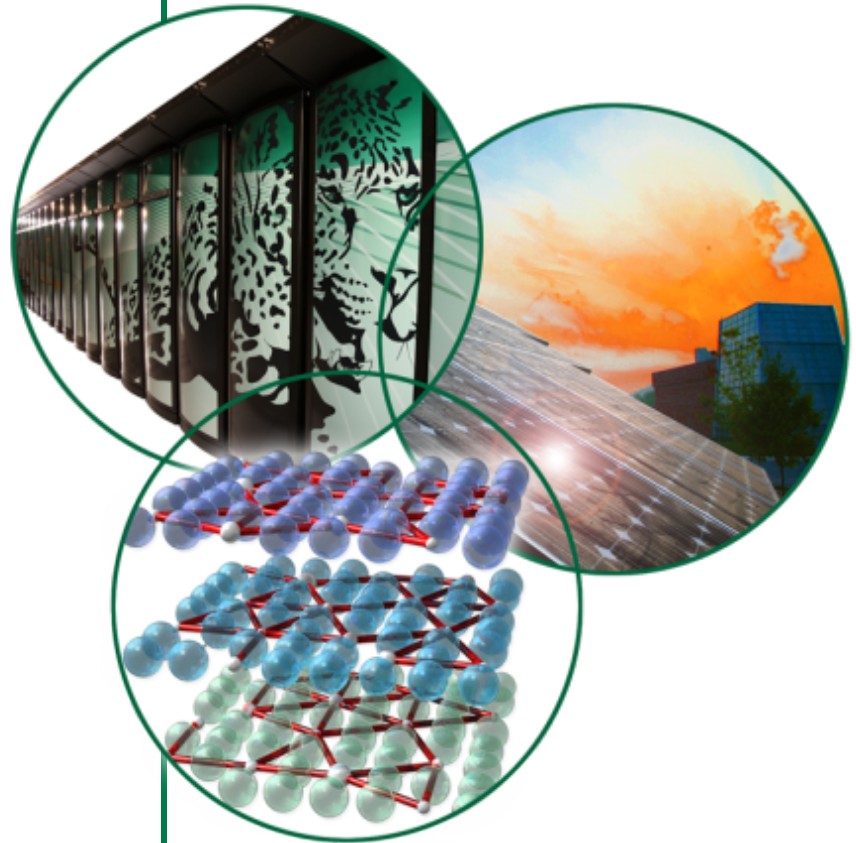
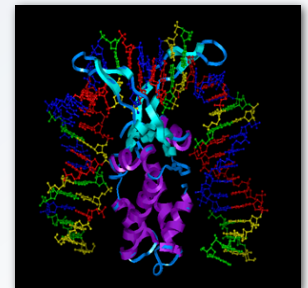
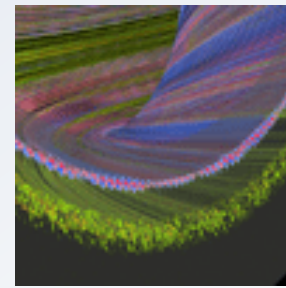
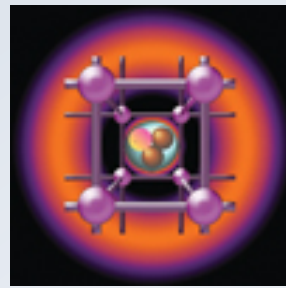
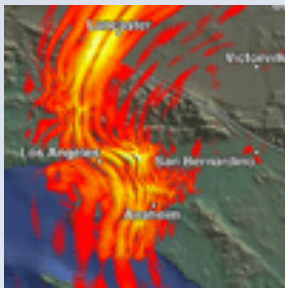
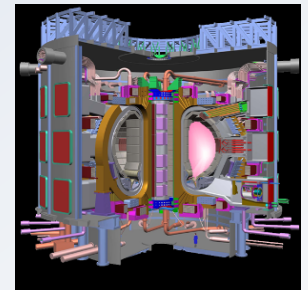
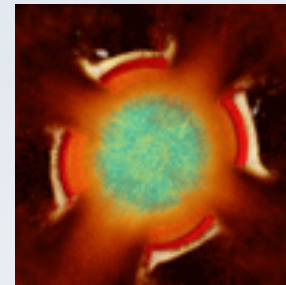
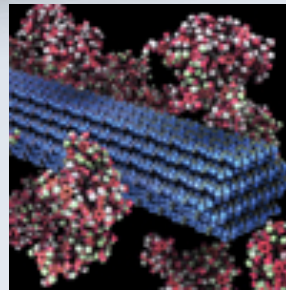
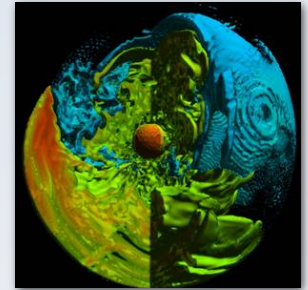
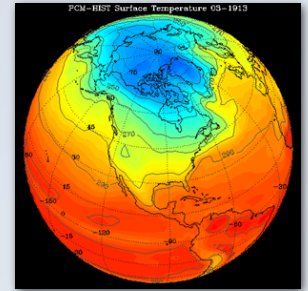


