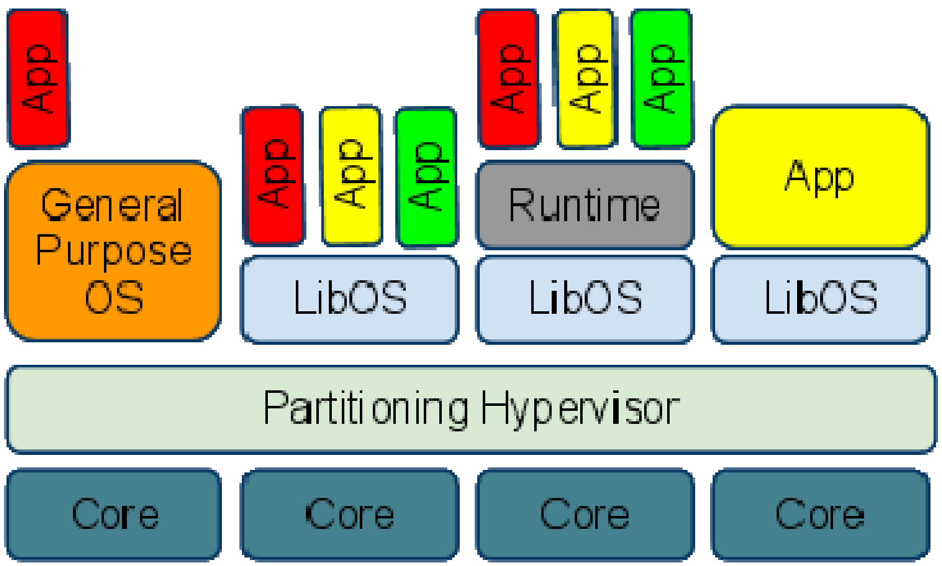
Kittyhawk Cloud Layer



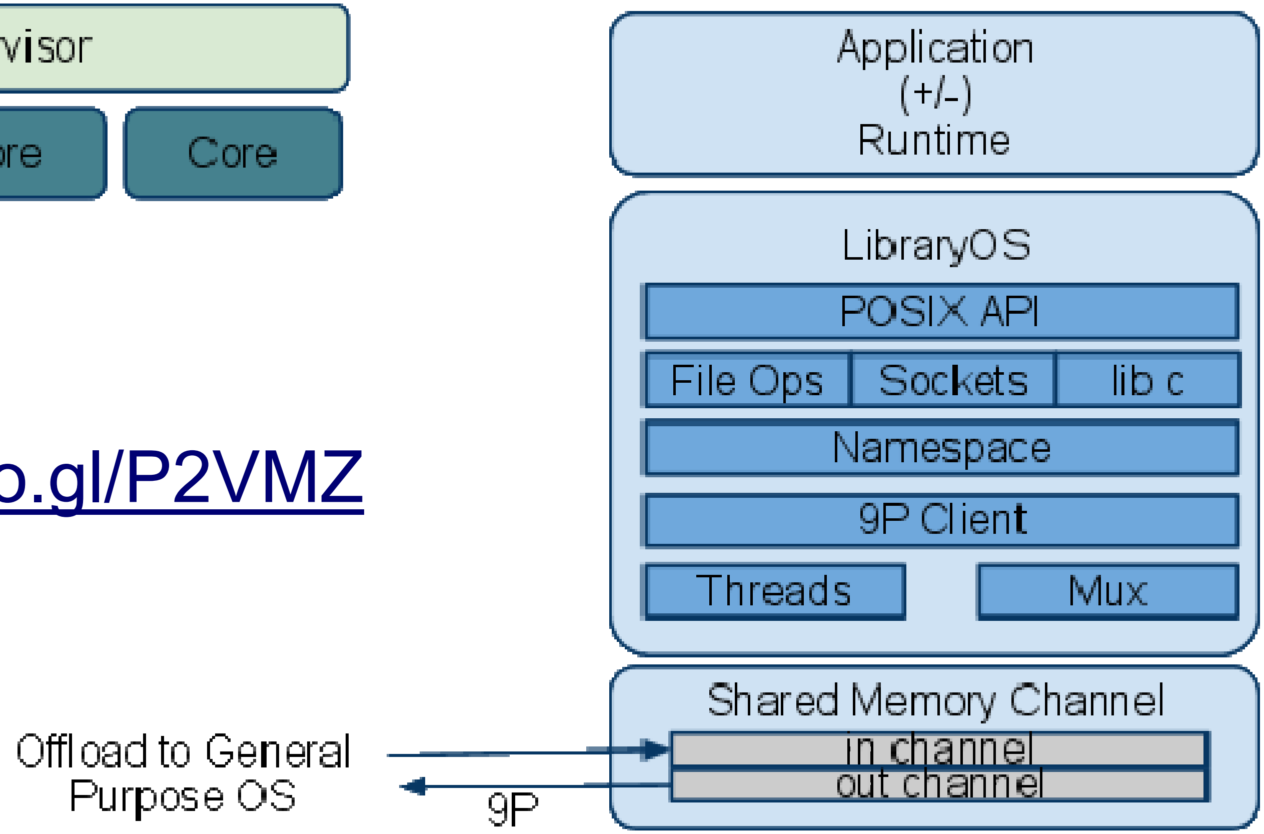
- J. Appavoo, V. Uhlig, J. Stoess, A. Waterland, B. Rosenburg, R. Wisniewski D. Da Silva, E. Van Hensbergen, and U. Steinberg, "Providing a Cloud Network Infrastructure on a Supercomputer", Science Cloud 2010: 1st Workshop on Scientific Cloud Computing, June 2010.
- More Info: http://kittyhawk.bu.edu

