Exceptional service in the national interest

Sandia National Laboratories

Creating the Next Generation of Stable, Interoperable and Performant Simulator – A Call for Open Standards

Jeremiah Wilke*, Joseph Kenny*, Robert Clay*, Simon Hammond[+], Arun Rodrigues[+], Scott Hemmert[+], James Ang[+], Sudhakar Yalamanchili[ł]

*Sandia CA, [+]Sandia ABQ, [ł]Georgia Tech

U.S. DEPARTMENT OF ENERGY

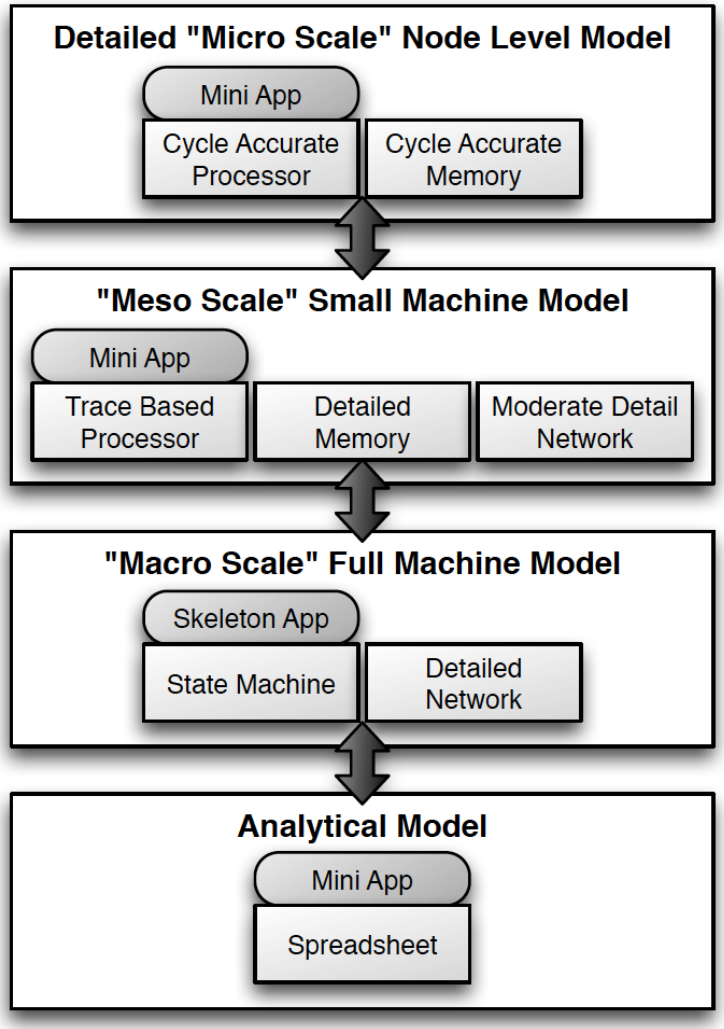
NNSA National Nuclear Security Administration

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

We need models of varying fidelity
How many tools do we need for all of them?

Sandia National Laboratories

- High fidelity
  - What you care about only at exists at cycle-accurate detail
    - Cache reuse policies in memory
    - Flit-level flow control
  - Validation of lower fidelity models
- Medium fidelity
  - Coarse-grained modeling of architecture at system scale
    - Adaptive routing without flit detail
    - Scaling of collectives with network congestion
  - Validation of constitutive models at scale
- Constitutive models
  - Potentially good accuracy with right fitting
  - Rapid parameter space exploration



We need models of varying fidelity
How many tools do we need for all of them?

Sandia National Laboratories

SST
Structural Simulation Toolkit

We need models of varying fidelity
How many tools do we need for all of them?

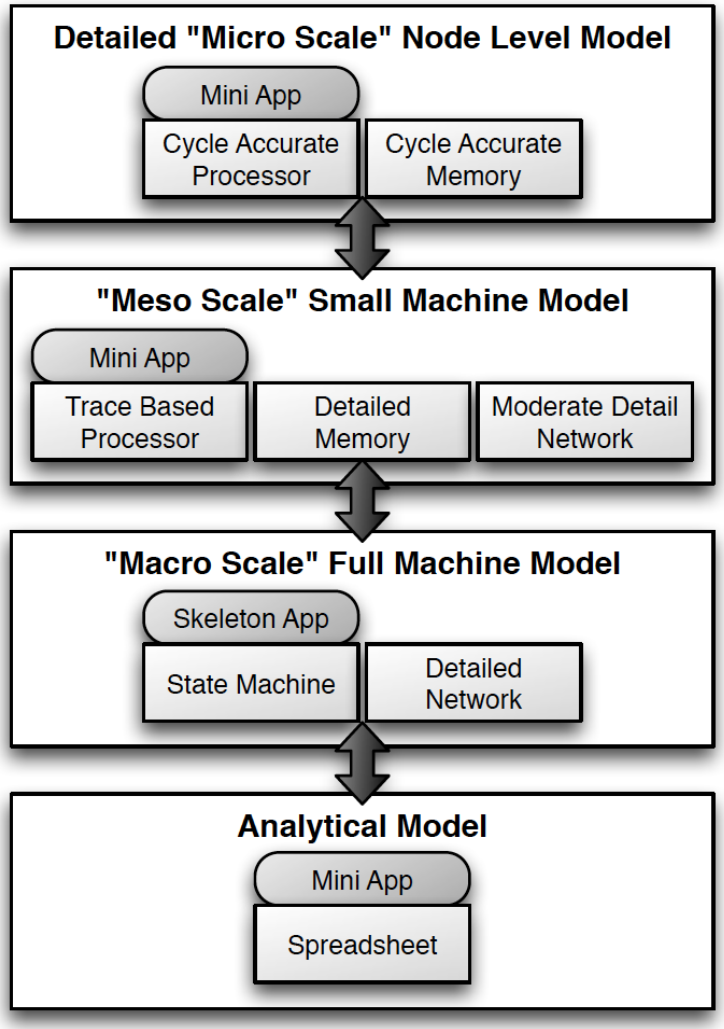
Sandia National Laboratories

SST
Sonic Screwdriver Toolchain

How do we defeat the Daleks?

Sandia National Laboratories

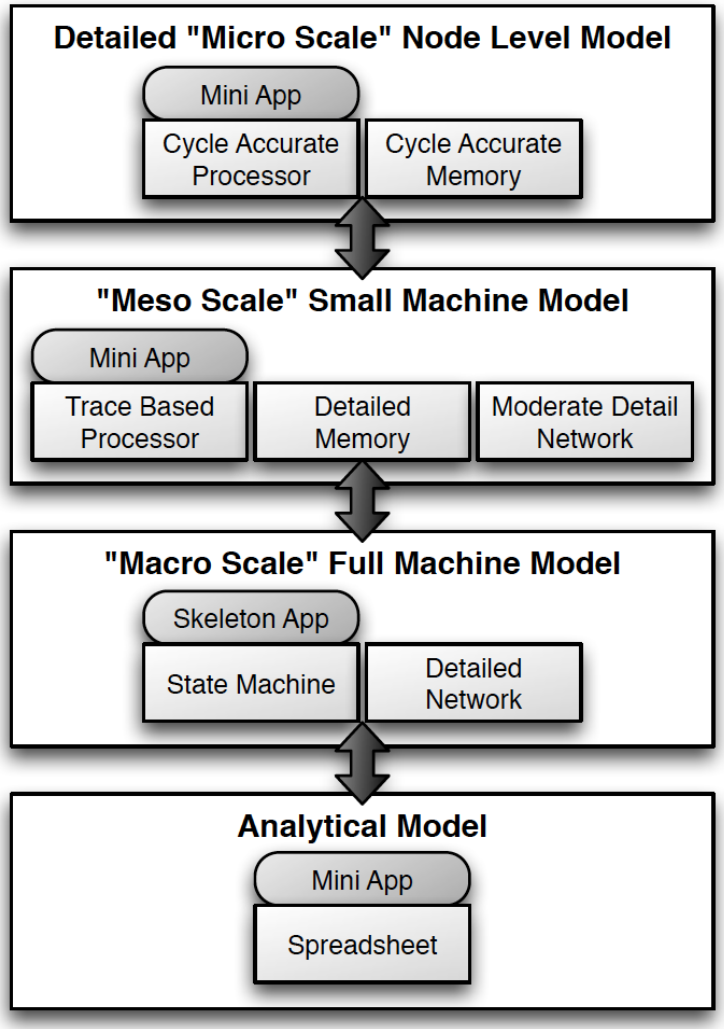
- Common Core
  - Scale up performance = scale up performance simulation = parallel simulation (PDES)
- Composability
  - Define standards for composing models that speak same language and share the same notion of time
- Community
  - Concerted effort to define standards and reuse code



How do we defeat the Daleks?

Sandia National Laboratories

- Common Core
  - Scale up performance = scale up performance simulation = parallel simulation (PDES)
- Composability
  - Define standards for composing models that speak same language and share the same notion of time
- Community
  - Concerted effort to define standards and reuse code

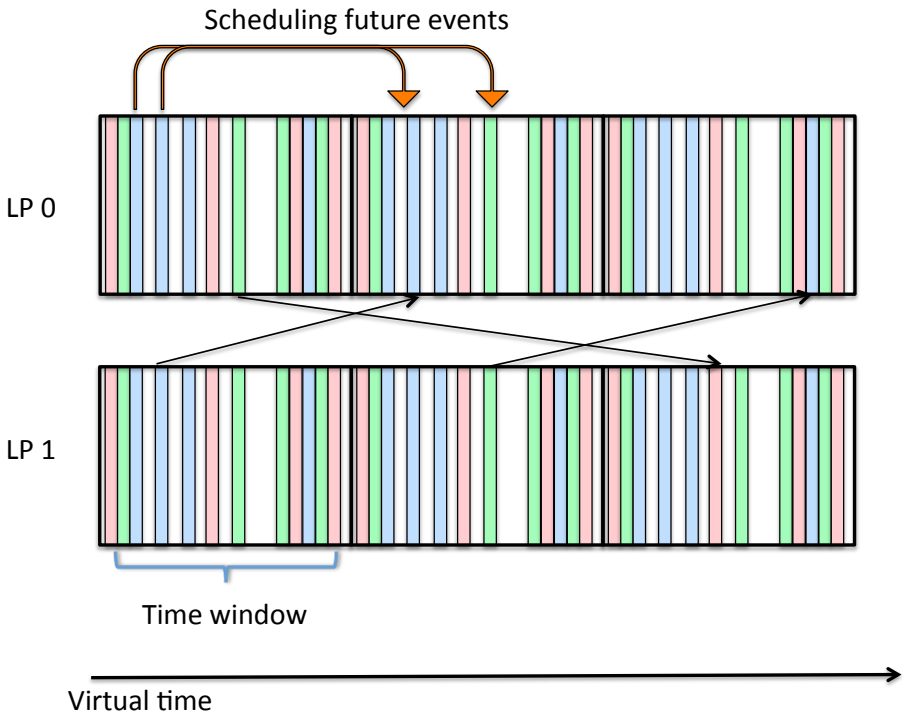


What is the major source of suffering in parallel discrete event simulation (PDES)?