xSim: The Extreme-scale Simulator

Christian Engelmann
Oak Ridge National Laboratory



Scientific Computing and Simulation at ORNL



Motivation

- At the forefront of extreme-scale scientific computing
 - Titan at ORNL: Currently 2nd fastest supercomputer in the world
 - 560,640 cores (AMD Opteron + NVIDIA Kepler GPUs, 17.6 PFlops)
- We are on road to exascale computing: 1,000 Pflop/s by 2023
 - Billions of cores (see next slides)
- There are several major challenges:
 - *Power consumption*: Envelope of ~20-40 MW (drives everything else)
 - *Programmability*: Accelerators and PIM-like architectures
 - *Performance*: Extreme-scale parallelism (up to 1B hardware threads)
 - *Data movement*: Complex memory hierarchy and locality
 - *Data management*: Too much data to track and store
 - *Resilience*: Faults will occur continuously

Discussed Exascale Road Map (2011)

Many design factors are driven by the power ceiling (op. costs)

<i>Systems</i>	<i>2009</i>	<i>2012</i>	<i>2016</i>	<i>2020</i>
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System memory	0.3 PB	1.6 PB	5 PB	10 PB
Node performance	125 GF	200GF	200-400 GF	1-10TF
Node memory BW	25 GB/s	40 GB/s	100 GB/s	200-400 GB/s
Node concurrency	12	32	O(100)	O(1000)
Interconnect BW	1.5 GB/s	22 GB/s	25 GB/s	50 GB/s
System size (nodes)	18,700	100,000	500,000	O(million)
Total concurrency	225,000	3,200,000	O(50,000,000)	O(billion)
Storage	15 PB	30 PB	150 PB	300 PB
IO	0.2 TB/s	2 TB/s	10 TB/s	20 TB/s
MTTI	1-4 days	5-19 hours	50-230 min	22-120 min
Power	6 MW	~10MW	~10 MW	~20 MW

Discussed Exascale Road Map (2014)

Remember, this road map is a moving target.

<i>Systems</i>	<i>2009</i>	<i>2012</i>	<i>2017/18</i>	<i>2023</i>
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System memory	0.3 PB	1.6 PB	5 PB (NVRAM?)	10 PB ?
Node performance	125 GF	200GF	200-400 GF ?	1-10TF ?
Node memory BW	25 GB/s	40 GB/s	100 GB/s ?	200-400 GB/s ?
Node concurrency	12	32	O(100) ?	O(1000) ?
Interconnect BW	1.5 GB/s	22 GB/s	25 GB/s	50 GB/s
System size (nodes)	18,700	100,000	500,000 ?	O(million) ?
Total concurrency	225,000	3,200,000	O(50,000,000) ?	O(billion) ?
Storage	15 PB	30 PB	150 PB ?	300 PB ?
IO	0.2 TB/s	2 TB/s	10 TB/s ?	20 TB/s ?
MTTI	1-4 days	5-19 hours	50-230 min ?	22-120 min ?
Power	6 MW	~10MW	~10 MW ?	~20 MW ?

HPC Hardware/Software Co-Design

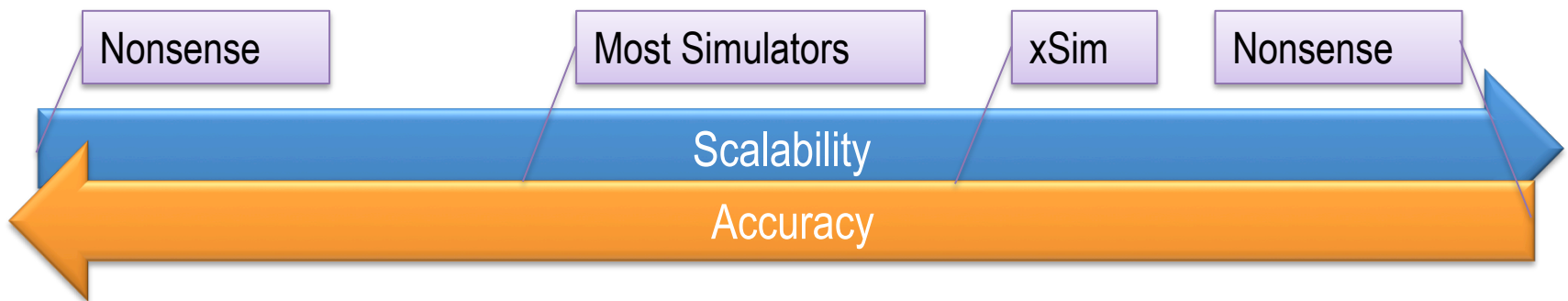
- Helps closing the system-peak vs. application performance gap
- Develops HPC systems and applications jointly to deal with architecture limitations and application needs
- Employs hardware prototypes of future HPC architectures for performance evaluation at small scale
- Utilizes software simulation of future HPC architectures for performance evaluation at small and large scale
- Simulations investigate the impact of different architectural parameters on parallel application performance
- Parallel discrete event simulation (PDES) is often used with cycle accuracy at small scale and less accuracy at large scale