LibraryOS

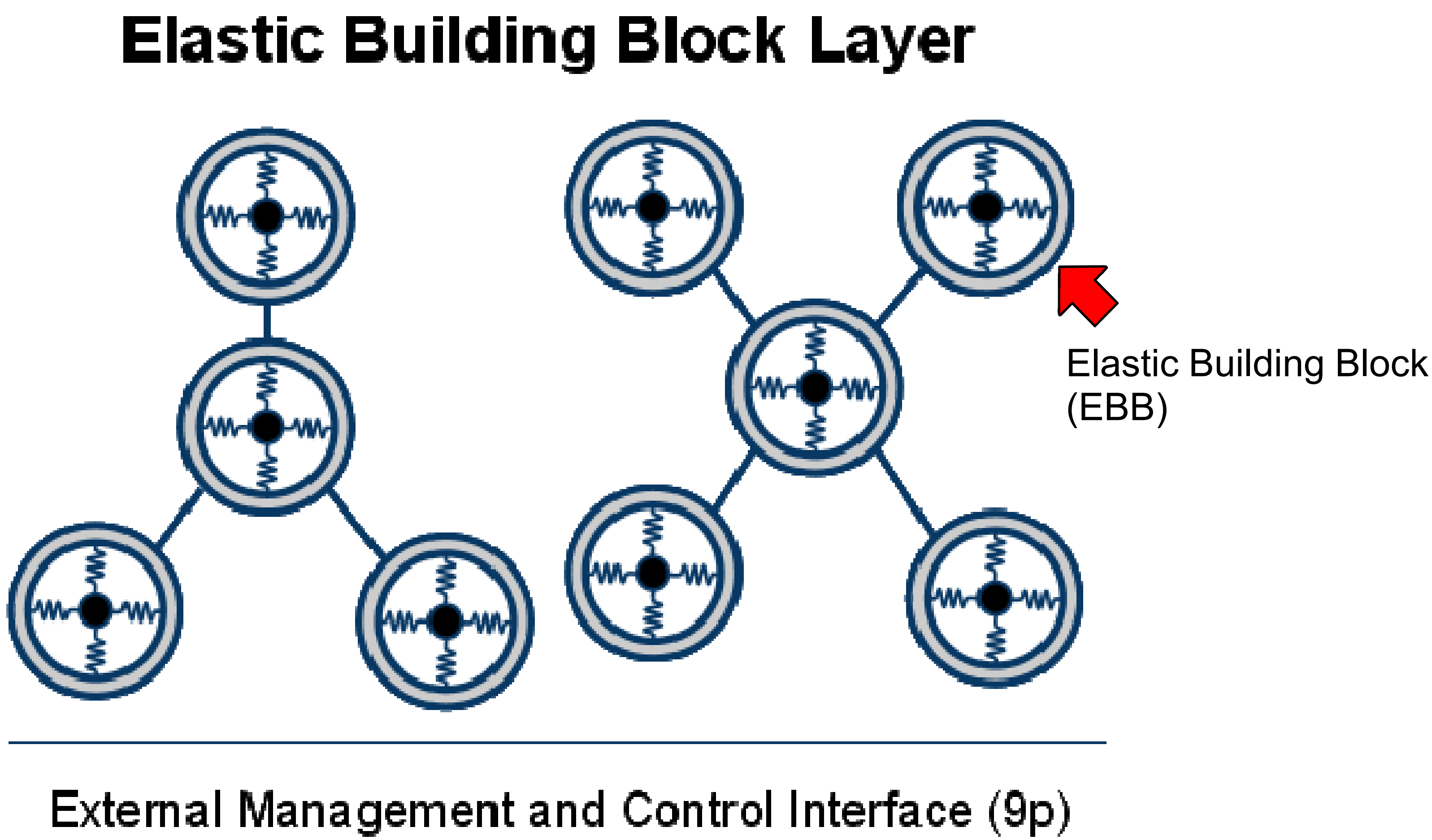
- [Libra:a library operating system for a jvm in a virtualized execution environment] by G.Amons et. al. Proc. of the 3rd international conference on virtual execution environments. 2007.
- ["Specialized Execution Environments: The Future of Systems Software"] by E. Van Hensbergen, D. Da SIlva, O. Krieger, M. Ostrowski, B. Rosenburg, and J.Xenidids. SIGOPS Operating Systems Review. Jan 2008.
- ["Partitioned Reliable Operating System Environment"] by E. Van Hensbergen. Operating Systems Review, April 2006
- ["The Effect of Virtualization on OS Interference"] by E. Van Hensbergen at 1st OSIHPA Workshop, 2005



More Info: [http://goo.gl/P2VMZ]



Scalable Elastic Systems Architecture



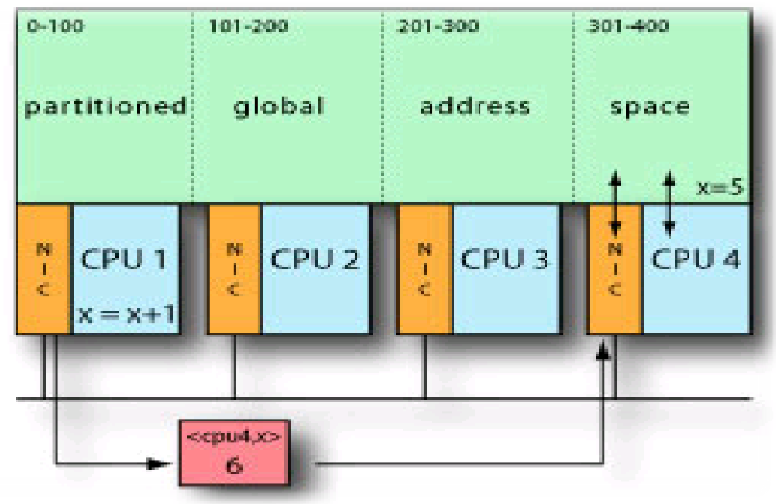
- **Scalable Elastic Systems Architecture**. Dan Schatzberg, Jonathan Appavoo, Orran Krieger, Eric Van Hensbergen, *ASPLOS RESoLVE Workshop*, ACM, 2011.