Sandia National Laboratories

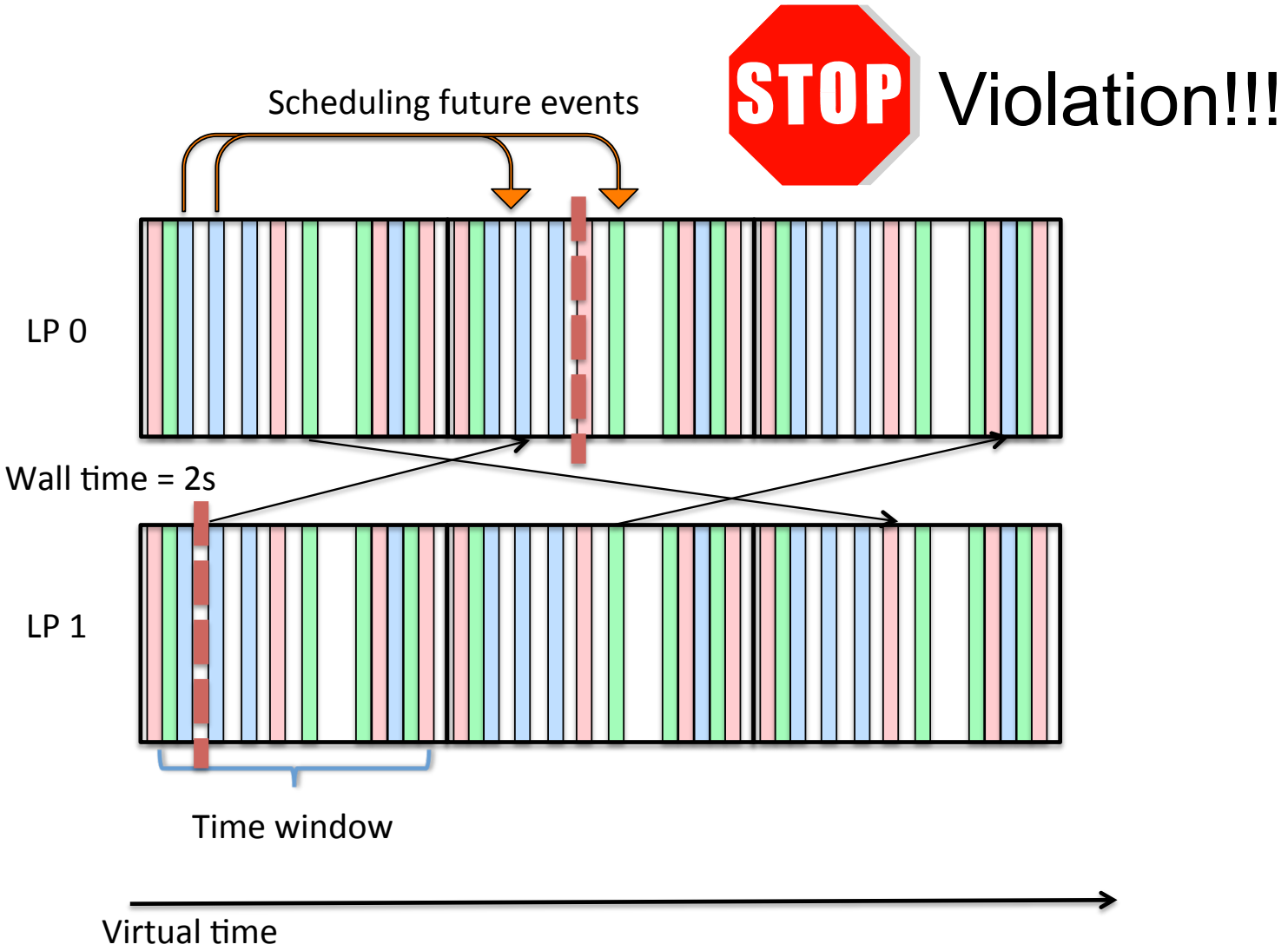
- Scale up machines = scale up simulations of machines = PDES
- Need event management and scheduling
  - Avoid time-order violations!

Dependencies between events



What is the major source of suffering in parallel discrete event simulation?

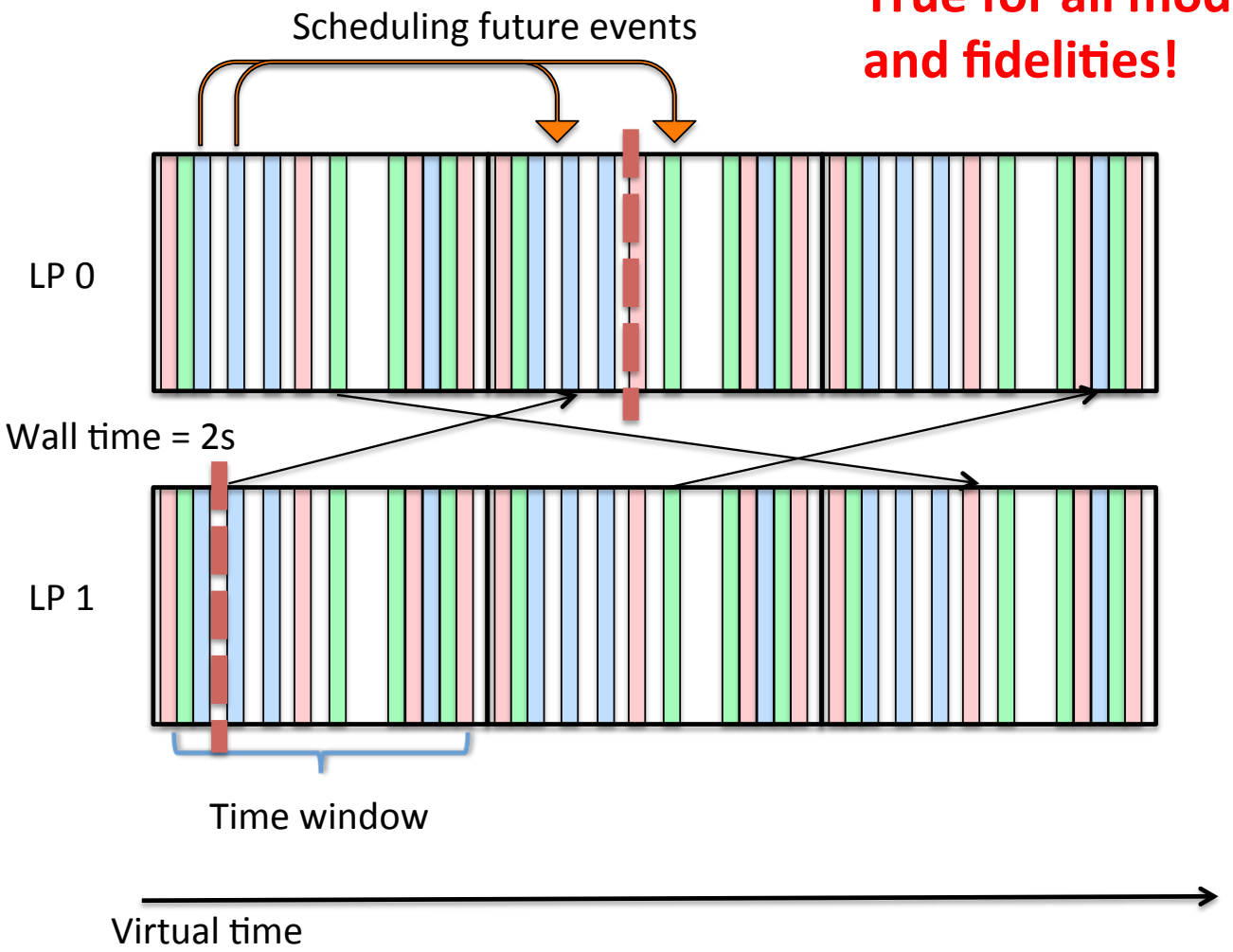
Sandia National Laboratories



What is the major source of suffering in parallel discrete event simulation?