Objectives

- Develop a HPC performance/resilience/power co-design toolkit with corresponding definitions, metrics, and methods
- Evaluate the performance, resilience, and power cost/benefit trade-off of different architectures *at scale*
- Help to coordinate interfaces and responsibilities of individual hardware and software components
- Provide the tools and data needed to decide on future architectures using the key design factors: performance, resilience, and power consumption
- Enable feedback to vendors and application developers

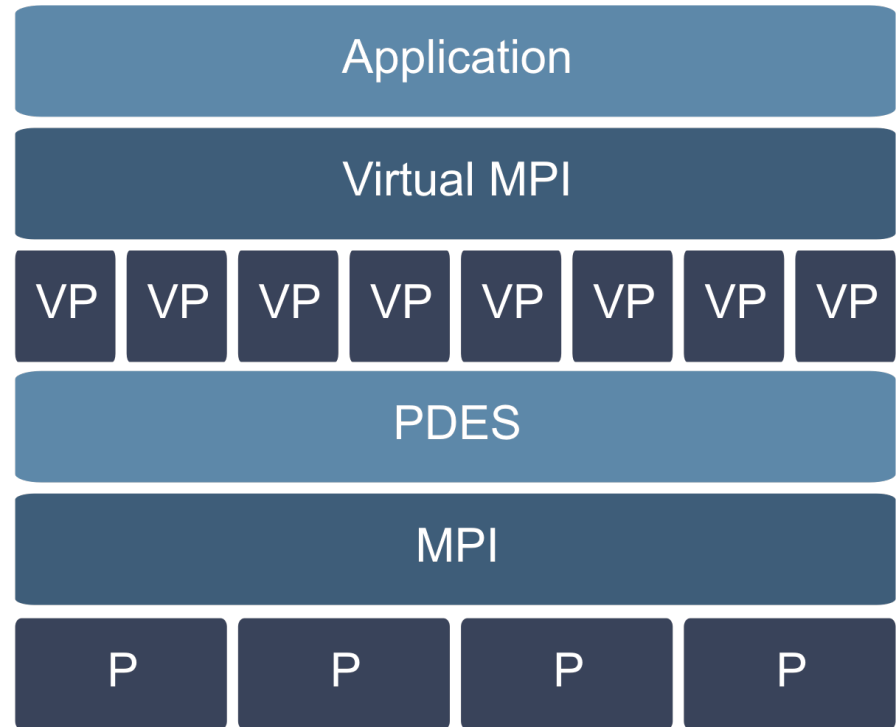
Overall Approach

- Execution of real applications, algorithms, or their models atop a simulated HPC environment at scale for:
 - Performance evaluation, including identification of resource contention and underutilization issues
 - Investigation at extreme scale, beyond the capabilities of existing simulation efforts
- xSim: A highly scalable solution that trades off accuracy