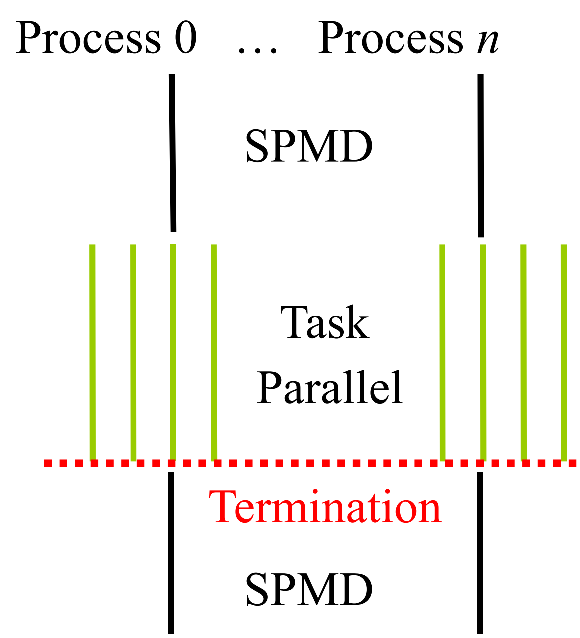
Research Elements

- Elastic System software
- Data-Driven Execution and Recovery
- Application validation

Task-Based Execution

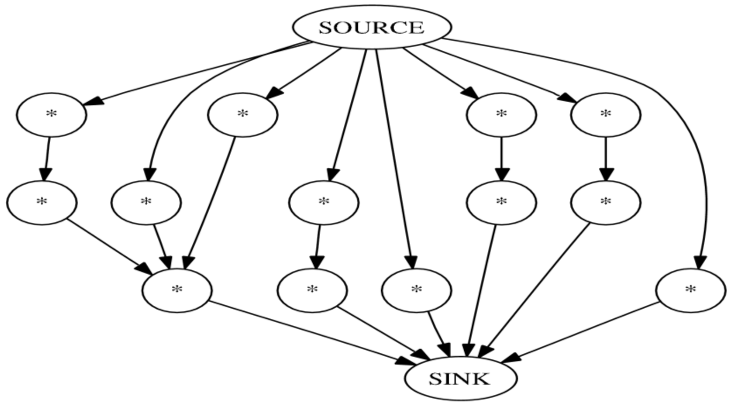
- Computation – collection of tasks
  - Data in global address space
- Tasks
  - Portable function pointer
  - Argument and dependence information
- Data – accessible by any process
- Runtime system
  - Schedules communication
  - Manages memory
  - Fault tolerance

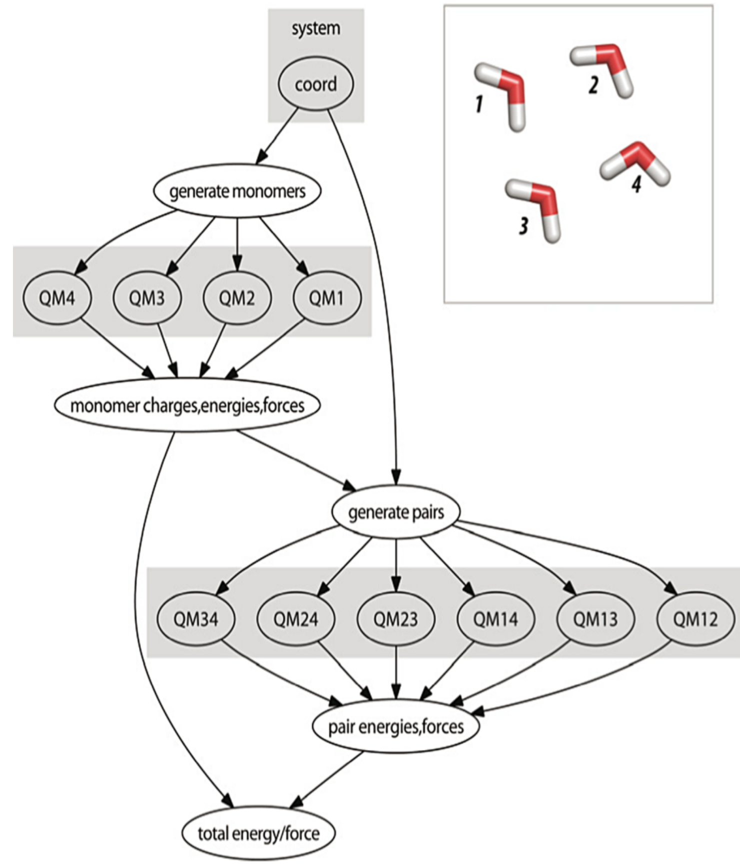


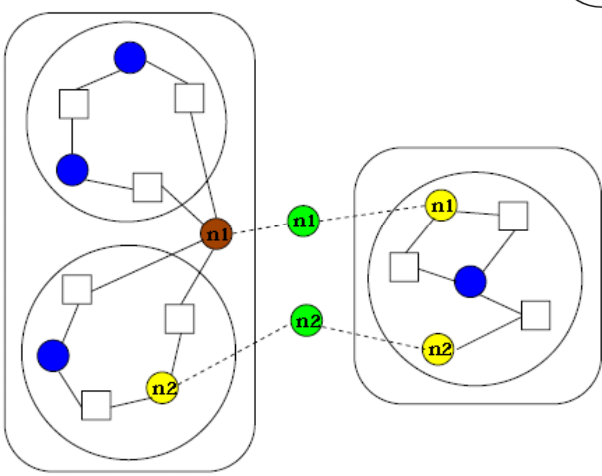


Data and Computation Abstractions for Dynamic and Irregular Computations. S. Krishnamoorthy, J. Nieplocha, P. Sadayappan. HiPC 2005