Sandia National Laboratories

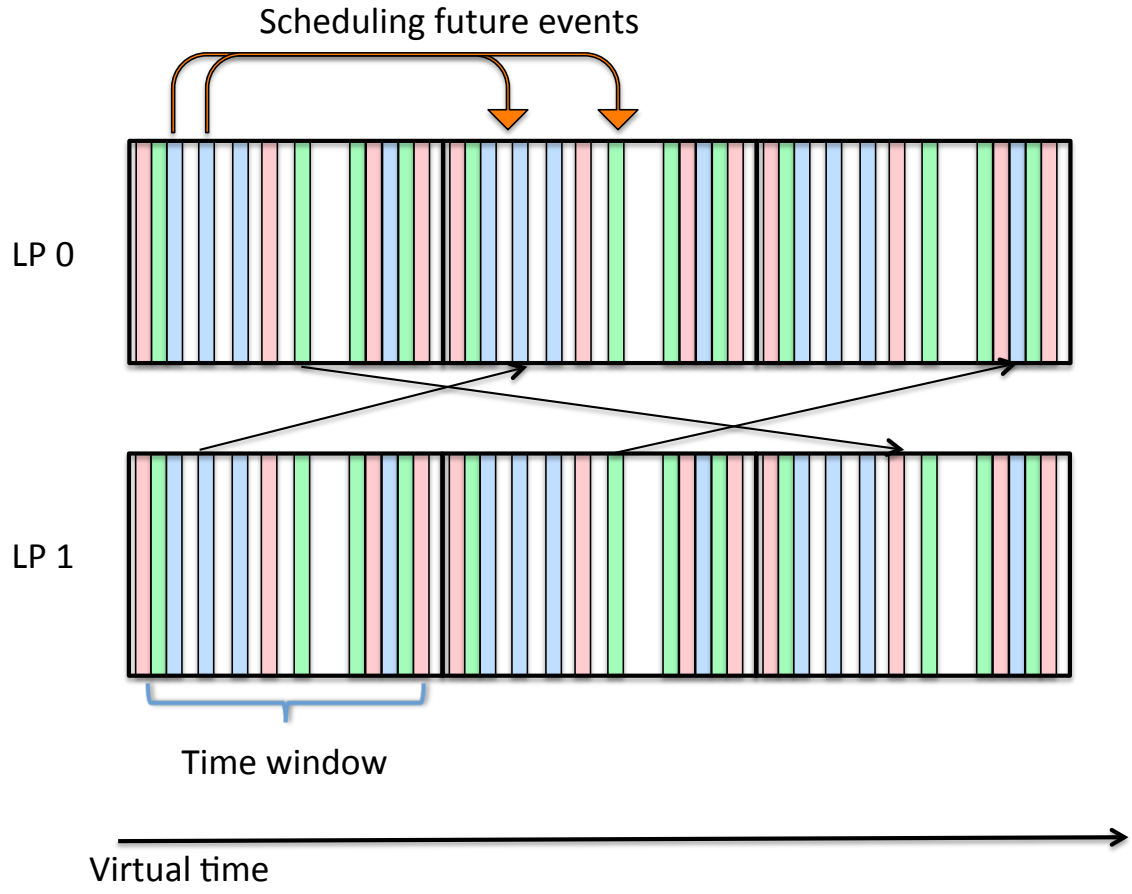
True for all models and fidelities!



What is the major source of suffering in parallel discrete event simulation?

Sandia National Laboratories

- Parallelism possible mainly from lookahead (safe time window) based on virtual latency between components



Solutions to the problem exist, but are non-trivial: Why rewrite over and over?

Sandia National Laboratories

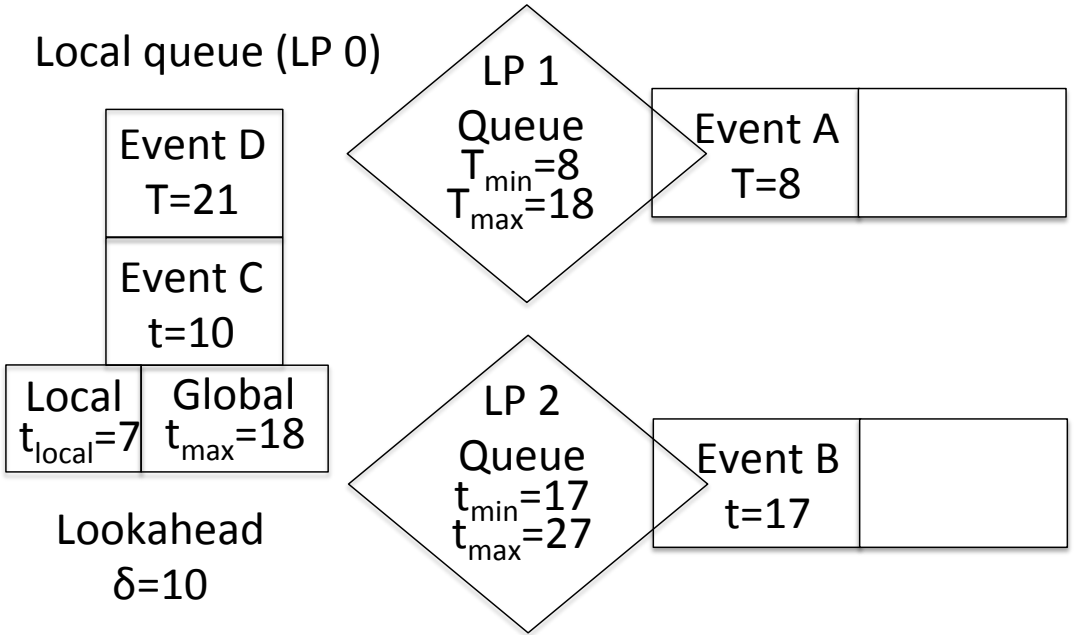
- "Naïve" conservative time-stepping algorithm
  - Global, collective communication
  - Communication optimizations to lower prefactor, but has scalability limits

while $t < t_{termination}$ **do**
  Run all events until $t + \delta$
  Log event sends to $S = \{N^{LP0}_{events}, N^{LP0}_{bytes}, N^{LP1}_{events}, \ldots\}$
  MPI_ReduceScatter with SUM array S
  MPI_ReduceScatter returns N^{total}_{events}, N^{total}_{bytes}
  $N^{left}_{bytes} = N^{total}_{bytes}$
  for $msgNum < N^{total}_{events}$ **do**
    MPI_Recv(void*, MPI_ANY, N^{left}_{bytes})
    $N^{left}_{bytes} = N^{left}_{bytes} - msgSize$
  end for
  Determine t^{local}_{min}
  MPI_Allreduce(t^{global}_{min}, t^{local}_{min})
  $t = t^{global}_{min}$
end while

Solutions to the problem exist, but are non-trivial: Why rewrite over and over?

Sandia National Laboratories

- Conservative algorithm with event queues: Optimization to limit communication for LP's that are "connected"
  - Local, point-to-point communication
  - More difficult to implement



What makes us write ad-hoc code instead of leverage existing code?

Sandia National Laboratories

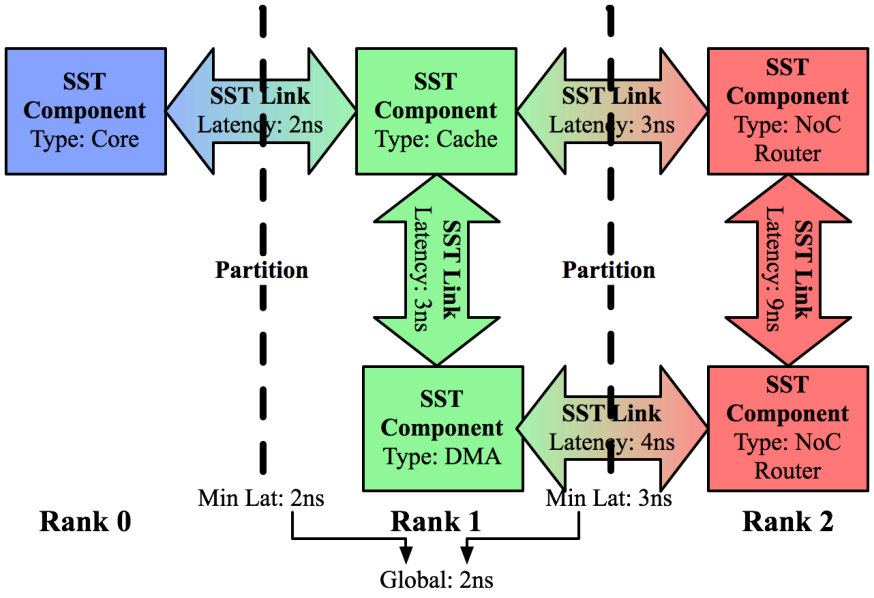
- Lack of standards
- Lack of documentation
- Huge monolithic code bases
- Compatibility across platforms
- A million and one dependencies
- Physical models dependent upon simulation framework

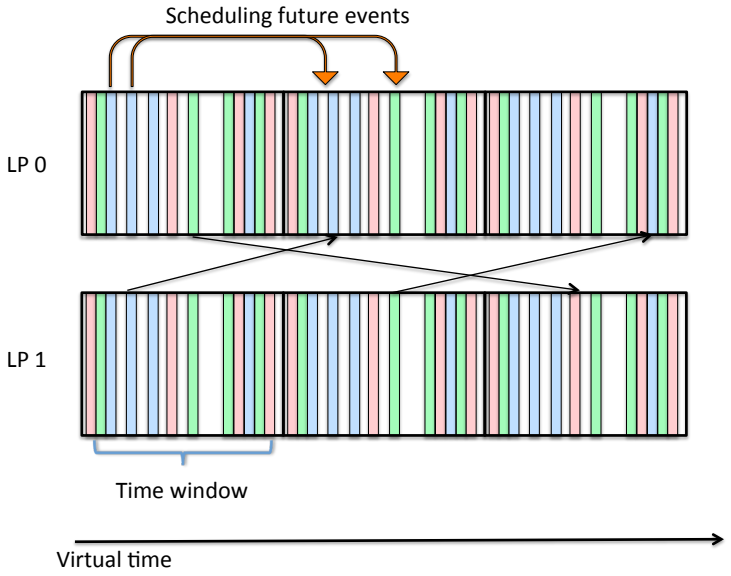
Experiments we care about are model-driven – simulator is a tool to get us what we really care about!