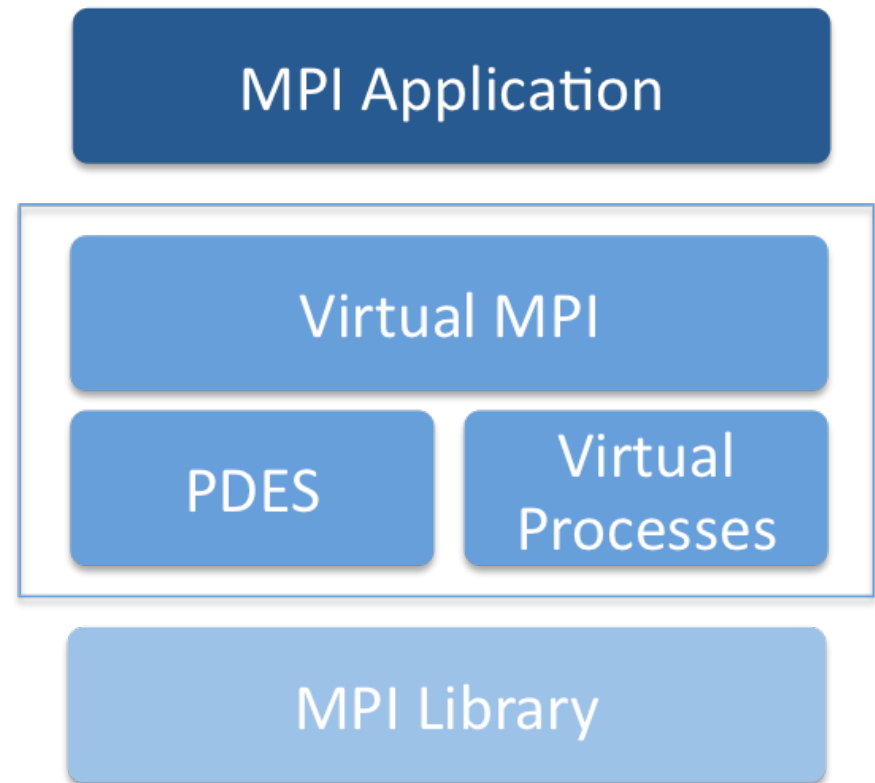
Technical Approach

- Combining highly oversubscribed execution, a virtual MPI, & a time-accurate PDES
- PDES uses the native MPI and simulates virtual procs.
- The virtual procs. expose a virtual MPI to applications
- Applications run within the context of virtual processors:
 - Global and local virtual time
 - Execution on native processor
 - Processor and network model



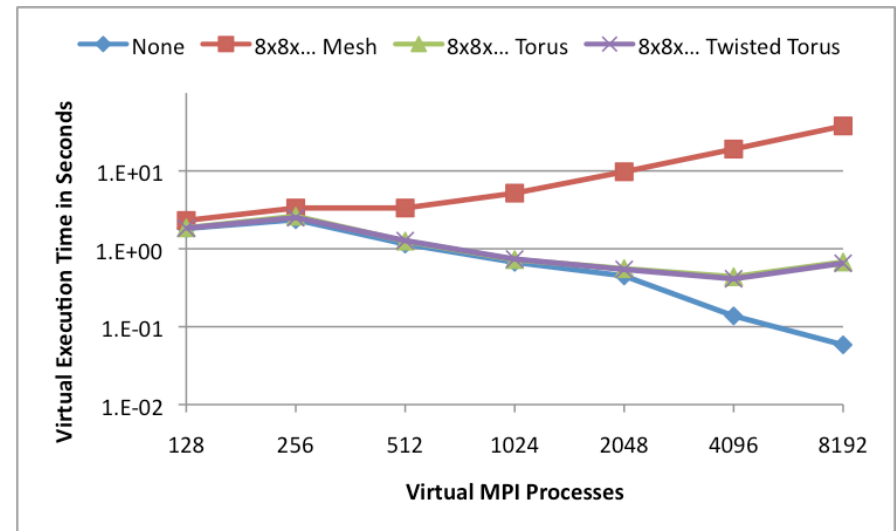
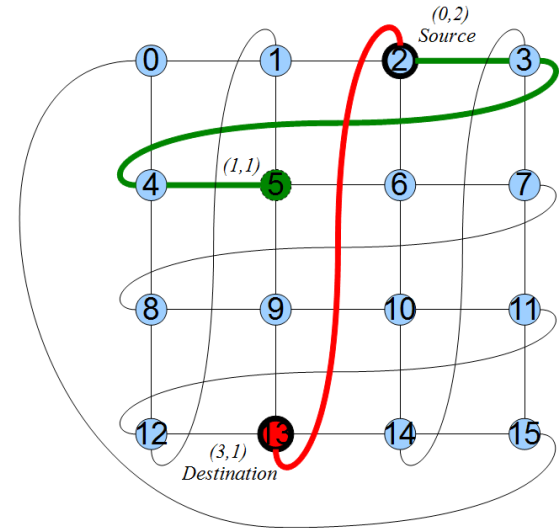
Simulator Design

- The simulator is a library
- Utilizes PMPI to intercept MPI calls and to hide the PDES
- Implemented in C with 2 threads per native process
- Support for C/Fortran MPI
- Easy to use:
 - Compile with xSim header
 - Link with the xSim library
 - Execute: *mpirun -np <np> <application> -xsim-np <vp>*