Prior Research: Scheduling

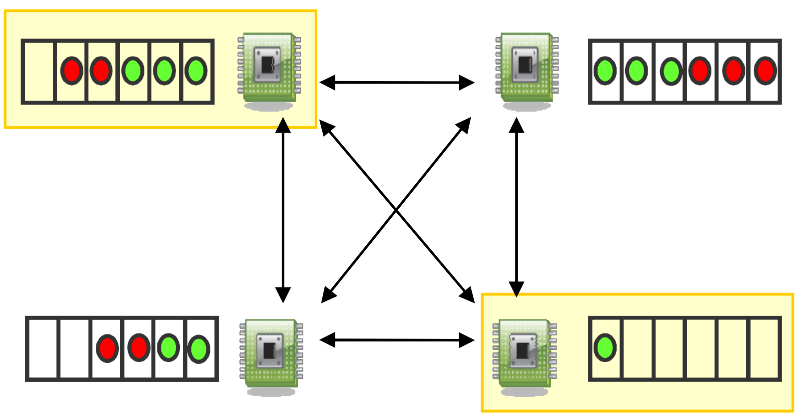


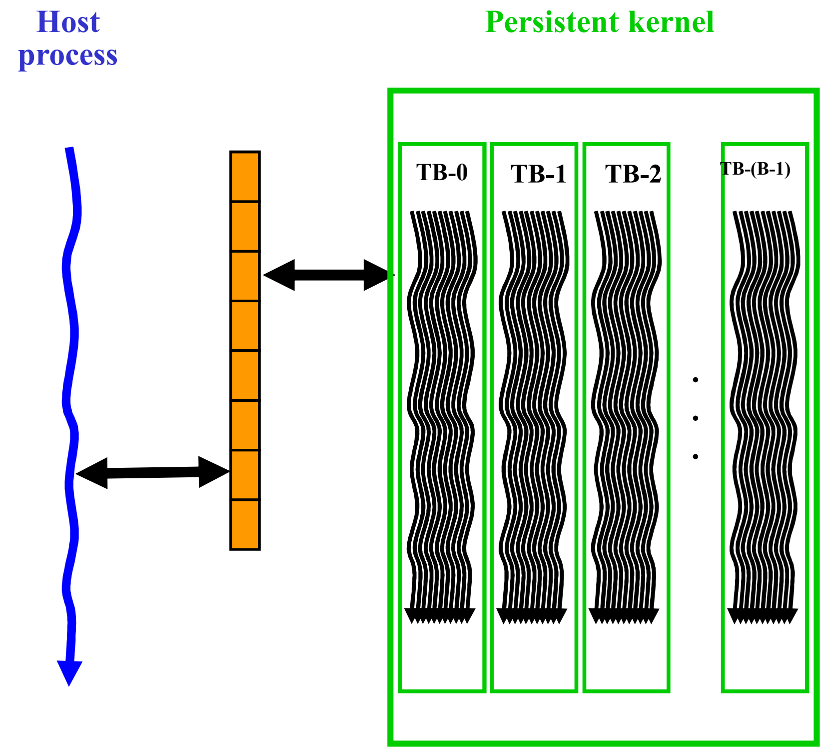




Hypergraph Partitioning for Automatic Memory Hierarchy Management. S. Krishnamoorthy, U. Catalyurek, J. Nieplocha, and P. Sadayappan. SC 2006
An Integrated Approach to Locality-Conscious Processor Allocation and Scheduling of Mixed-Parallel Applications. N. Vydyanathan, S. Krishnamoorthy, G.M. Sabin, U.V. Catalyurek, T.M. Kurc, P. Sadayappan, J.H. Saltz. TPDS 2009
Integrated Data and Task Management for Scientific Applications. J. Nieplocha, S. Krishamoorthy, M. Valiev , M. Krishnan , B. Palmer, and P. Sadayappan. ICCS 2008

Prior Research: Dynamic Load Balancing





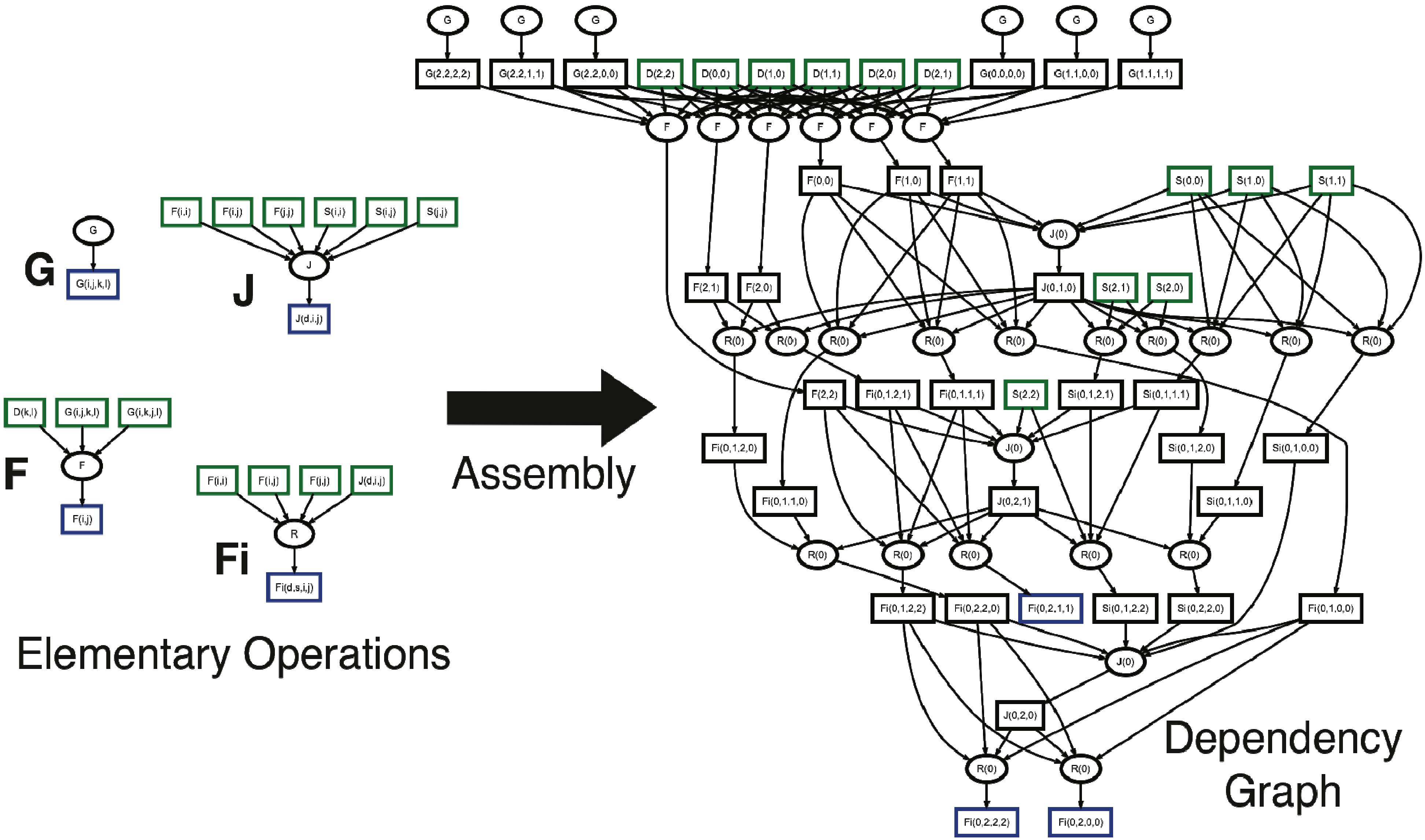
Solving large, irregular graph problems using adaptive work-stealing. G. Cong, S. Kodali, S. Krishnamoorthy, D. Lea, V, Saraswat, T. Wen. ICPP 2008