Design considerations for an optimal high fidelity (cycle-level) simulator

Sandia National Laboratories

- Many events per time window
- No major time gaps (generally always have events)
- Components with different link latencies and clocks
- Domain specific synchronization algorithms



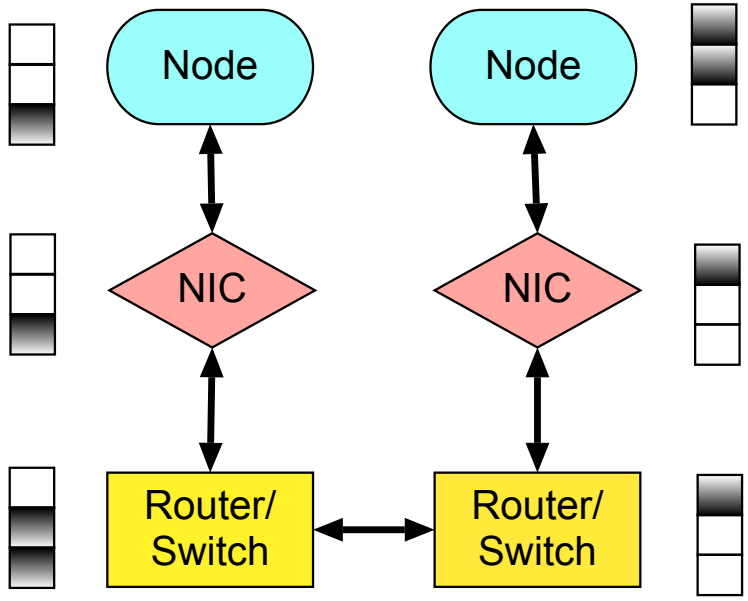


Design considerations for an optimal coarse-grained structural simulator

Sandia National Laboratories

- May have a few events per time window – or might have a lot
- Can have large gaps - time windows with no events in them
- Huge number of components, but with the same link latency

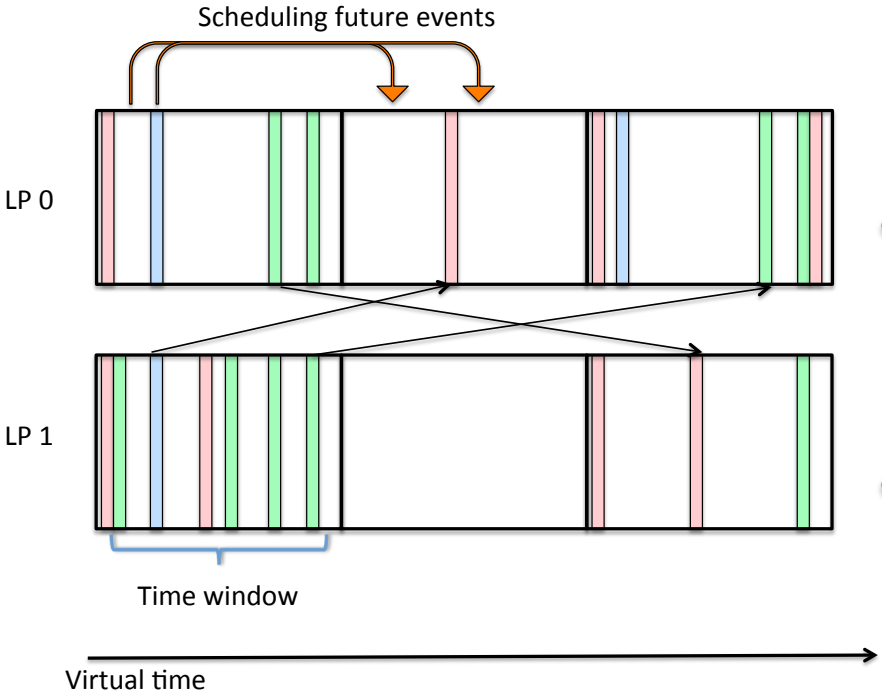
Abstract machine model with congestion via buffers and queues



Design considerations for an optimal coarse-grained structural simulator

Sandia National Laboratories

- May have a few events per time window – or might have a lot
- Can have large gaps - time windows with no events in them
- Huge number of components, but with the same link latency




```
int USER_MAIN(int argc, char **argv)
{
 MPI_Init(&argc, &argv);

 ...

 for (int iter=0; iter < niter; ++iter){
   MPI_Isend(left_block, nelems_left_block, MPI_DOUBLE,
       row_send_partner, row_tag, MPI_COMM_WORLD, &reqs[0]);
   MPI_Isend(right_block, nelems_right_block, MPI_DOUBLE,
       col_send_partner, col_tag, MPI_COMM_WORLD, &reqs[1]);
   MPI_Irecv(next_left_block, nelems_left_block, MPI_DOUBLE,
       row_recv_partner, row_tag, MPI_COMM_WORLD, &reqs[2]);
   MPI_Irecv(next_right_block, nelems_right_block, MPI_DOUBLE,
       col_recv_partner, col_tag, MPI_COMM_WORLD, &reqs[3]);

   do_dgemm('T', 'T', nrows, ncols, nlink, 1.0, left_block, nrows,
      right_block, ncols, 0, product_block, nrows);
  }

  ...

  MPI_Finalize();
}
```

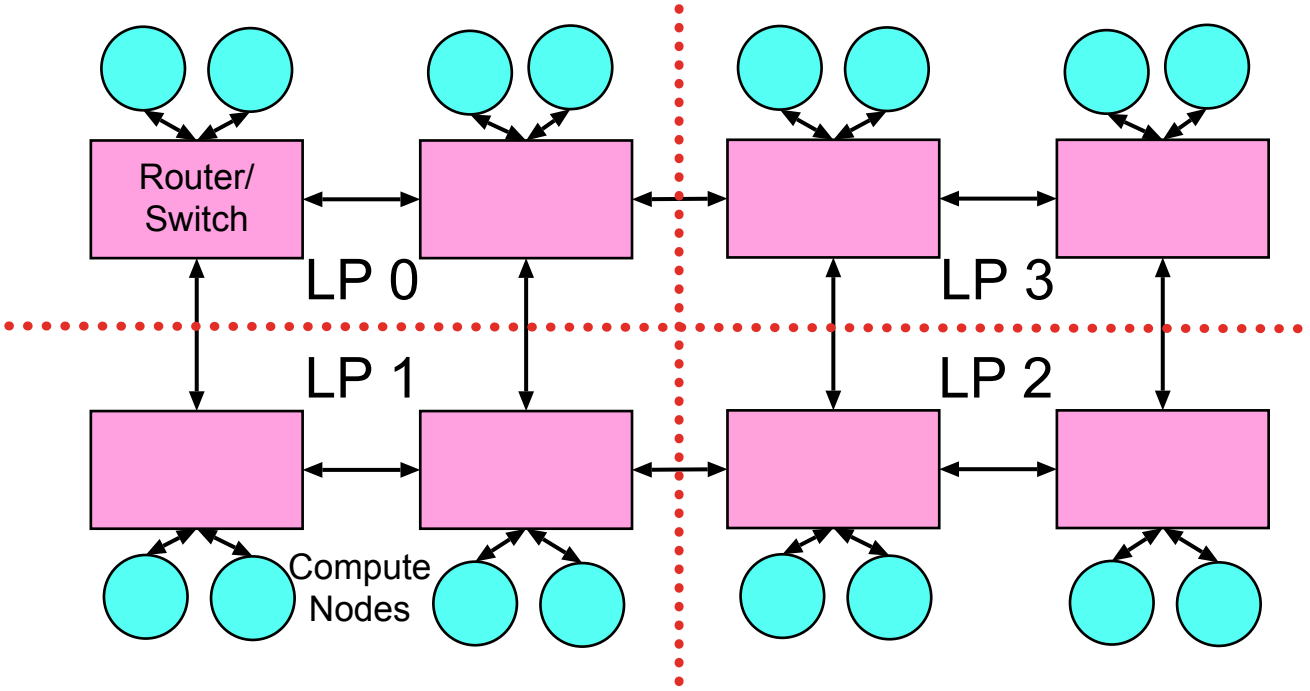
MPI calls start generating network traffic

Compute call might take 5ms but link latencies in the network are only 100ns!

Design considerations for an optimal coarse-grained structural simulator

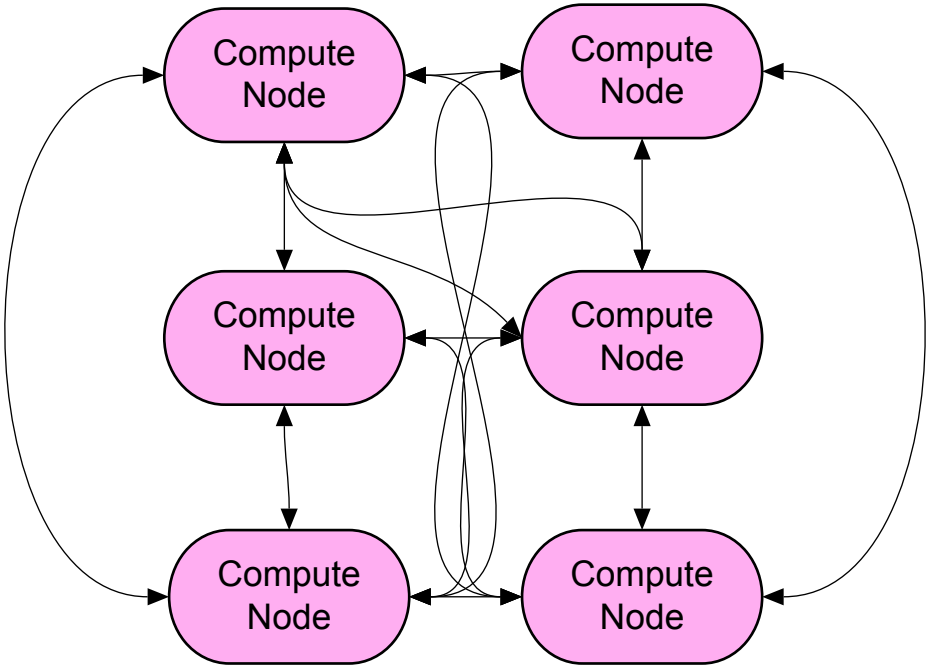
Sandia National Laboratories

- May have a few events per time window – or might have a lot
- Can have large gaps - time windows with no events in them
- Huge number of components, but with the same link latency



Design considerations for a simulator based on analytical models

- Only a few events per time window
- Many different components, but all connected to each other



$$\Delta T = \alpha + \beta N$$

α = Latency

β = Inverse bandwidth

N = Message size

Unifying elements across all fidelities

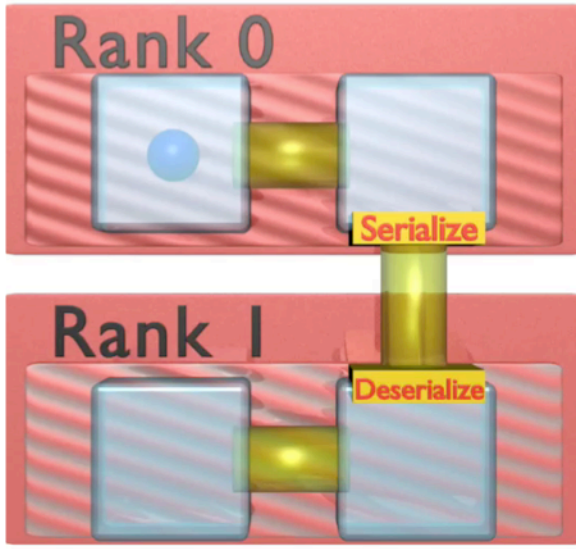
Sandia National Laboratories

- Sending network messages
  - Portability layer to network APIs
  - Serialization library for event objects
- Local/global virtual time
  - Event ordering and correctness
  - Model input and statistics collection
- Partitioning components across parallel workers
  - Mapping of LPs to physical nodes
  - Optimize partition for cheaper communication
- Managing/scheduling events
  - Map/calendar/heap data structures
  - Cancel events

