SAND2014-1307C

Exceptional service in the national interest

Sandia National Laboratories

Macroscale Architecture Simulation for Data-dependent Applications: Adaptive Mesh Refinement

Joseph P. Kenny

U.S. DEPARTMENT OF ENERGY

Office of Science U.S. Department of Energy

NNSA National Nuclear Security Administration

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 NO. 2011-XXXXP

Sandia National Laboratories

SST

ExaCT

BERKELEY LAB

Macroscale Architecture Simulation for Data-dependent Applications: Adaptive Mesh Refinement

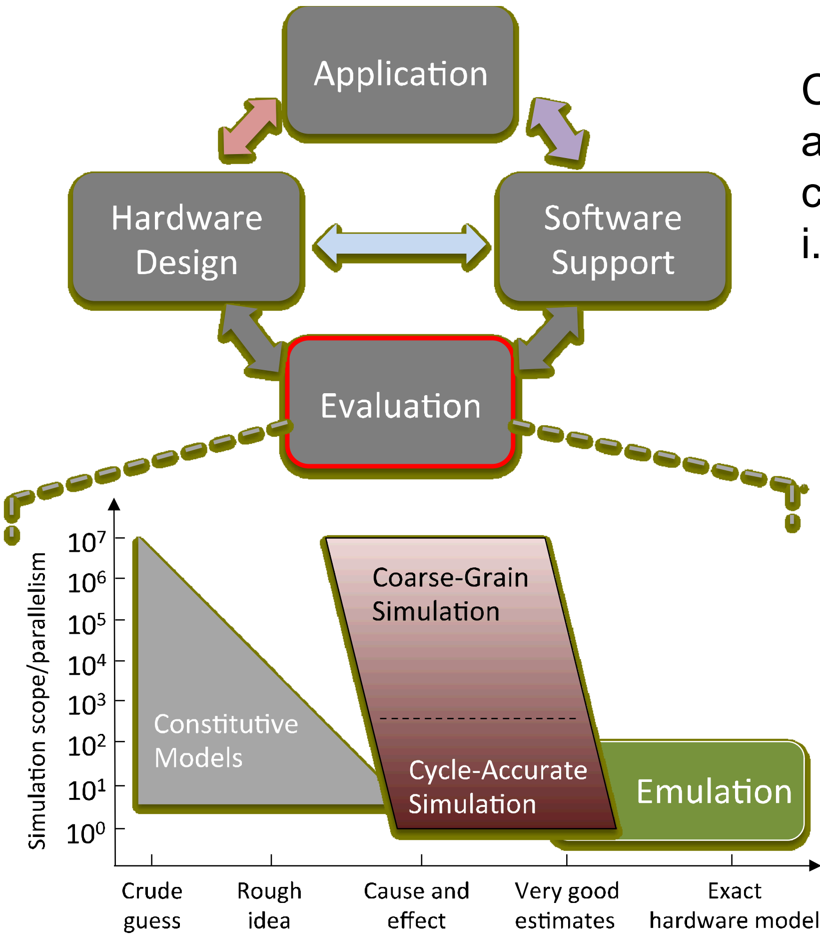
- System-level simulation at extreme scales requires coarse-grained models
  - Cycle accurate models are prohibitively slow
  - Coarse-grained models must be fast/cheap but accurately reproduce characteristics of interest (e.g. congestion)
- SST/macro: a coarse-grained system-level structural simulator
- Boxapp: AMR proxy app from ExaCT Combustion CoDesign Center
  - How do you simulate a data-dependent application without producing data?

Who did what?

SST/macro network models: Jeremiah Wilke, Gilbert Hendry, Curtis Janssen, Helgi Adalsteinsson, Ali Pinar (SNL/ASC)

Boxapp: Cy Chan, Vincent Beckner, John Bell and John Shalf (LBL/ExaCT)

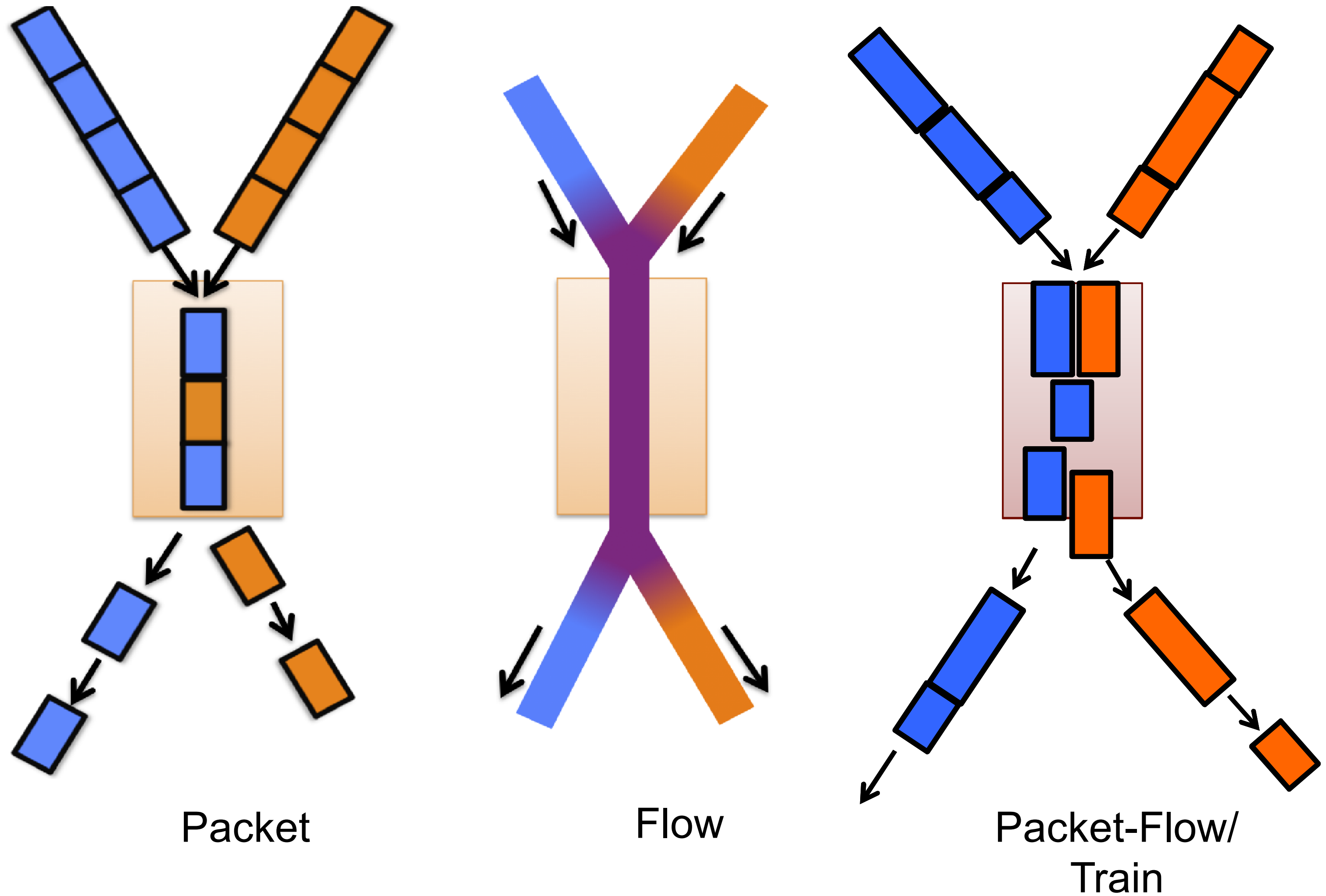
Coarse-grained Modeling: Why is "System-Level" Critical?