Scalable Work Stealing. J. Dinan, S. Krishnamoorthy, B. Larkins, J. Nieplocha, P. Sadayappan. SC 2009

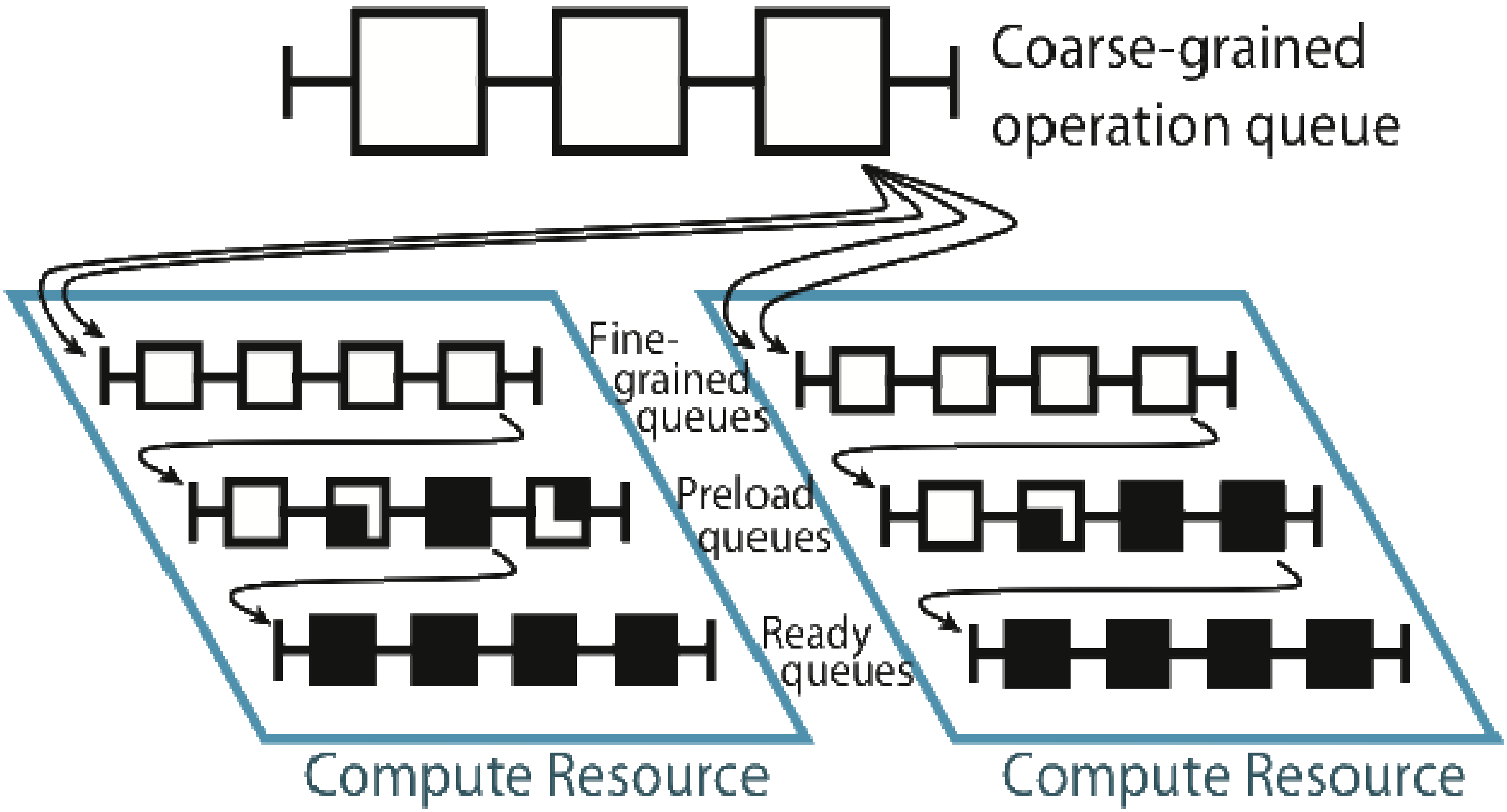
Load Balancing on Single- and Multi-GPU Systems. L. Chen, O. Villa, S. Krishnamoorthy, and G. Gao. IPDPS 2010

Lifeline-based Global Load Balancing. V. Saraswat, P. Kambadur, S. Kodali, D. Grove, S. Krishnamoorthy. PPoPP 2011