Unifying elements across all fidelities

Sandia National Laboratories

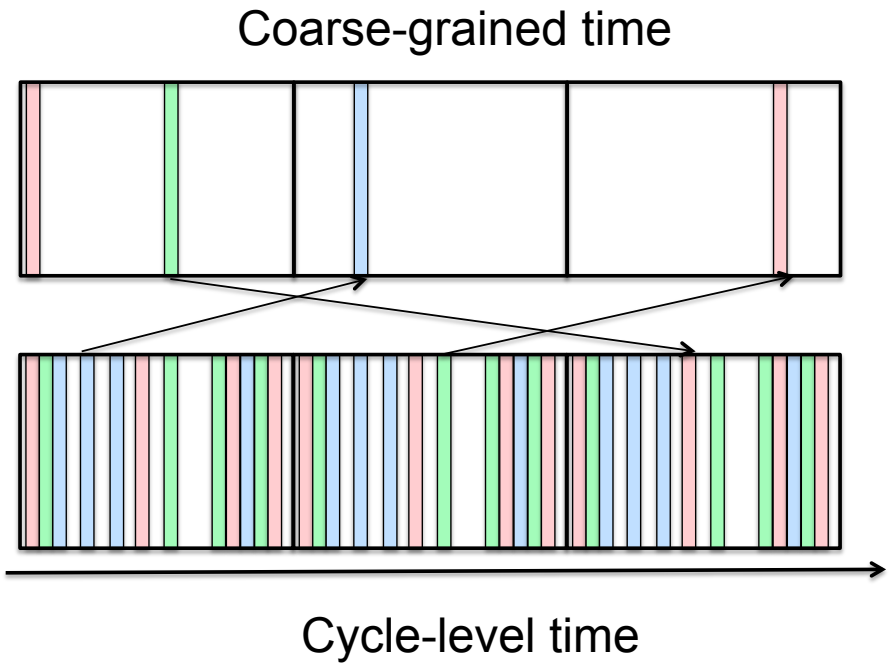
- Sending network messages
  - Portability layer to network APIs
  - Serialization library for event objects
- Local/global virtual time
  - Event ordering and correctness
  - Model input and statistics collection
- Partitioning components across parallel workers
  - Mapping of LPs to physical nodes
  - Optimize partition for cheaper communication
- Managing/scheduling events
  - Map/calendar/heap data structures
  - Cancel events



Unifying elements across all fidelities

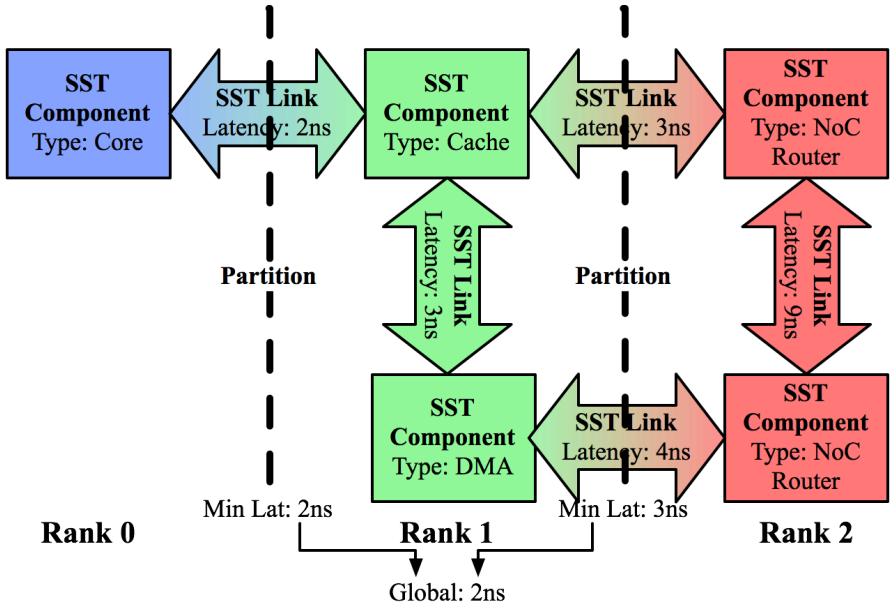
Sandia National Laboratories

- Sending network messages
  - Portability layer to network APIs
  - Serialization library for event objects
- Local/global virtual time
  - Event ordering and correctness
  - Model input and statistics collection
- Partitioning components across parallel workers
  - Mapping of LPs to physical nodes
  - Optimize partition for cheaper communication
- Managing/scheduling events
  - Map/calendar/heap data structures
  - Cancel events



Unifying elements across all fidelities

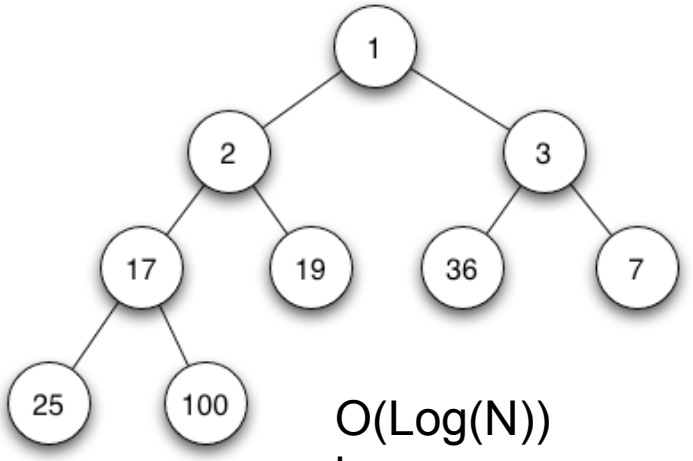
- Sending network messages
  - Portability layer to network APIs
  - Serialization library for event objects
- Local/global virtual time
  - Event ordering and correctness
  - Model input and statistics collection
- Partitioning components across parallel workers
  - Mapping of LPs to physical nodes
  - Optimize partition for cheaper communication
- Managing/scheduling events
  - Map/calendar/heap data structures
  - Cancel events



Unifying elements across all fidelities

Sandia National Laboratories

- Sending network messages
  - Portability layer to network APIs
  - Serialization library for event objects
- Local/global virtual time
  - Event ordering and correctness
  - Model input and statistics collection
- Partitioning components across parallel workers
  - Mapping of LPs to physical nodes
  - Optimize partition for cheaper communication
- Managing/scheduling events
  - Map/calendar/heap data structures
  - Cancel events



O(Log(N)) heap

T=1,T=2, T=3	T=7		T=17, T=19
T=25			T=36
			T=100

O(1) calendar

Disunifying elements across fidelities: partitioning strategy and parallel algorithm

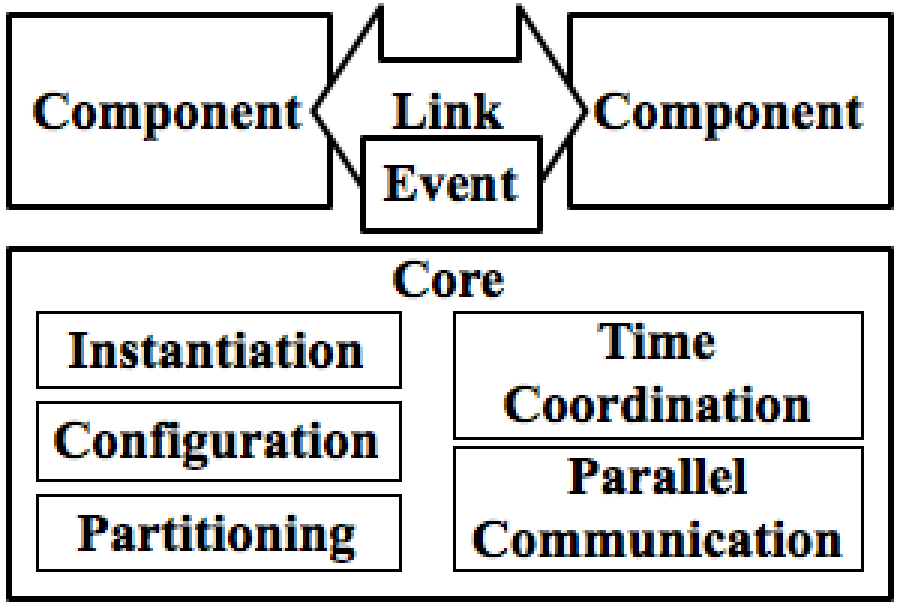
Sandia National Laboratories

- Partitioning strategy
  - Cycle-level has heterogeneous components – partitioning really requires intelligent graph partitioner
  - Coarse-grained has many homogenous components – partitioning still requires intelligent partitioner, but more about minimizing graph connectivity than best link latency
- Parallel algorithm
- Global assumptions on interoperability components
  - Can any memory subsystem model interact with any processor or network model? Or are event messages only “self-compatible”?