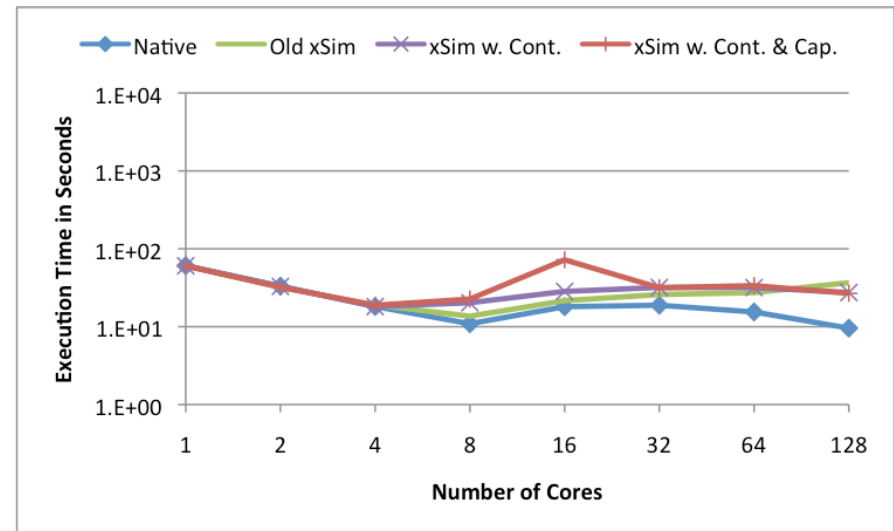
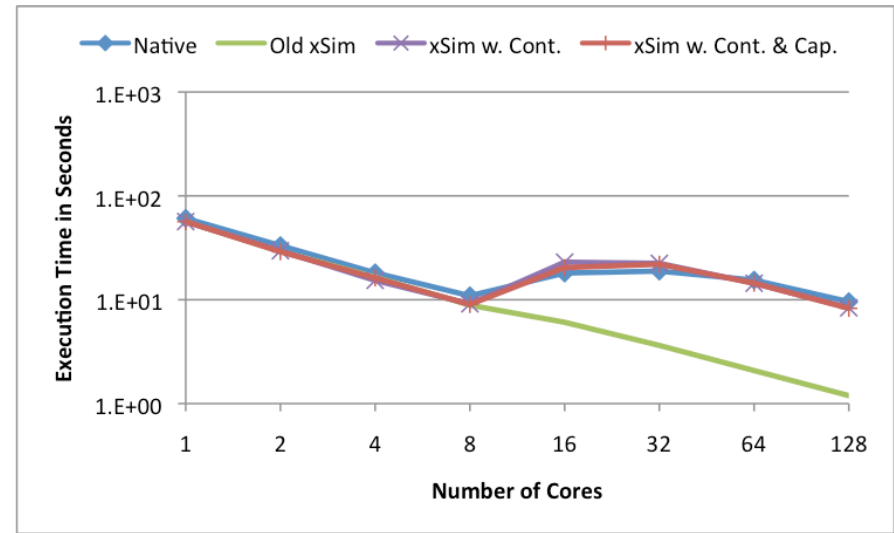
Processor and Network Models

- Scaling processor model
 - Relative to native execution
- Configurable network model
 - Link latency & bandwidth
 - Contention and routing
 - Star, ring, mesh, torus, twisted torus, and tree
 - Hierarchical combinations, e.g., on-chip, on-node, & off-node
 - Simulated rendezvous protocol
- Example: NAS MG in a dual-core 3D mesh or twisted torus