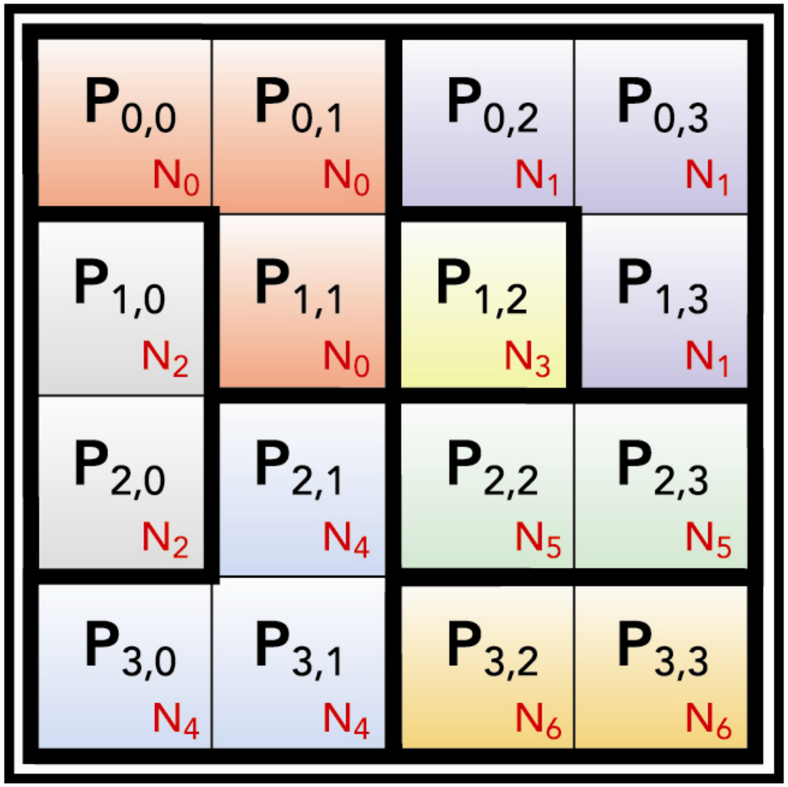
Data-driven Application Design



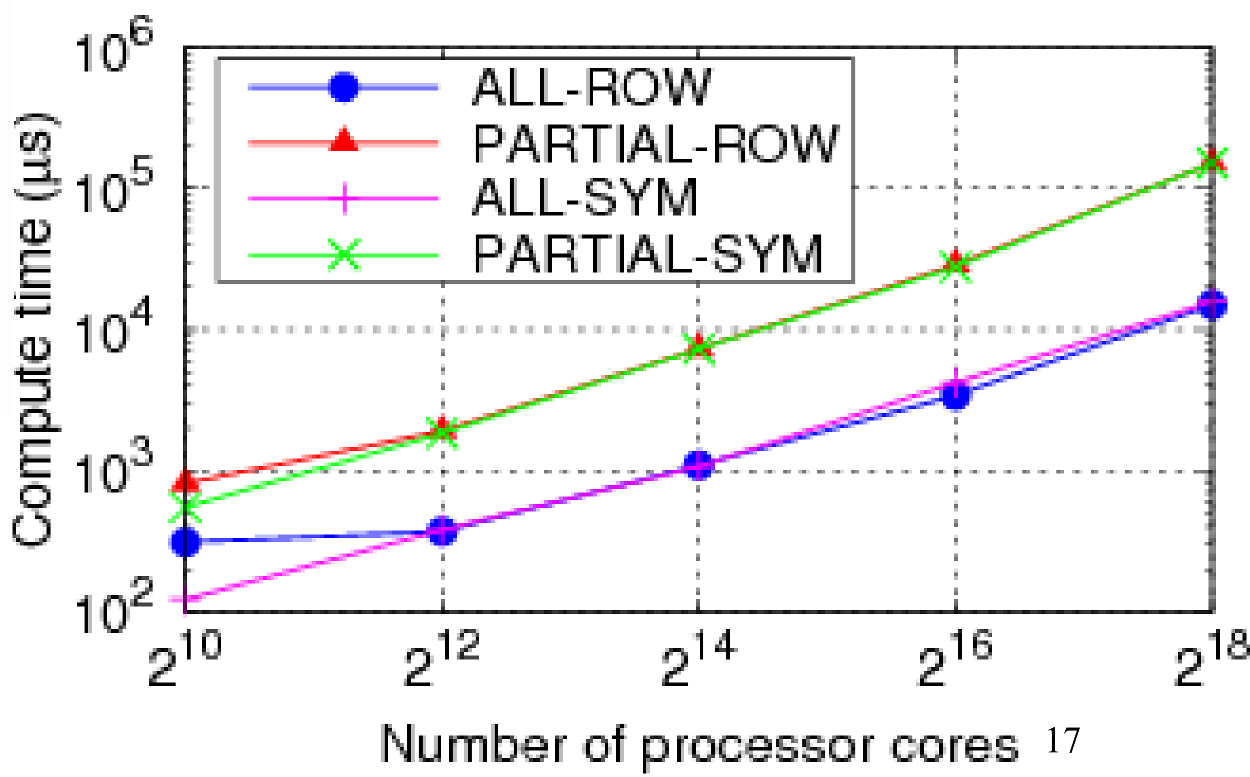
Hierarchical Task Queues



Fault-Tolerant Data Store



Tolerating Correlated Failures for Generalized Cartesian Distributions via Bipartite Matching. N. Ali, S. Krishnamoorthy, M. Halappanavar, J. Daily. Computing Frontiers 2011



Selective Restart

- Task-data relationship used to identify tasks affected by a loss
- Re-execute only tasks contributing to lost memory domains
  - Track communication
- Isolate impact of partial communication operations
- Re-execution by all processes – automatic rebalancing
- Learns from software transactional memory