Polymorphic components can be tuned for different problems

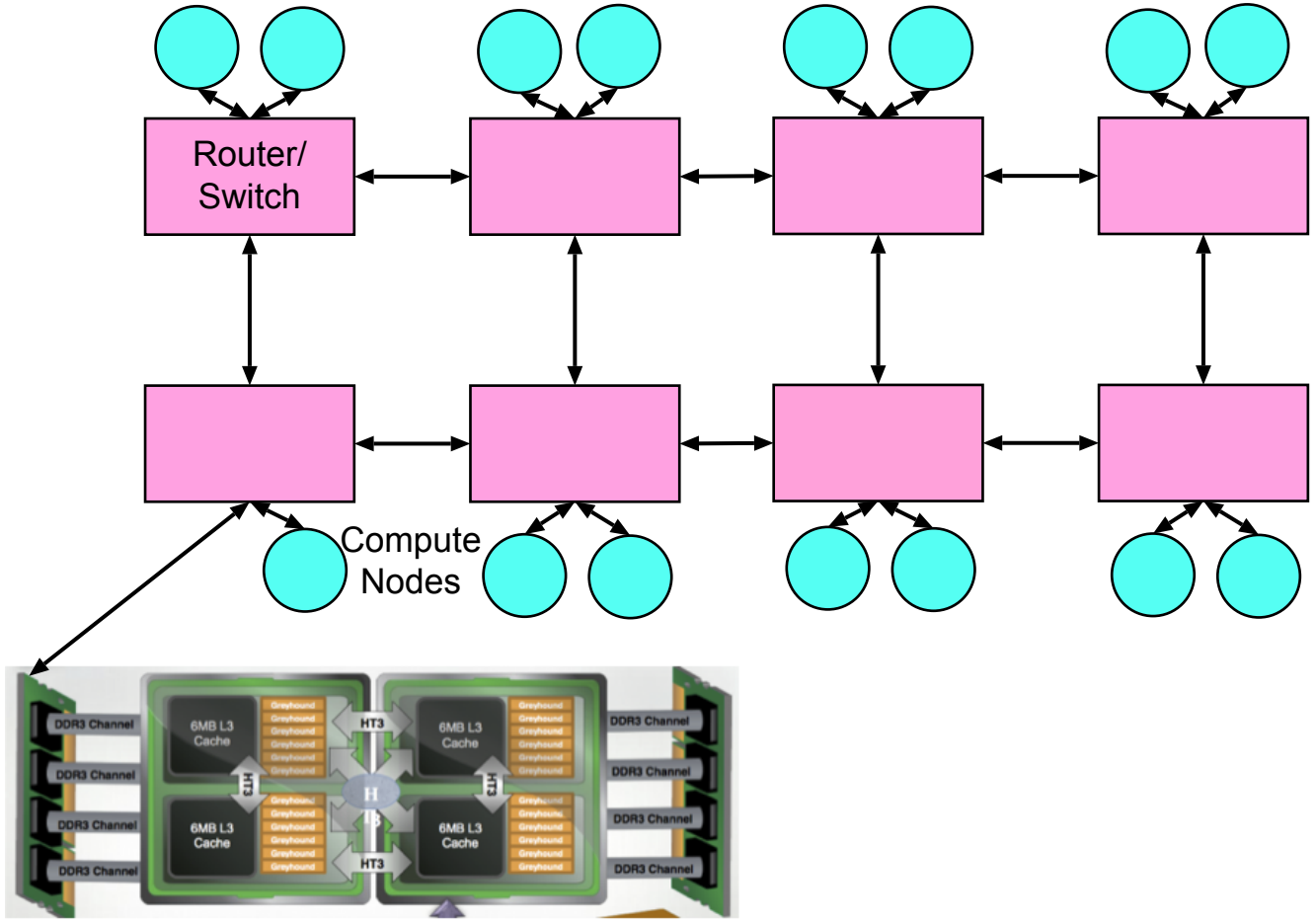
Sandia National Laboratories

- Components don't need to be aware of partitioning strategy
- Components don't need to be aware of parallel algorithm

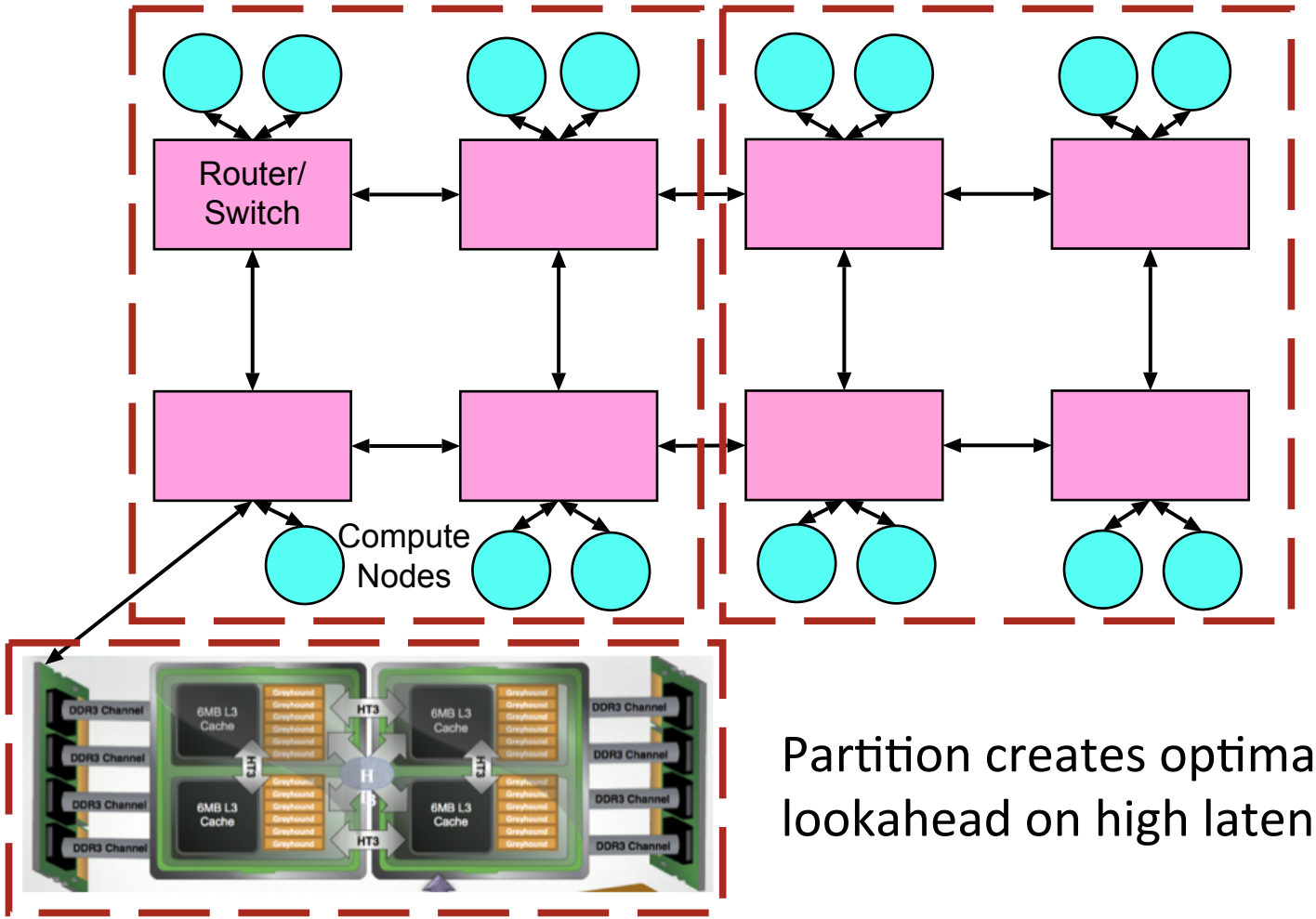


Challenge problem: mixing fidelities

LP with heavy-weight node operating on different time scales, but look ahead determined by coarse-grained links!