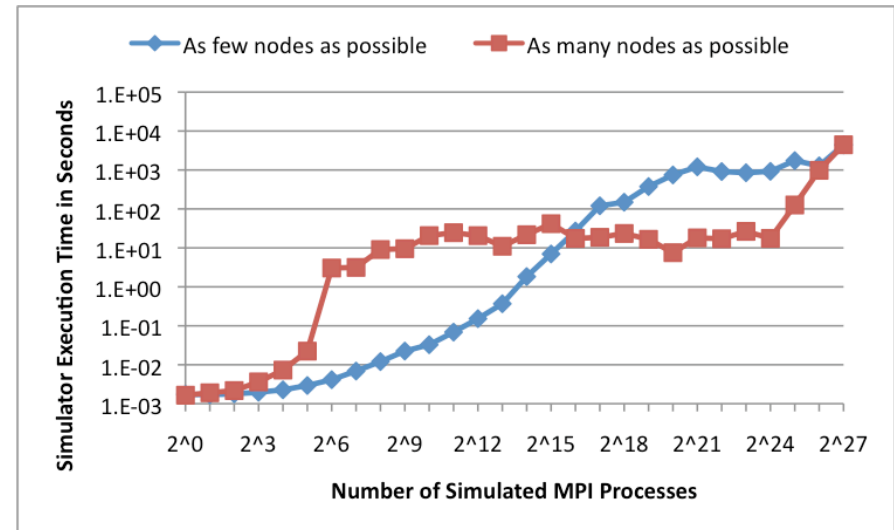
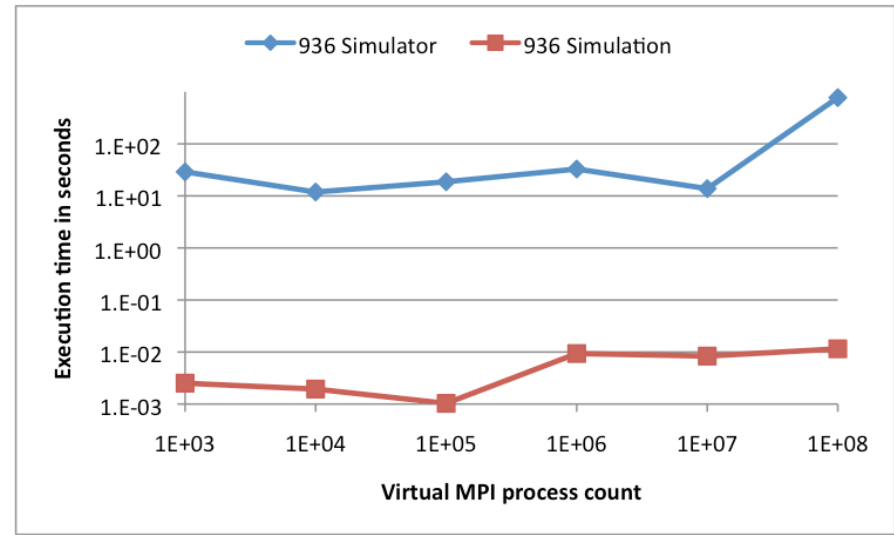
Network Contention Modeling

- Access contention modeling
- Capacity contention modeling
- NAS Parallel Benchmark:
 - IS, an integer sort (class C)
- Simulation error reduced:
 - From 87.5% to 17.5% for IS
- Simulation overhead reduced:
 - From 283% to 187% for IS



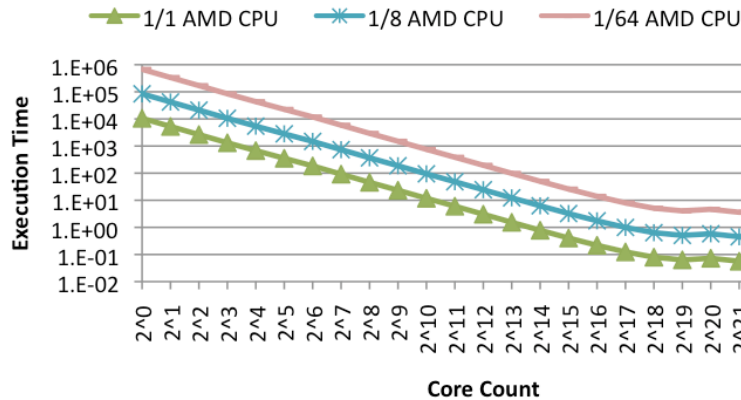
Scaling up by Oversubscribing

- Running on a 960-core Linux cluster with 2.5TB RAM
- Executing 134,217,728 (2^{27}) simulated MPI ranks
 - 1TB total user-space stack
 - 0.5TB total data segment
 - 8kB user-space stack per rank
 - 4kB data segment per rank
- Running MPI hello world
 - Native vs. simulated time
 - Native time using as few or as many nodes as possible