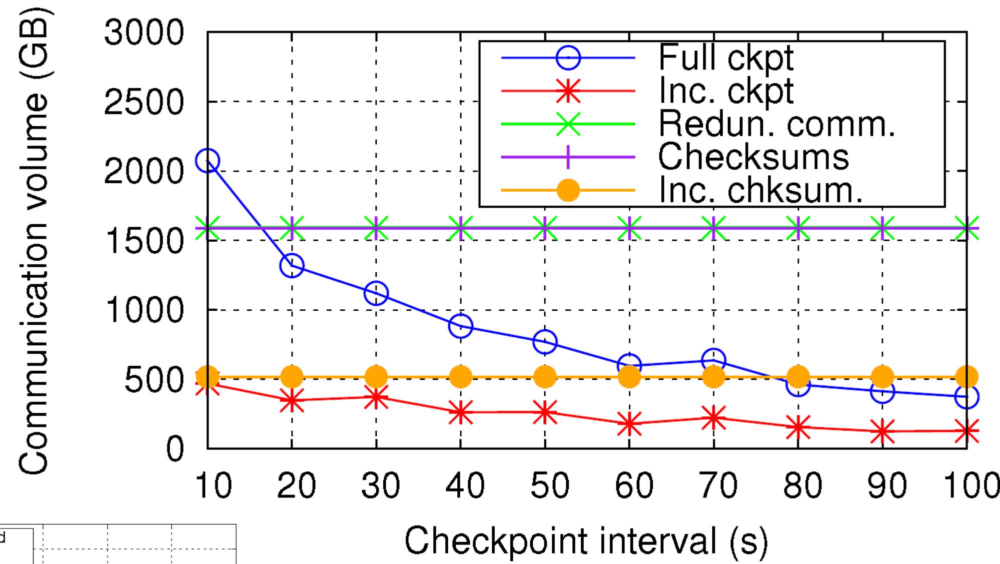
Selective Recovery From Failures In A Task Parallel Programming Model. J. Dinan, A. Singri, P. Sadayappan, and S. Krishnamoorthy. Resilience 2010

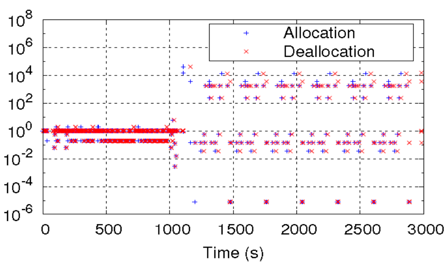
Research Elements

- System software
- Work/Data distribution
- Application validation

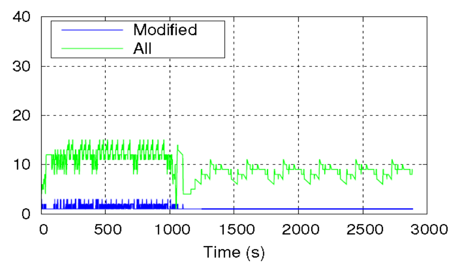
Application Characterization-Based Tolerance

Application-Specific Fault Tolerance via Data Access Characterization. N. Ali, S. Krishnamoorthy, N. Govind, K. Kowalski, P. Sadayappan. EuroPar 2011

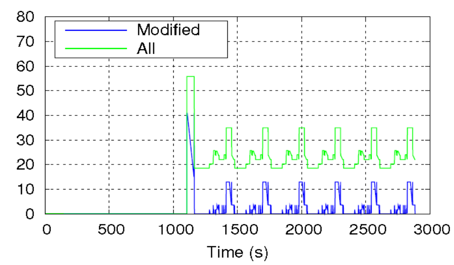




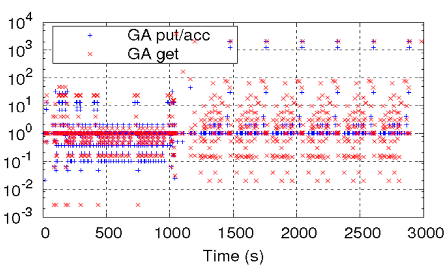
(a) GA (de)allocation (MB)



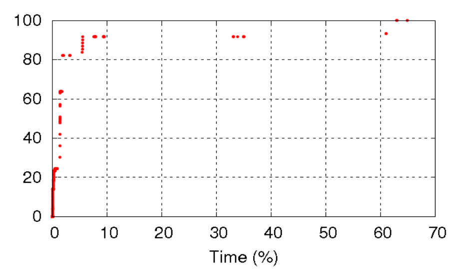
(b) Number of GAs



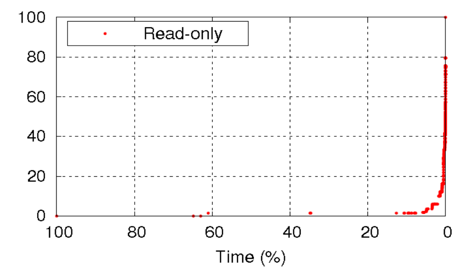
(c) Total GA size (GB)



(d) Reuse factor



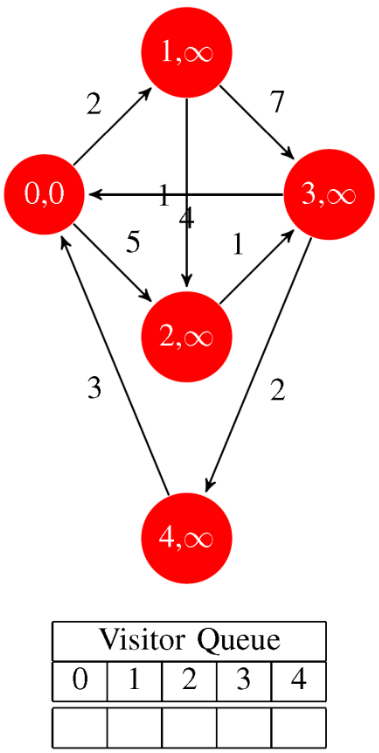
(e) Data liveness (%)



(f) Read-only window (%)