Co-design of algorithms, runtime support, and hardware requires intermediary to close loop:
i.e. simulation!

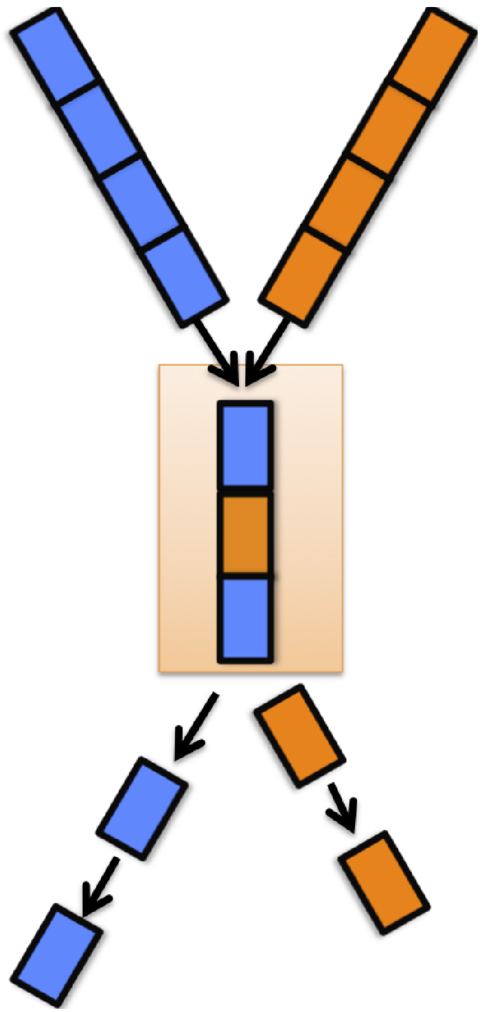
Coarse-grained simulation sits in sweet spot: cheap enough to simulate large systems, but accurate enough to capture real causes/effects

Congestions Models



Sources of error: Packet model

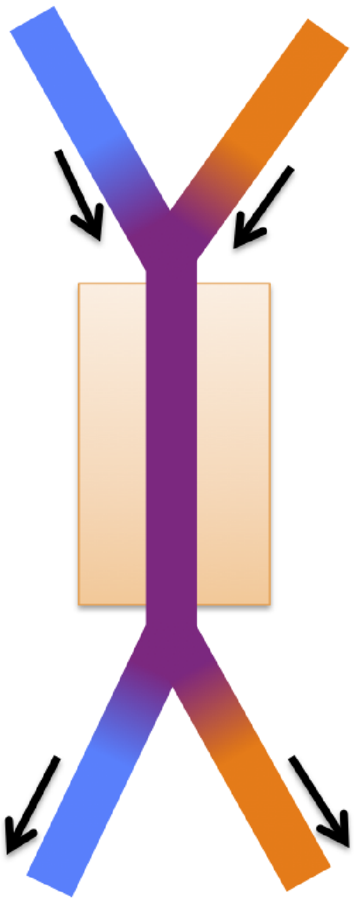
- Serialization latency: Store-and-Forward
  - We cannot model flits! Every packet has to use store-and-forward routing. There is no "cut-through" routing without flits.
  - 100B (actual packets) latency error per switch is 20ns at 5 GB/s.
  - 4KB (coarse packets) latency error per switch is 800 ns at 5 GB/s.
- Fair arbitration
  - We cannot model flits! Packets cannot share (multiplex) on a link. Artificially wasted bandwidth.
  - 4KB (coarse packets) ''arbitration'' error is same as latency error
- Routing
  - For coarse packets, routing decision is made for large chunk (4KB) rather than ``real'' size.
  - Minimal routing has no errors
  - Valiant routing has basically no errors
  - Adaptive routing becomes ``dumber''



Sources of error: Flow model

Solved as a fluid dynamics problem. Flows are large point-to-point messages. Links are ``pipes'' that partition bandwidth amongst competing flows.

