Rapid Performance Prediction and Sustainability of High Performance Computing

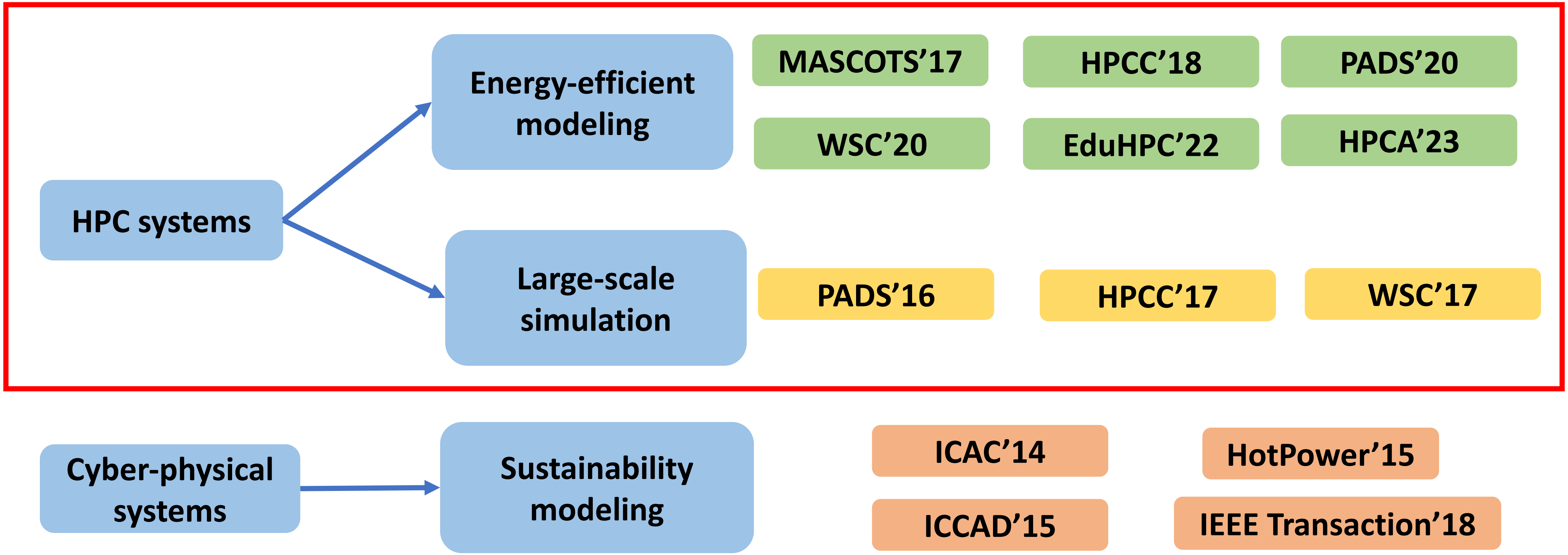
Kishwar Ahmed

Electrical Engineering and Computer Science

The University of Toledo

My Research

Efficient Modeling and Simulation of Large Complex Systems



HPC systems

Energy-efficient modeling

MASCOTS’17

HPCC’18

PADS’20

WSC’20

EduHPC’22

HPCA’23

Large-scale simulation

PADS’16

HPCC’17

WSC’17

Cyber-physical systems

Sustainability modeling

ICAC’14

HotPower’15

ICCAD’15

IEEE Transaction’18

Outline

- **Background and motivation**
- Power and performance prediction modeling of HPC
- Energy-efficient modeling of HPC
- Path forward