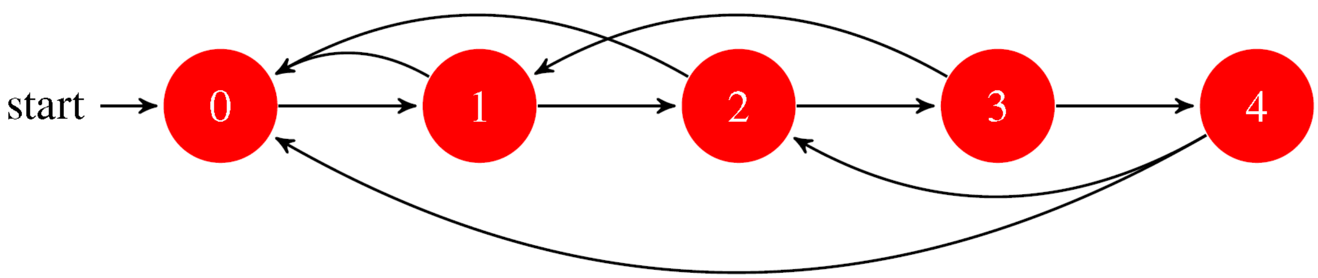
Skeleton Applications

[Multithreaded Asynchronous Graph Traversal for In-Memory and Semi-External Memory](). R. Pearce, M. Gokhale, N. Amato, SC 2010

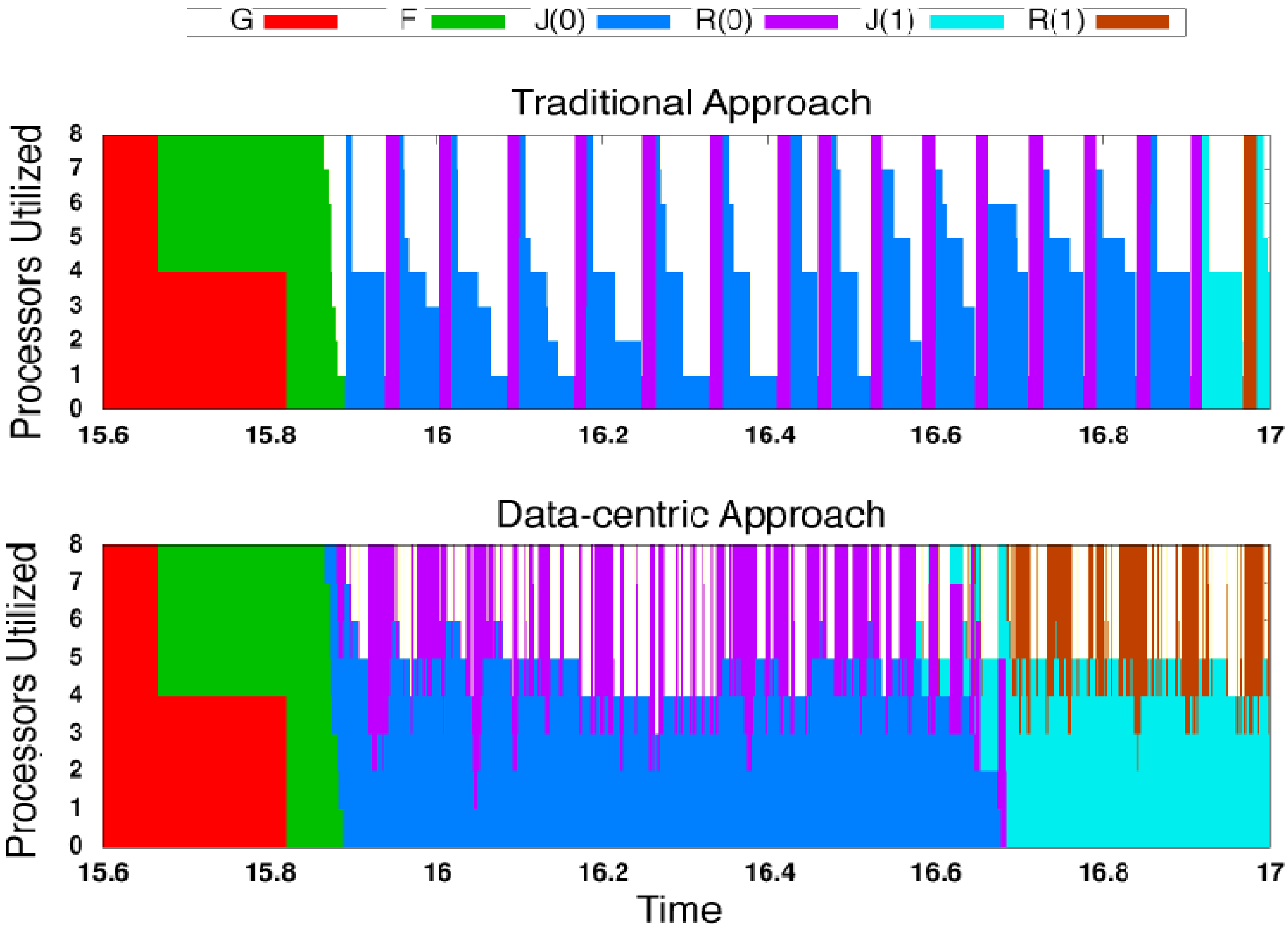


Algorithm 1 Hartree-Fock pseudocode

```
Specify the system (molecule) and the basis functions
Calculate the overlap, core Hamiltonian, and two-electron integrals (S_μν, H^core_μν, (μν|λσ))
Diagonalize the overlap matrix S and obtain the transformation matrix X
Start with a guess density matrix D
while ¬converged do
    Calculate the two-electron contribution G using (μν|λσ) and D
    Obtain the Fock matrix F = H^core + G
    Transform the Fock matrix F′ = X†FX
    Diagonalize F′, get C′, ε, where C′ is the transformed coefficient matrix and ε denotes the eigenvalues.
    Transform C = XC′
    Calculate new density matrix D
end while
```



Validation using SST/Macro



End-to-End Validation

- System software
  - Initial development on BG systems
- Validation at scale
  - SST/Macro simulator

Conclusion

- This is an end-to-end effort
- Builds on past research on various components
- Rethinking application architecture and software layers

Thank You