- Ripple effect
  - For zero congestion, even 1 GB messages modeled in a few events
  - Congestion events cascade throughout entire system
  - For heavy congestion, quickly becomes MORE expensive than packet models
- Routing
  - Difficult to quantify congestion
  - Difficult to detect congestion
  - Adaptive routing algorithms difficult to replicate

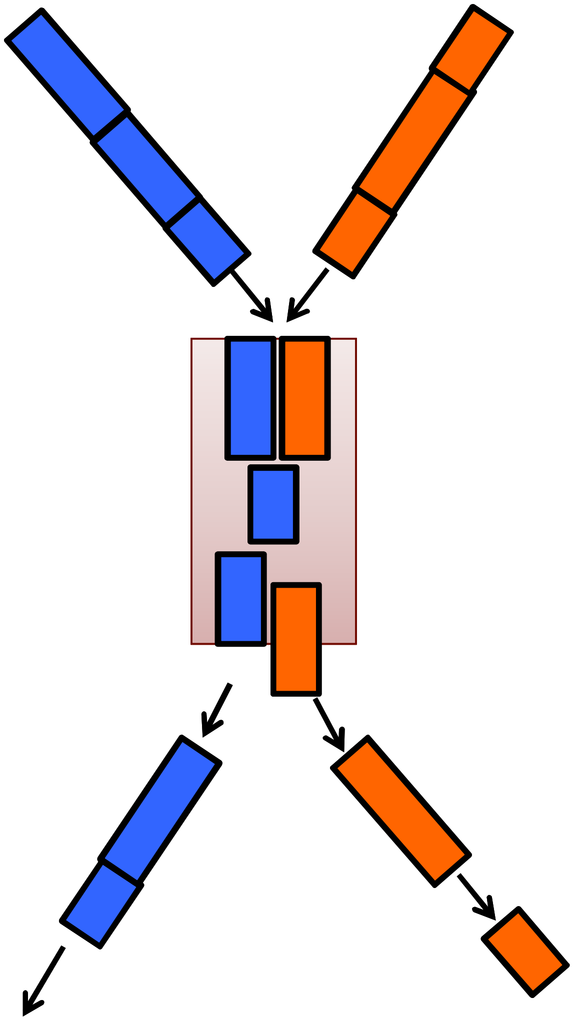


Sandia National Laboratories SST ExaCT BERKELEY LAB

Sources of error: Packet-Flow model

Mixture of both models. Congestion arbitrated in discrete chunks, but chunks have notion of bandwidth.

- Serialization latency: Cut-Through
  - Packet allocates available bandwidth and immediately forwards
- Fair arbitration
  - Some link sharing error, but packets no longer exclusively use entire link. Bandwidth can be shared.
- Routing
  - For coarse packets, routing decision is still made for large chunk (4KB) rather than ``real'' size.



HPC Simulation and SST/macro

Simulation Type	On-line	Native C,C++,Fortran,DSL
	Off-line	Trace replay
Network Model	Structural	Cycle-accurate, packet, flow
	Analytic	Latency/Bandwidth, LogGP
Compute Model	Direction Execution	
	Performance Counter Convolution	
	Parameterized (Coarse-grained) Model	

SST/macro
Structural Simulation Toolkit for Macroscale

Primary: On-line, structural simulator with coarse-grained compute models

Secondary: Off-line, trace driven (DUMPI trace collector)

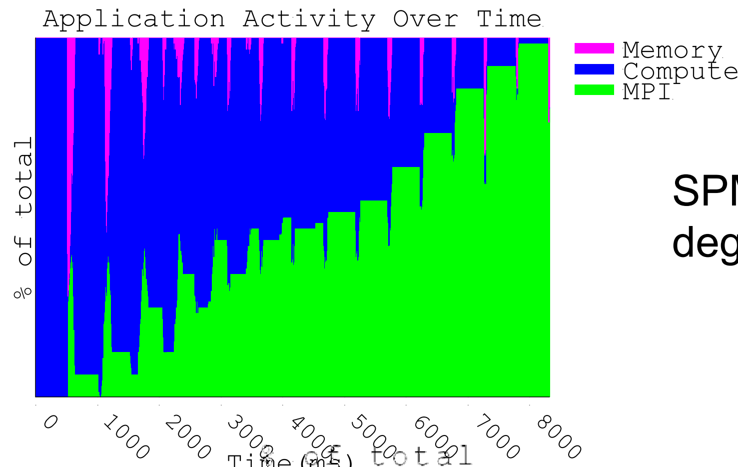
Simulator	Ref no.	On/Off-line	Computation	Congestion	Language
LogGOPS	[2]	Trace	Model	Yes	DSL (GOAL)
BSIM	[6]	On-line	Coarse Model	Yes	Native
Mambo/Seshat	[8]	On-line	Cycle-Acc	No	Native
PSINS	[9]	Trace	Time-dependent	Yes	n/a
MPI-SIM	[10]	On-line	Direct	No	Native
Dimemas	[11]	Trace	PerfCtr	No	n/a
WARPP	[4]	Trace	PerfCtr	Yes	Native

Table 1. Survey of analytic HPC simulators. Computation may be time-dependent trace, performance counter convolution (PerfCtr), direct execution, or coarse-grained.

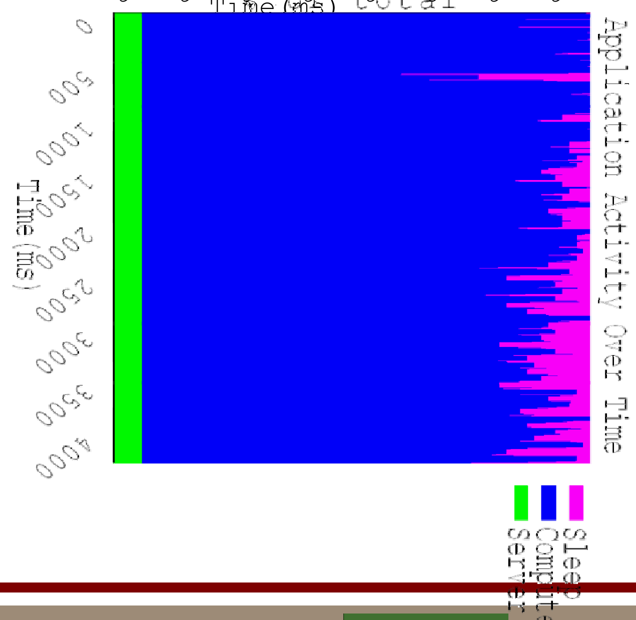
Simulator	Ref no.	On/Off-line	Computation	Network	Language
BigSim	[12,13]	Both	PerfCtr/Model	Packet	Native
SIMGRID	[14,15]	Both	PerfCtr	Flow	Native
MARS	[16]	Trace	Time	Packet	n/a
MPI-NeTSim	[17]	On-line	Direct	Packet	Native
PACE	[18]	Both	Abstract	Abstract	DSL (CHIP^{3}S)
SST/macro	[19]	Both	PerfCtr/Model	Packet/Flow	Native

Table 2. Survey of structural HPC simulators. Computation may be time-dependent trace, performance counter convolution (PerfCtr), or coarse-grained model.