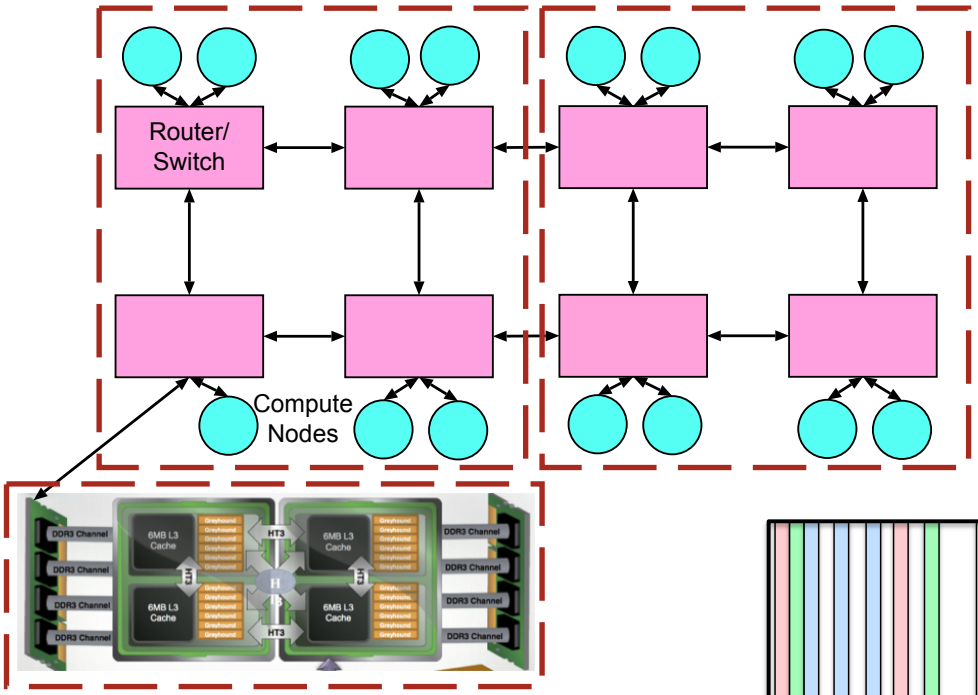


Challenge problem: mixing fidelities

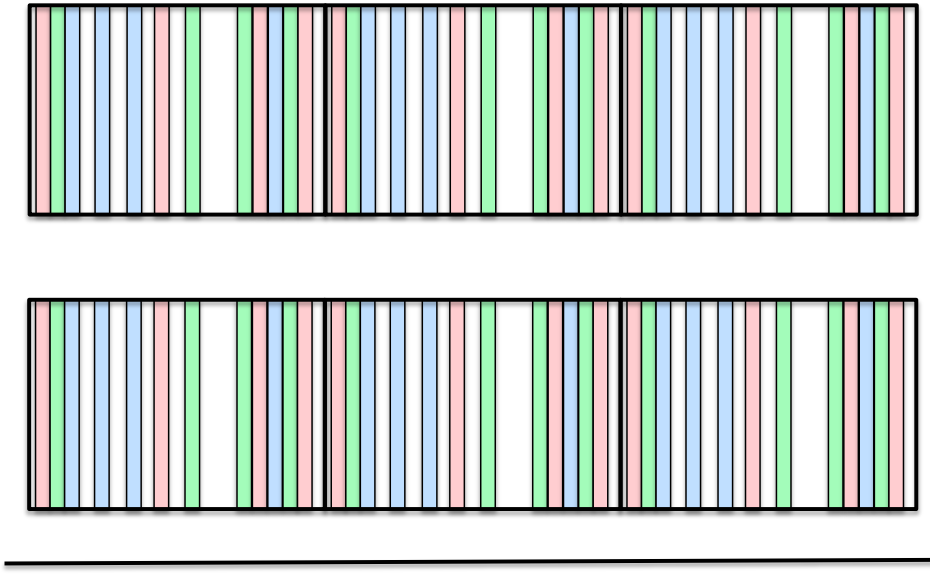


Partition creates optimal lookahead on high latency links

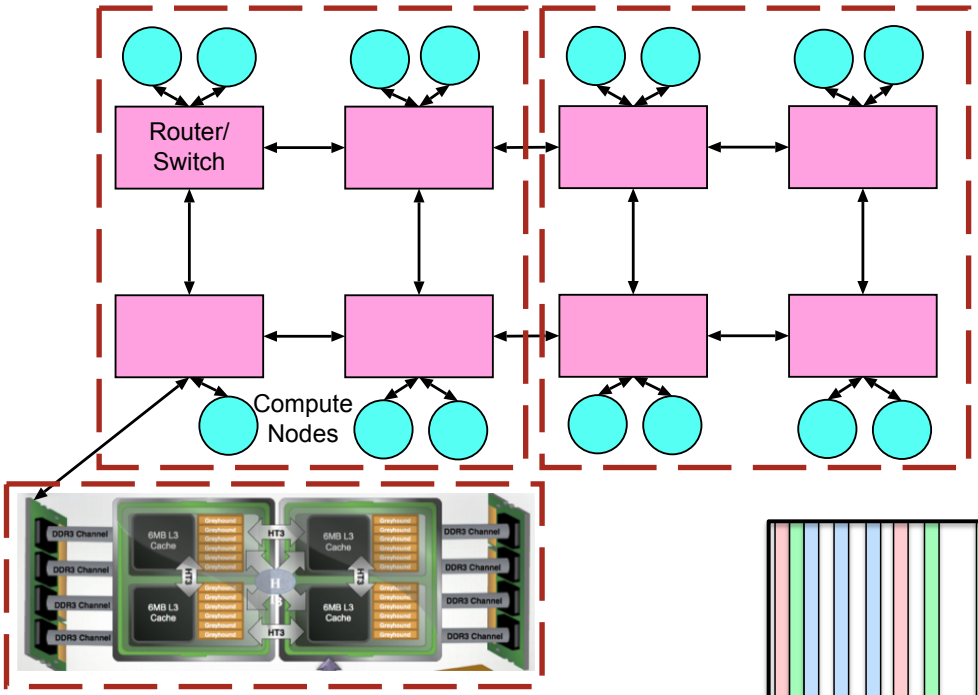
Challenge problem: mixing fidelities

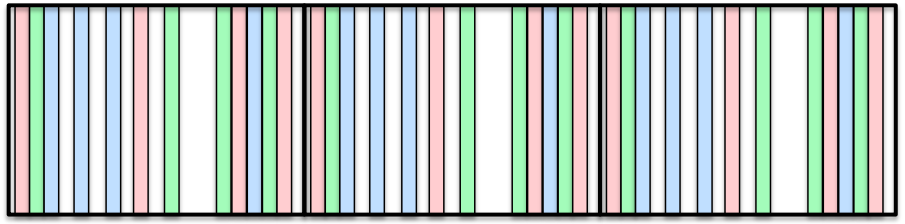


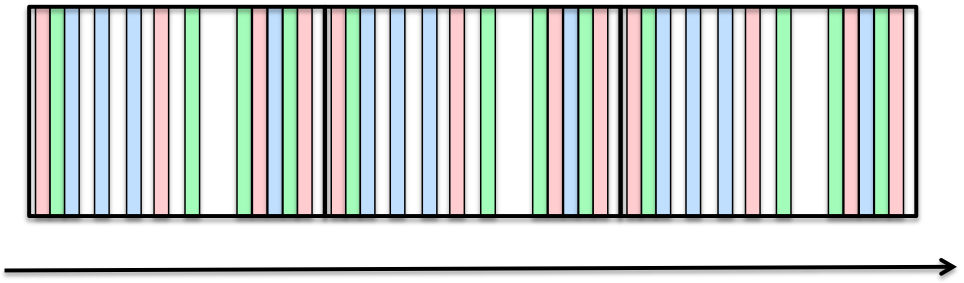
Good partitioning balances number of events per LP



Challenge problem: mixing fidelities





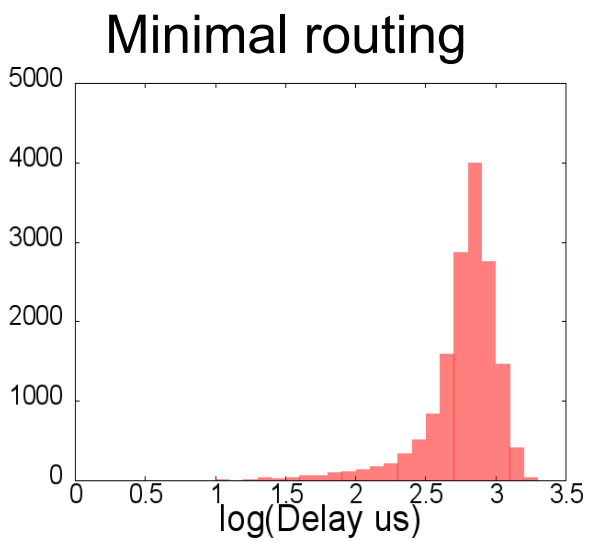


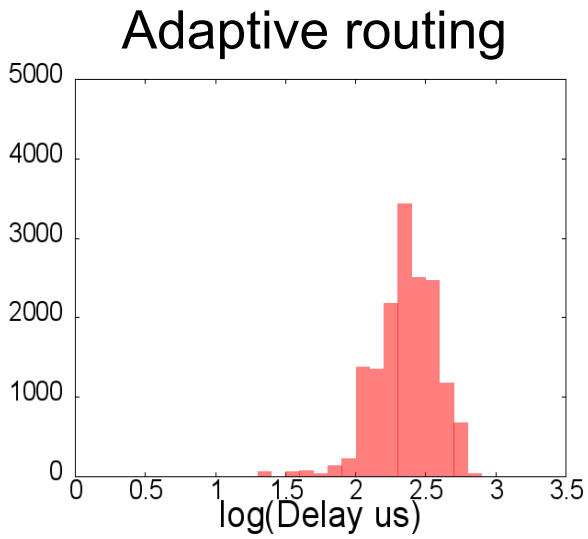
Simulator core problem, not a domain-specific problem

Challenge problem: histogram of message delays in PDES run

Sandia National Laboratories

- Tag packet going out, going in with time
  - Add single field to existing network message object
  - Notion of global time
- Histogram object that reduces/collects data
  - hist->add_one(delay)
  - Object must output usable data format/figure at end of simulation
  - Data must be reduced across LPs

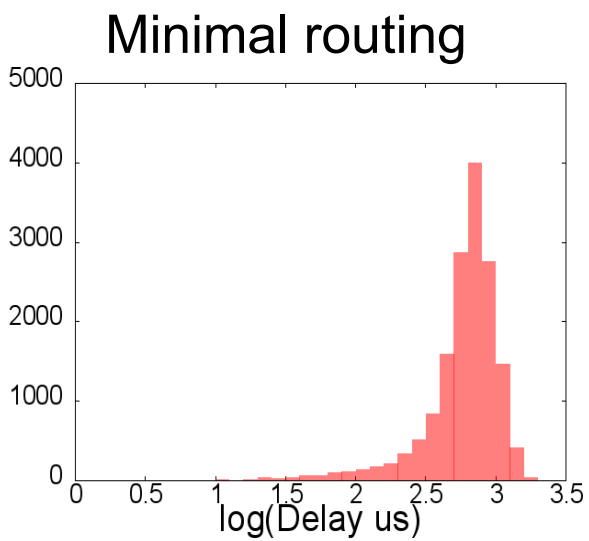


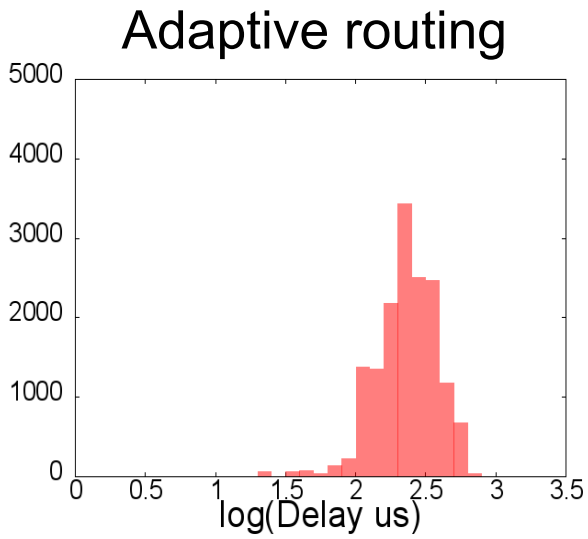


Challenge problem: histogram of message delays in PDES run

Sandia National Laboratories

Majority of work should already be done for us in common core!