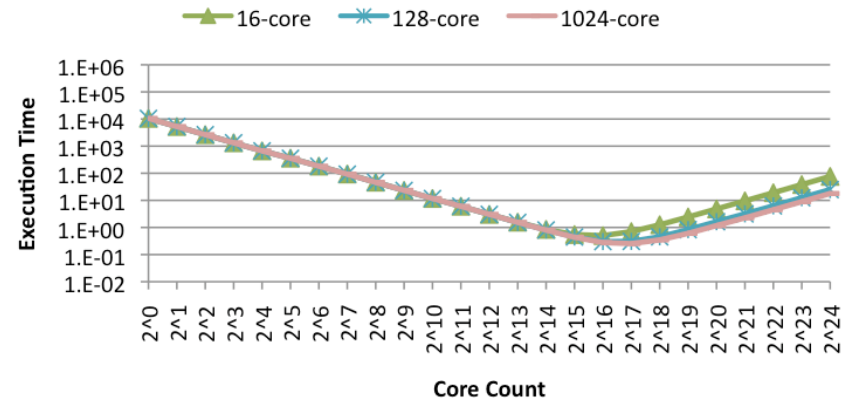


Scaling a Monte Carlo Solver to 2^{24} Ranks

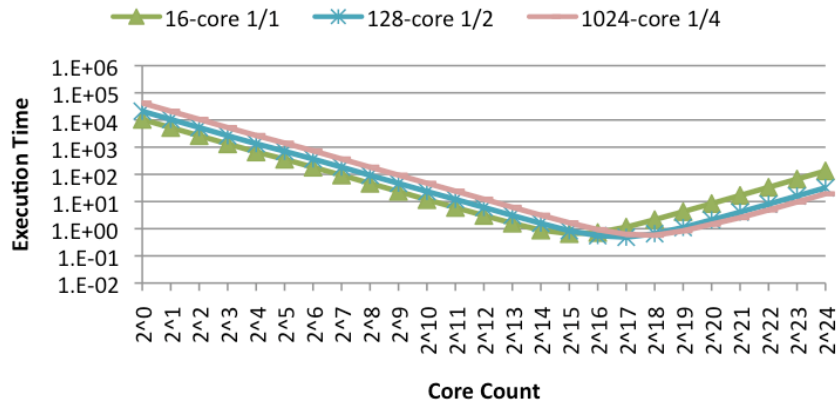
Perfect Network - 128x



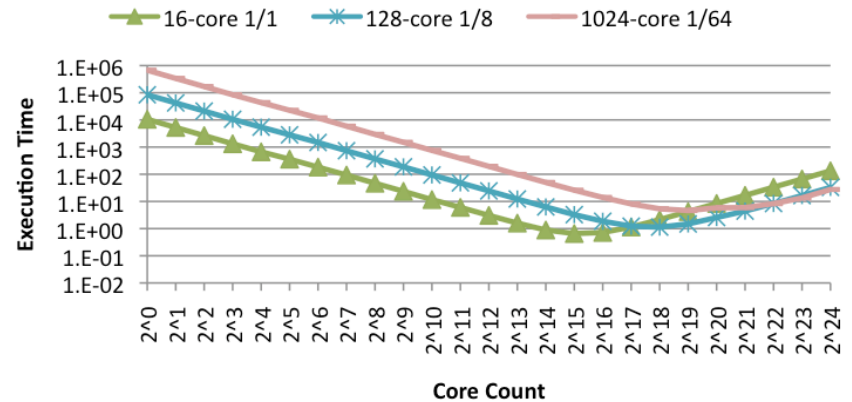
8-8-X Twisted Torus - 128x



16-16-X Twisted Torus - 128x



16-16-X Twisted Torus - 128x



Simulating OS Noise at Extreme Scale

- OS noise injection into a simulated HPC system
 - Part of the processor model
 - Synchronized OS noise
 - Random OS noise
- Experiment: 128x128x128 3-D torus with 1 mμ s latency and 32 GB/s bandwidth

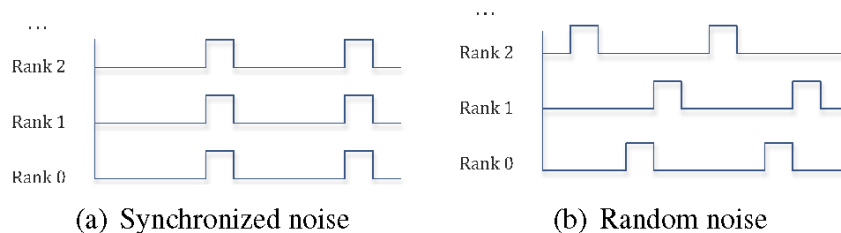
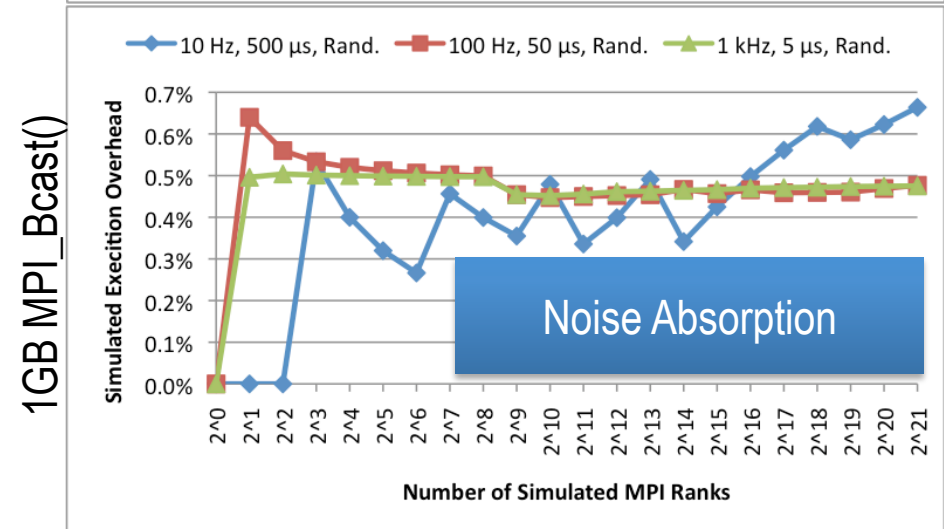
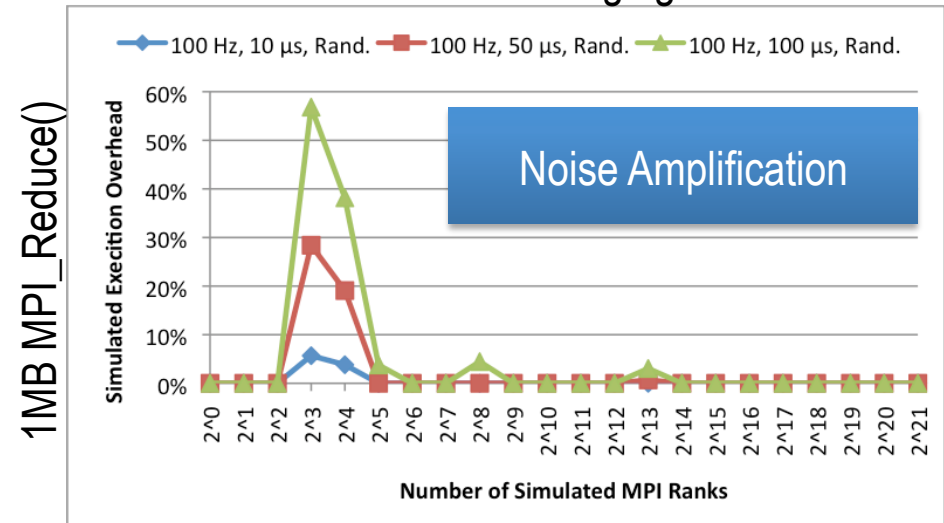