SST/macro Analysis Tools: Fixed-Time Quanta (FTQ)



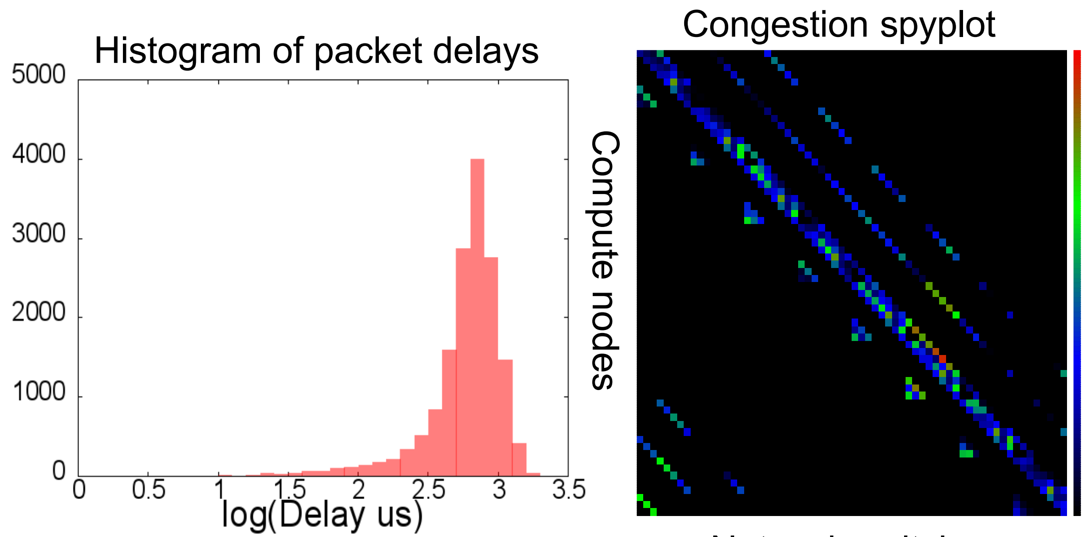
SPMD MPI code in presence of node degradations for matrix-matrix multiplication



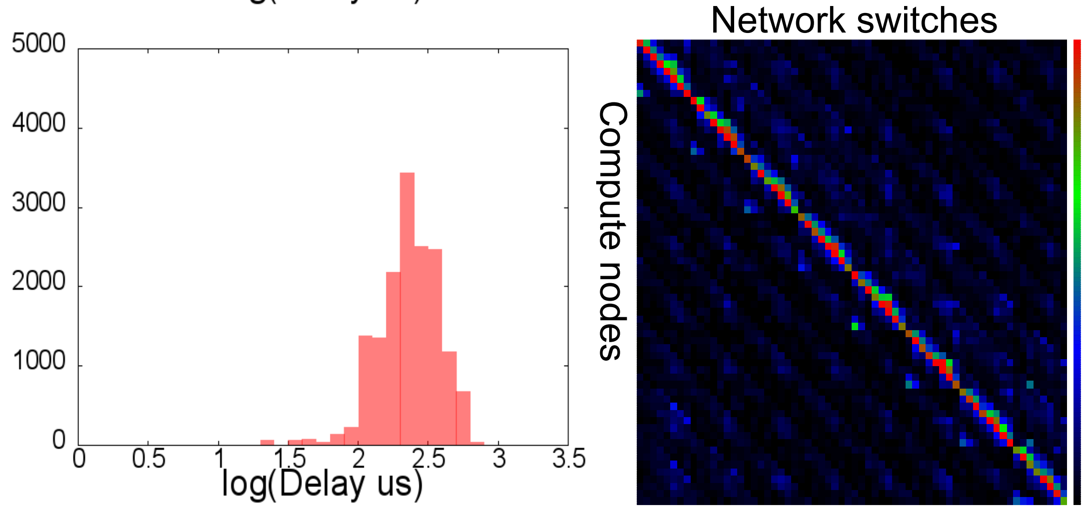
Asynchronous task model even in presence of node degradations

SST/macro Analysis Tools: Congestion Analysis

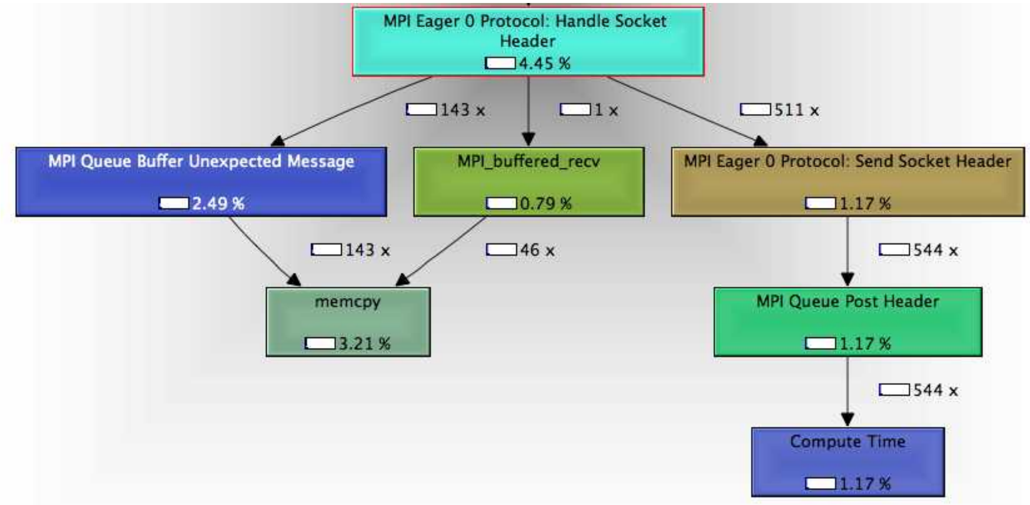
Minimal routing
Adversarial traffic
Network latency is 5.2x injection latency