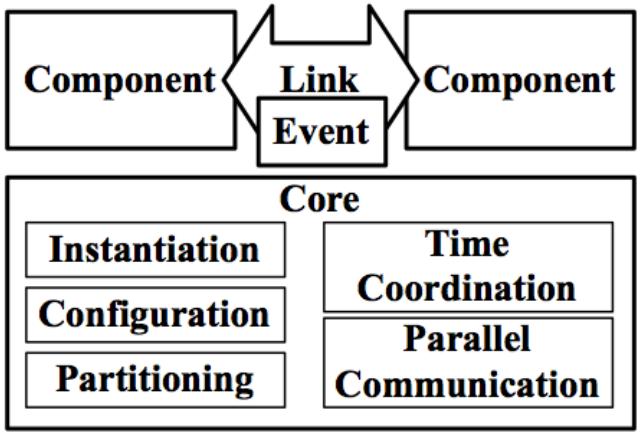


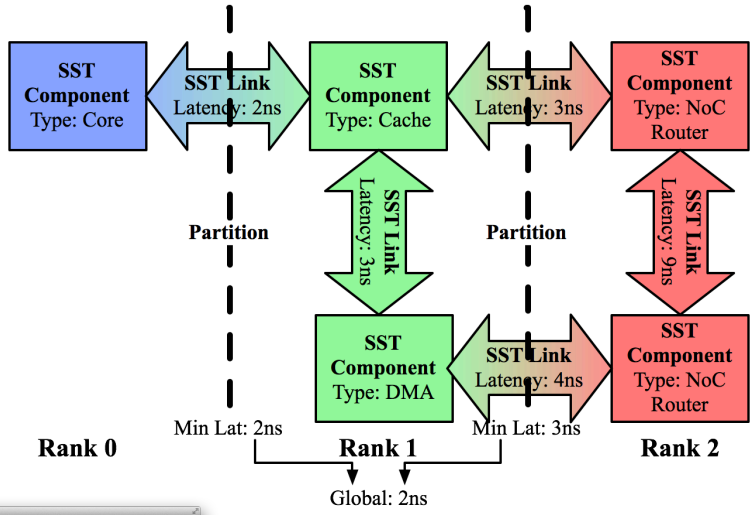


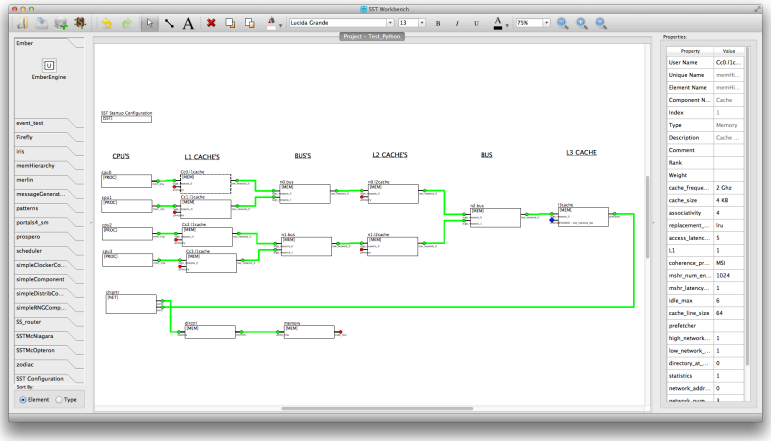
MODSIM is Camp David, not Sinai

Sandia National Laboratories

Structural simulation toolkit at Sandia (SST) is a jumping off point for discussing universal simulation standards, not C++ framework written on stone tablets!



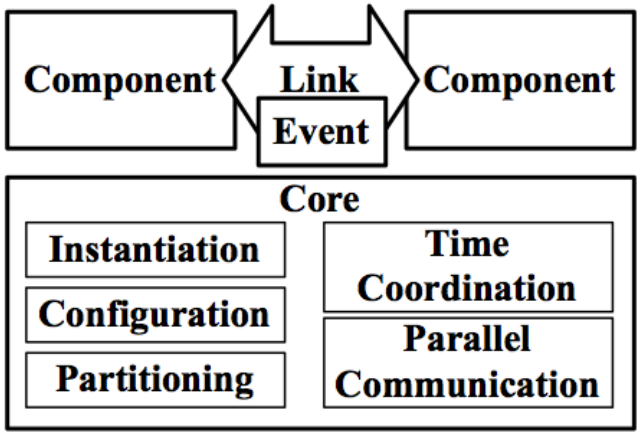


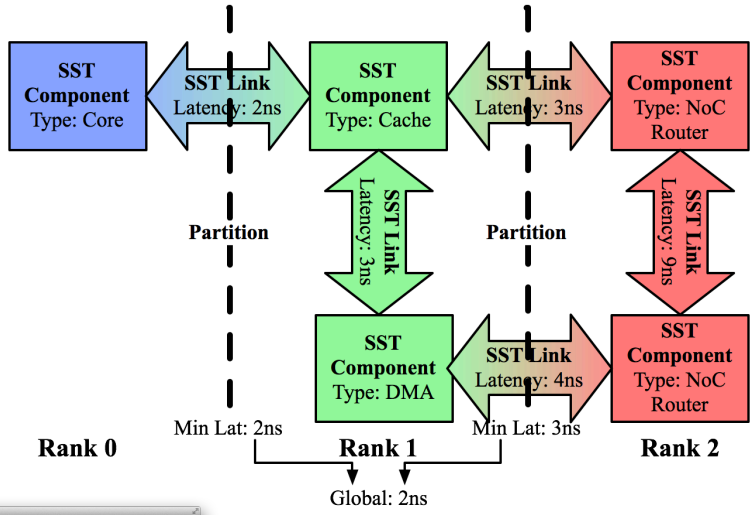


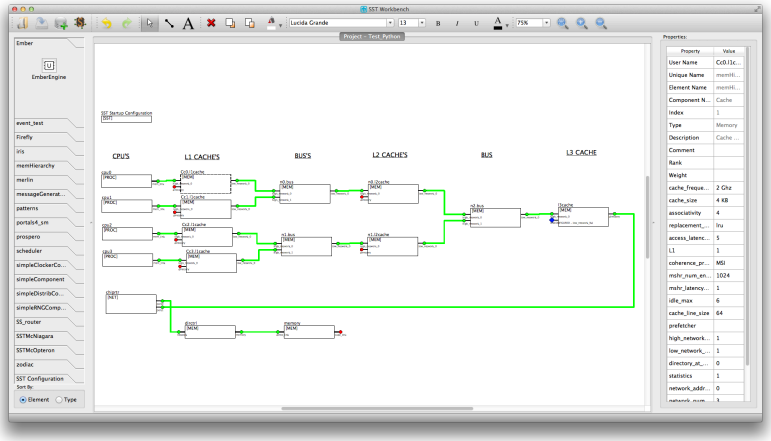
MODSIM is Camp David, not Sinai

Sandia National Laboratories

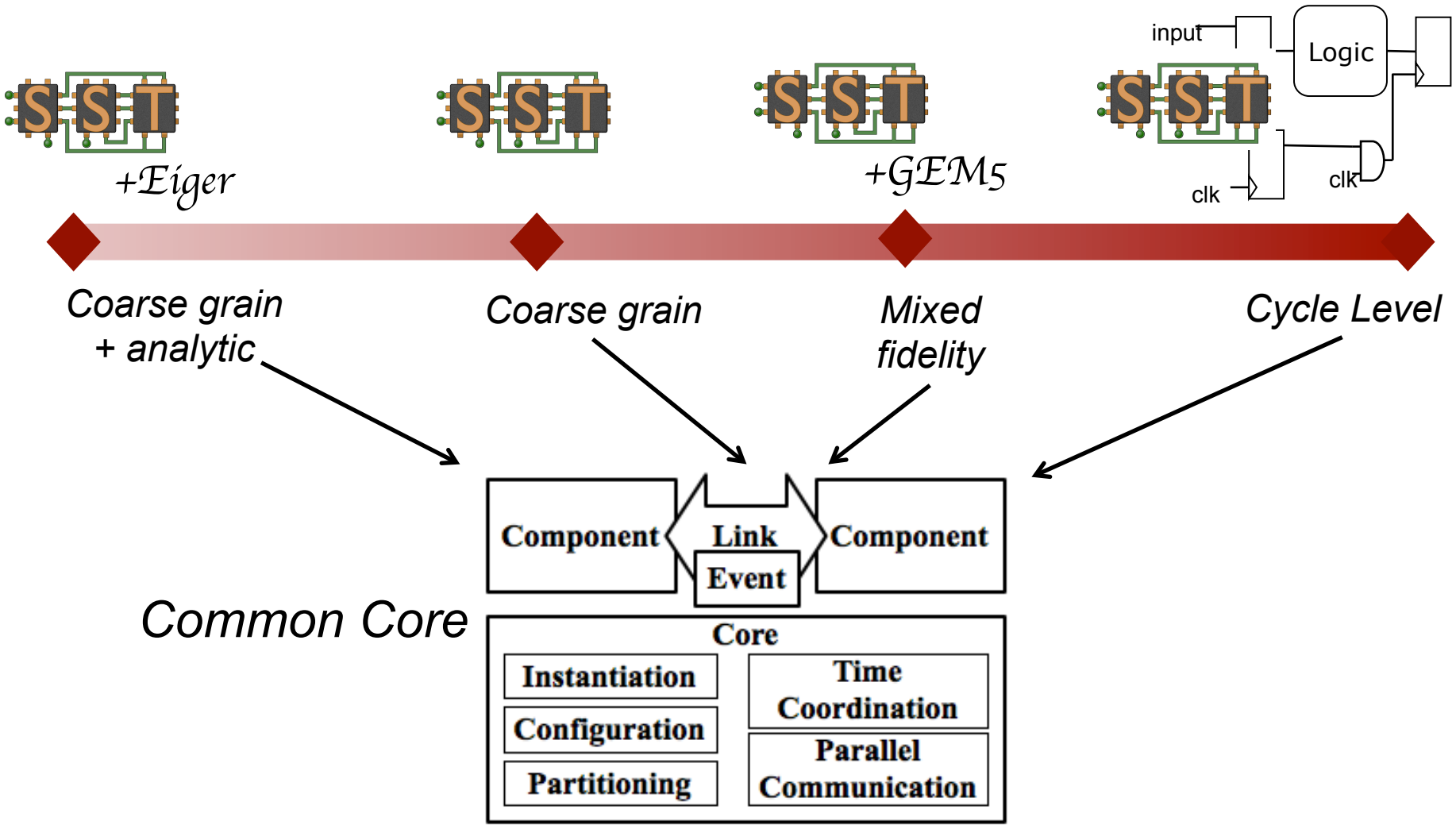
We think our PDES core is mature and lightweight enough to make your life easer – enable you to use your own code, not force you to use ours!







Range of simulation: success stories so far



MODSIM Summary

Sandia National Laboratories

- Gaps:
  - Lack of standards, lack of code reuse
- Bigger picture and potential collaborators:
  - Everyone? Anyone who wants to scale experiments through PDES or wants to compose models mixing different fidelity/physics
- What would make it easier to leverage results from other groups?
  - If you find yourself writing a PDES core, who you gonna call…
- ***Development/adoption of standards will be driven by collaboration and refined through use cases***