Figure 2. Generated noise injected into the simulation

Random OS Noise with Changing Noise Period



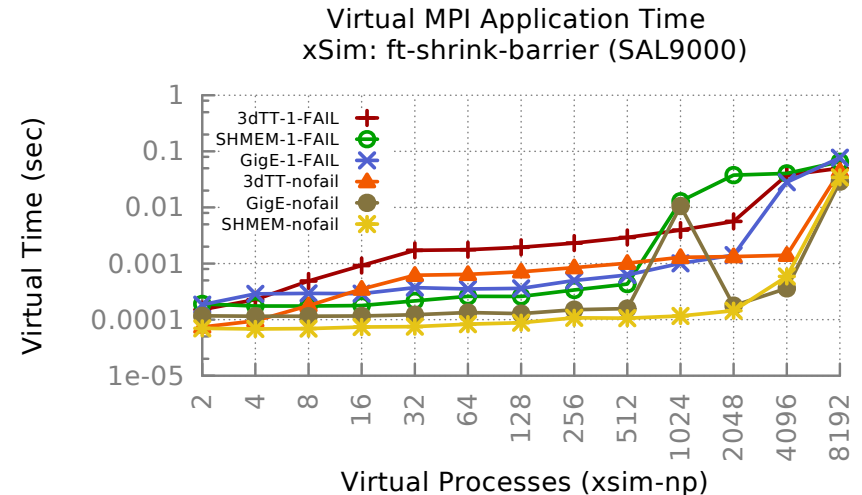
Random OS Noise with Fixed Noise Ratio

Resilience Simulation Features

- Simulated MPI process failure
 - Accurate fault injection, propagation and detection based on modeled architecture
- Simulated MPI application checkpoint, abort, and restart
 - Support for checkpoint/restart cycles until completion
- Simulated fault tolerant MPI
 - Support for resilient solvers using proposed fault tolerant MPI extensions

Table 1: Varying the checkpoint interval and system MTTF

$MTTF_S$	C	E_1	E_2	F	$MTTF_A$
—	1,000	5,248 s	—	0	—
6,000 s	500	5,258 s	7,957 s	1	3,978 s
6,000 s	250	6,377 s	7,074 s	1	3,537 s
6,000 s	125	6,601 s	6,750 s	1	3,375 s
3,000 s	500	5,258 s	10,584 s	2	3,528 s
3,000 s	250	6,377 s	8,618 s	2	2,872 s
3,000 s	125	6,601 s	7,948 s	2	2,649 s



Conclusion

- The Extreme-scale Simulator (xSim) is a performance investigation toolkit that utilizes a PDES and oversubscription
- It supports a basic processor model and an advanced network model to simulate a future-generation HPC system
- It is the first performance toolkit supporting MPI process failure injection, checkpoint/restart, and MPI-FT

Questions