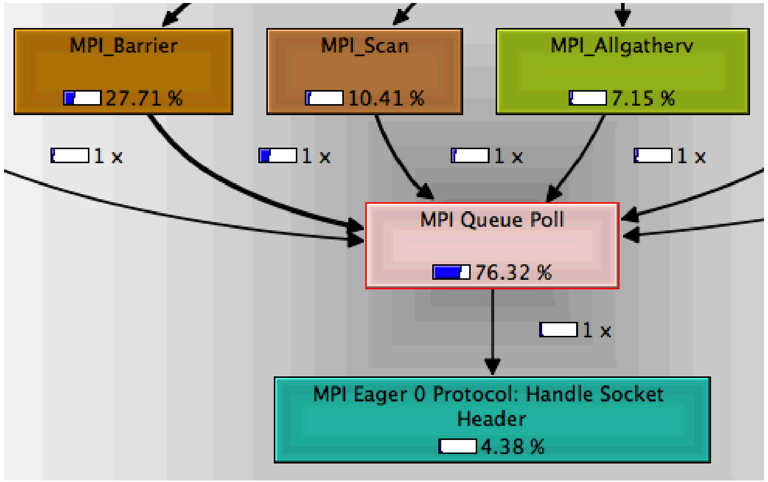


UGAL routing
Adversarial traffic
Network latency is 1.8x injection latency

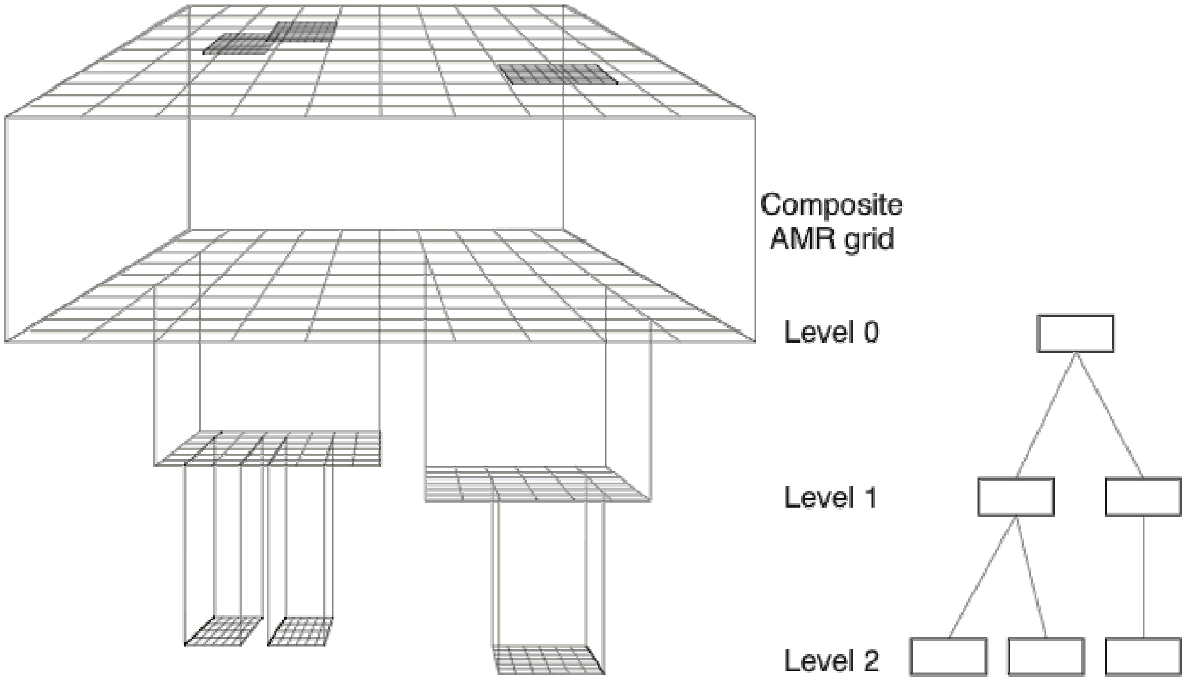


SST/macro Analysis Tools: Callgraph/Profiling





Adaptive Mesh Refinement (AMR)



- Multiple levels
- Set of boxes at each level
- Fine boxes enhance resolution at areas of interest
- Boxes exchange data within and across levels
- Irregular communication and unbalanced computation

Boxapp

- Data dependent simulations present challenges
  - Iterative/converging methods can usually just be hardcoded
  - AMR is a challenging case, refinement, load balancing, layout is all data-dependent.

AMR Analysis and Simulation Goals

- On-node performance modeling with ExaSAT
  - Compiler-driven static analysis and modeling
- Need network simulation capability
  - Leverage SST/macro software simulator
- Asynchronous execution model
- Simulate performance on many potential exascale machine configurations
- Analyze the effects of:
  - Data distribution
  - Network topology

Analysis Toolchain and Methodology

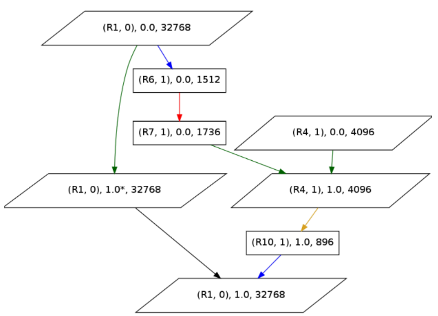
Problem Specification: CASTRO

BoxLib AMR Library

Box List

Level 0
0: ((0, 0, 0) (15,31,15)) 16 32 16 :: 3
0: ((16, 0, 0) (39,31,15)) 24 32 16 :: 1
Level 1
1: ((30, 0, 0) (47,31,31)) 18 32 32 :: 2
1: ((48,14,10) (67,29,29)) 20 16 20 :: 3
...
Level 2
2: ((72, 0,34) (83,19,59)) 12 20 26 :: 1
2: ((72, 0,60) (83,15,75)) 12 16 16 :: 2
...

AMR Dependency Analysis Tool



Dependency Graph

XML

<boxes>
<box id="R1" loc="0" />
<box id="R4" loc="1" />
</boxes>

<events>
<comp id="E10" dep="E5,E11" time="0.0676" />
<comm id="E12" dep="E2" from="R1" to="R4" size="1512" />
...
</events>

SST/macro Network Simulation

Performance Estimates

XML Specification

```
<boxes>
<box id="R1" loc="0" />
<box id="R4" loc="1" />
</boxes>

<events>
<comp id="E10" dep="E5,E11” at="R4" time="0.0676" />
<comm id="E12" dep="E2” from="R1" to="R4" size="1512" />
...
</events>
```

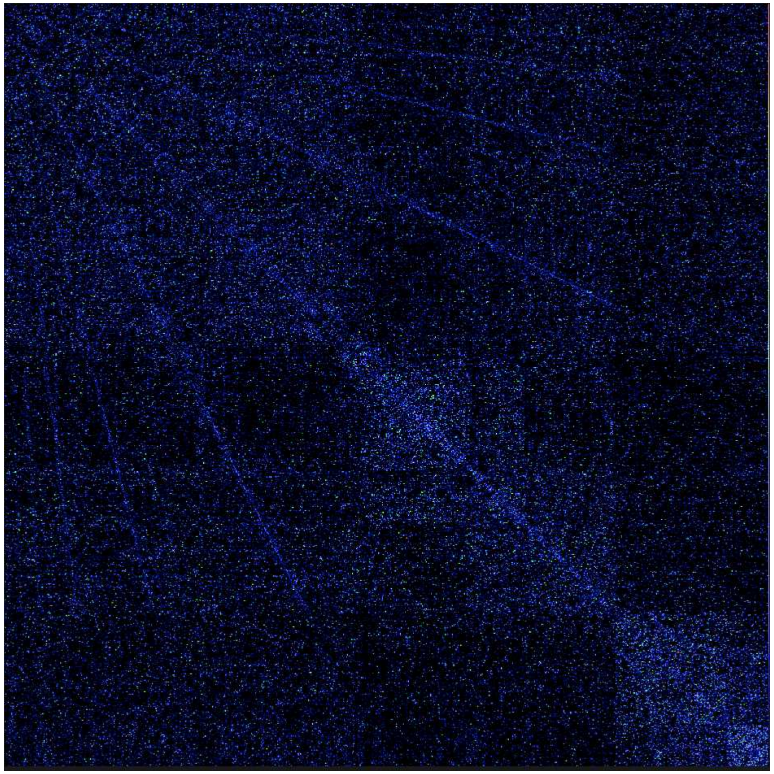
Boxes specify their locations

Events specify their dependencies

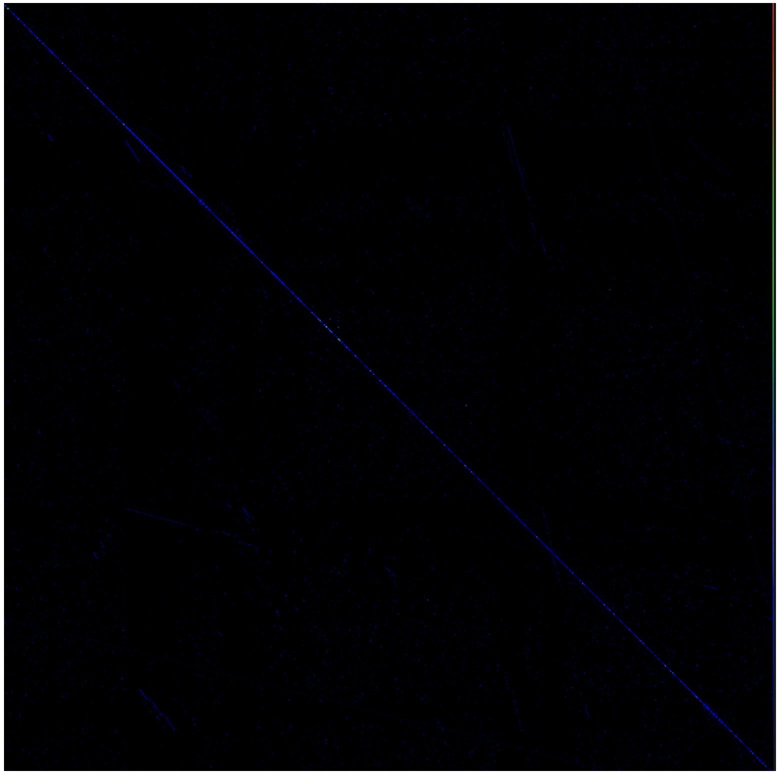
- List of boxes and events to drive simulator
  - Boxes can be re-assigned to different locations
  - Computation events have execution time estimates
  - Communication events have source, destination, and size

Boxapp Communication

Exanode 1, 1200 nodes, 3D Torus



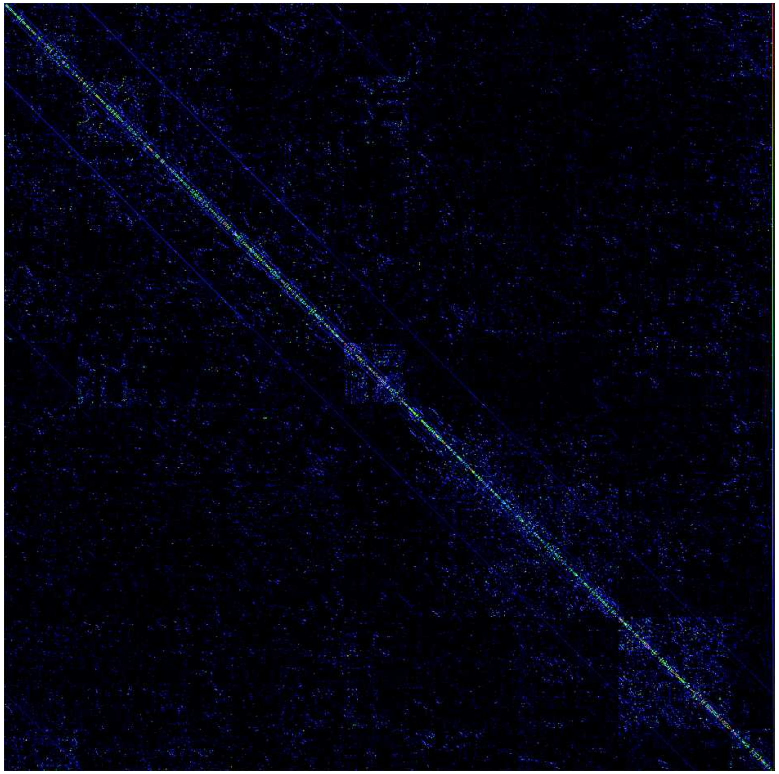
Knapsack



Space Filling Curve

Boxapp Communication

Exanode 1, 1200 nodes, 3D Torus

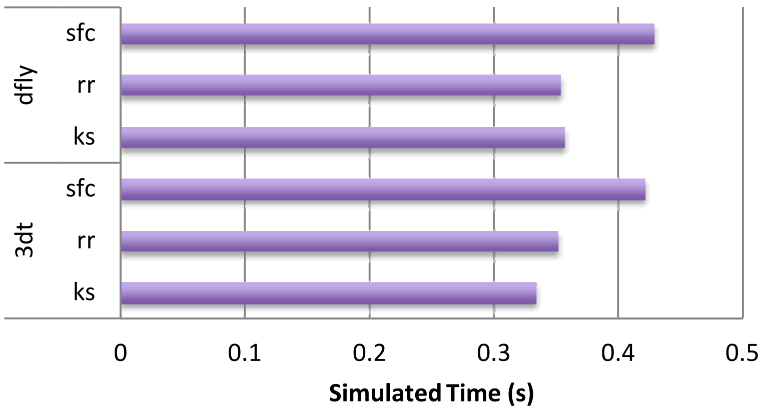


Round Robin

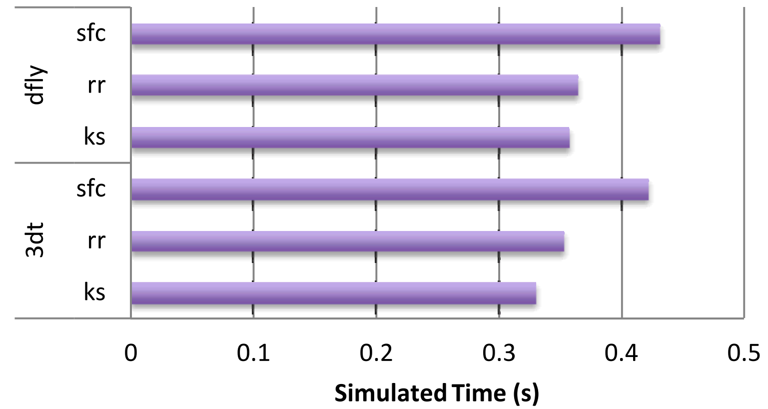
Boxapp Simulated Times

1200 nodes

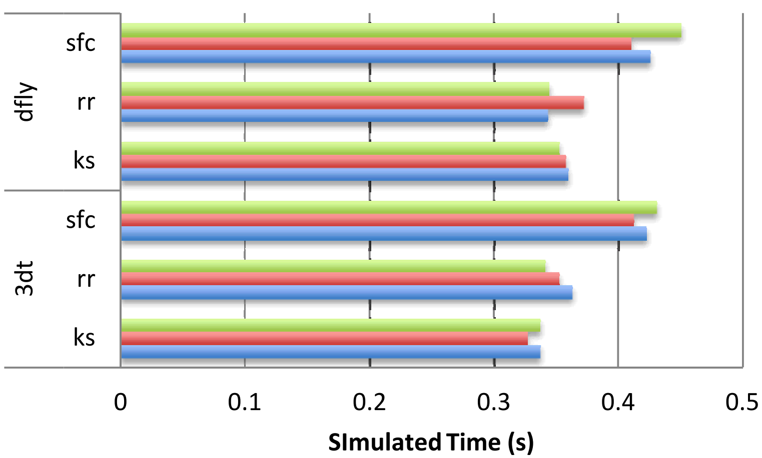
Exanode 1



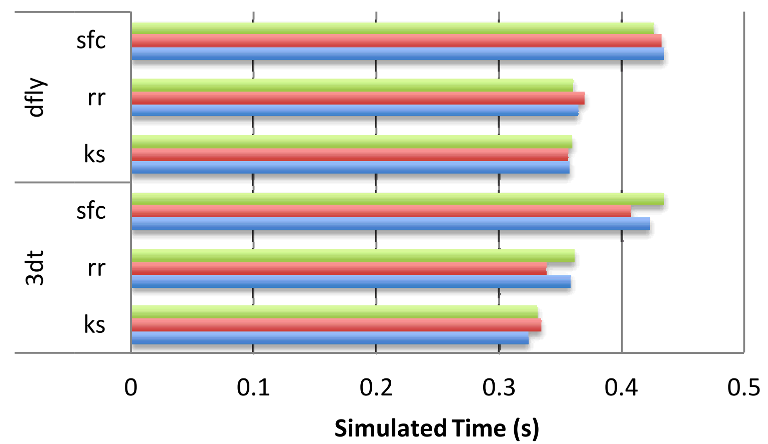
Exanode 2



Exanode 1



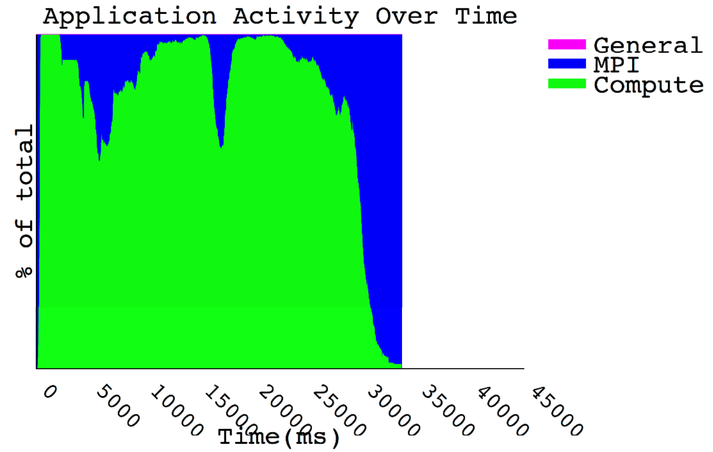
Exanode 2



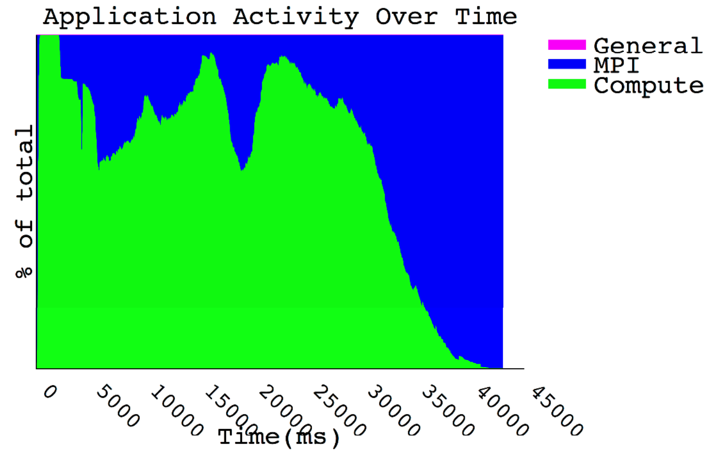
Boxapp Idle Time

Exanode 1, 1200 nodes, 3D Torus

Knapsack

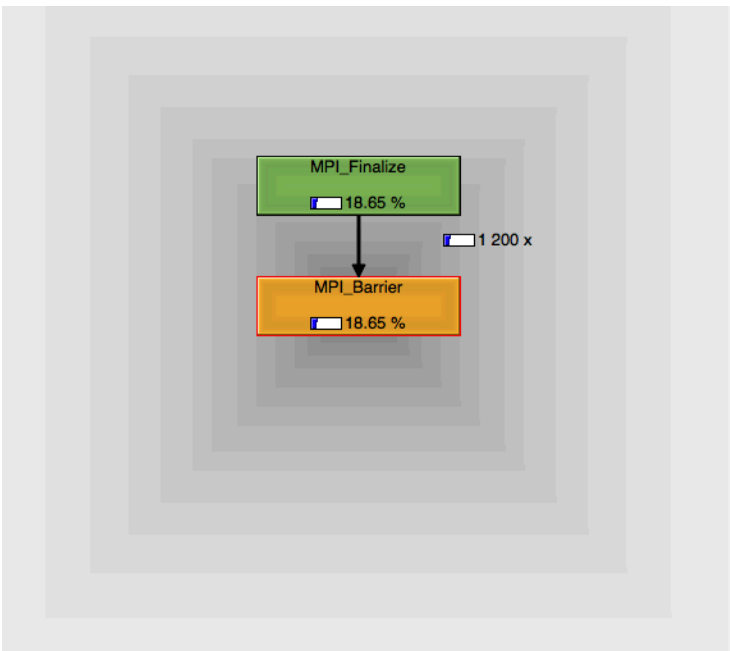


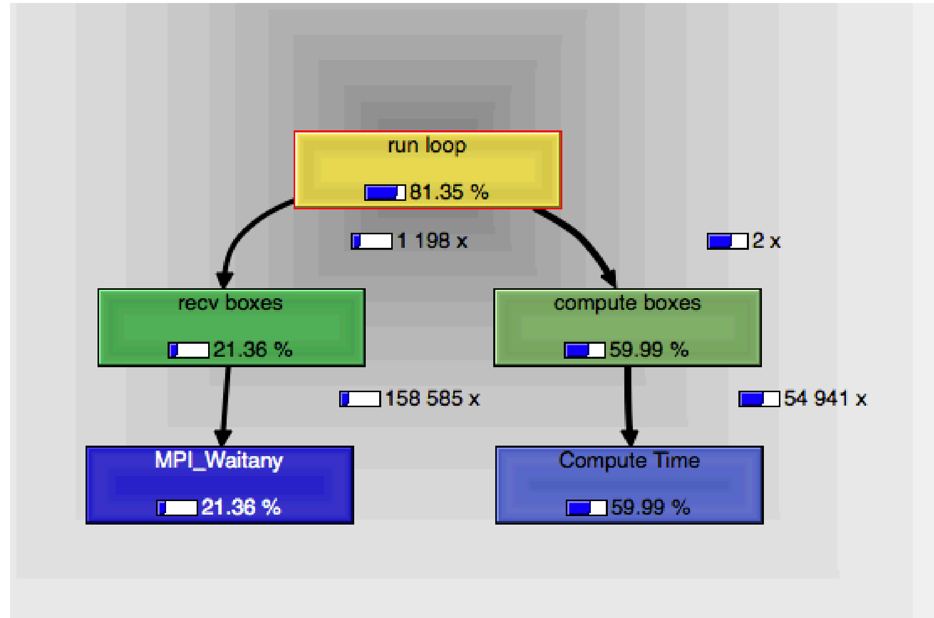
Space Filling Curve



Boxapp Idle Time

Exanode 1, 1200 nodes, 3D Torus





Future Work

- Expand simulation scale
  - 10,000 nodes boxapp runs now quite comfortable
  - 100,000 endpoints (MPI tasks) should be sufficient for “exascale”
    - Serial DES, or is parallel required?
- Explore more aspects of AMR
  - Better layout algorithms?
  - Fine grained parallelism?