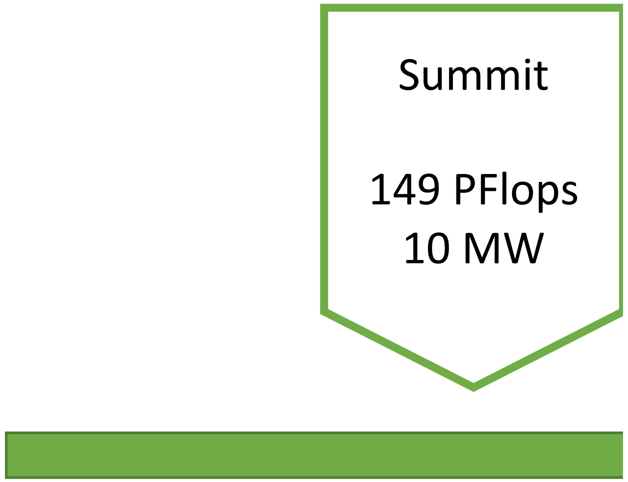
High-performance computing (HPC) system

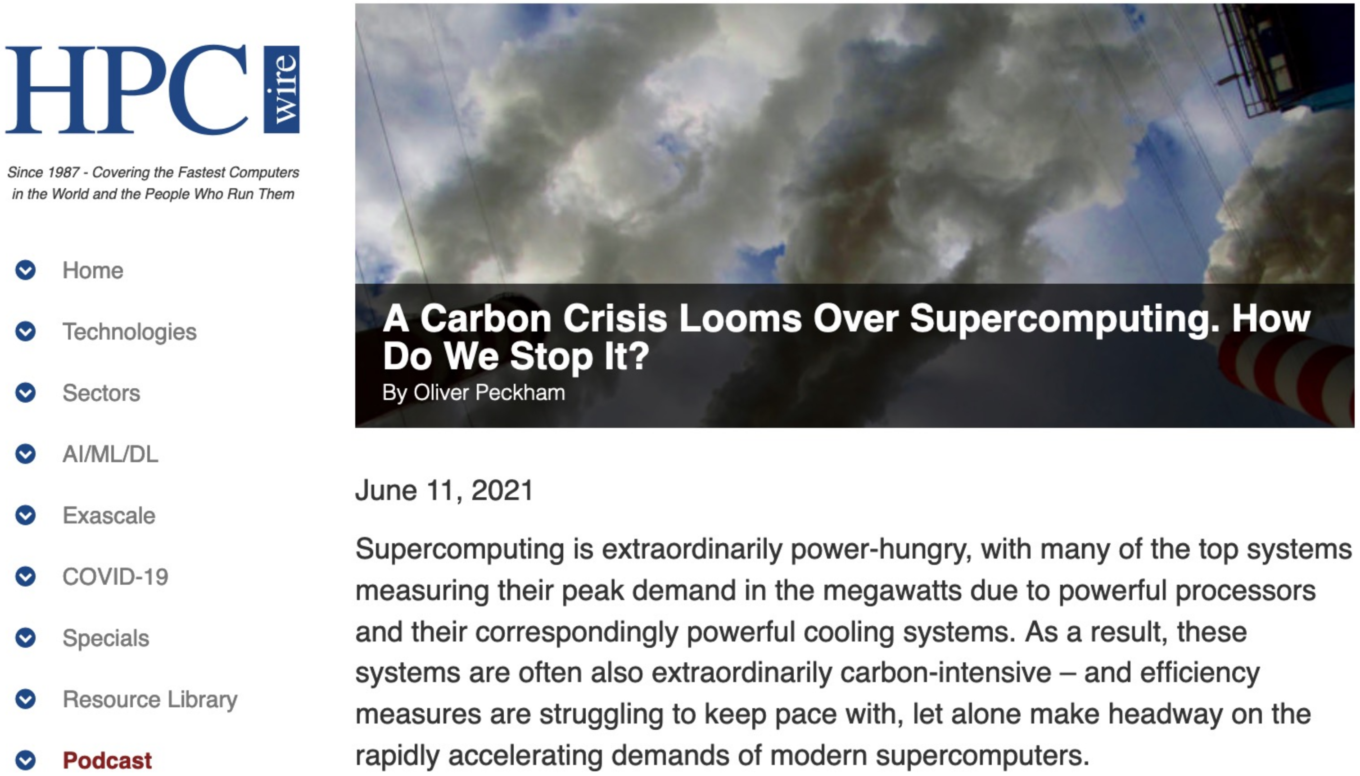
CRAY
OAK RIDGE National Laboratory
U.S. DEPARTMENT OF ENERGY
Hewlett Packard Enterprise
AMD
FRONTIER

Frontier supercomputer @ORNL, Oak Ridge, TN

HPC is Power-Hungry!

Top supercomputers
(June 2022)





A Carbon Crisis Looms Over Supercomputing. How Do We Stop It?

By Oliver Peckham

June 11, 2021

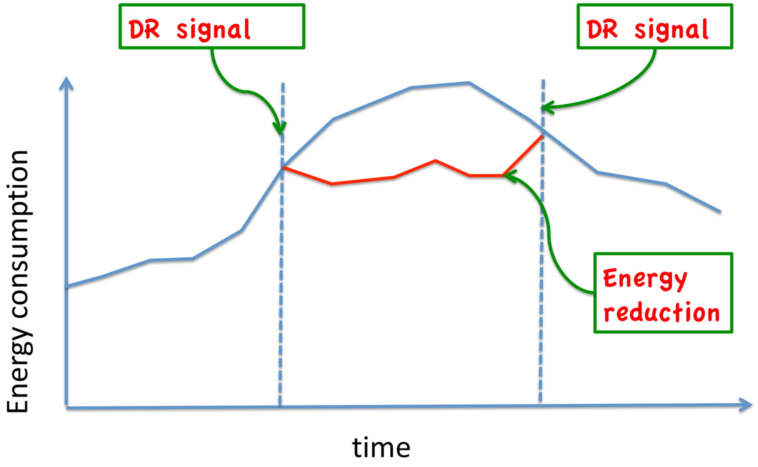
Supercomputing is extraordinarily power-hungry, with many of the top systems measuring their peak demand in the megawatts due to powerful processors and their correspondingly powerful cooling systems. As a result, these systems are often also extraordinarily carbon-intensive – and efficiency measures are struggling to keep pace with, let alone make headway on the rapidly accelerating demands of modern supercomputers.

ettascale

Exascale challenge: Perform exascale computing power with a maximum of 20MWs of electricity consumption

Demand Response

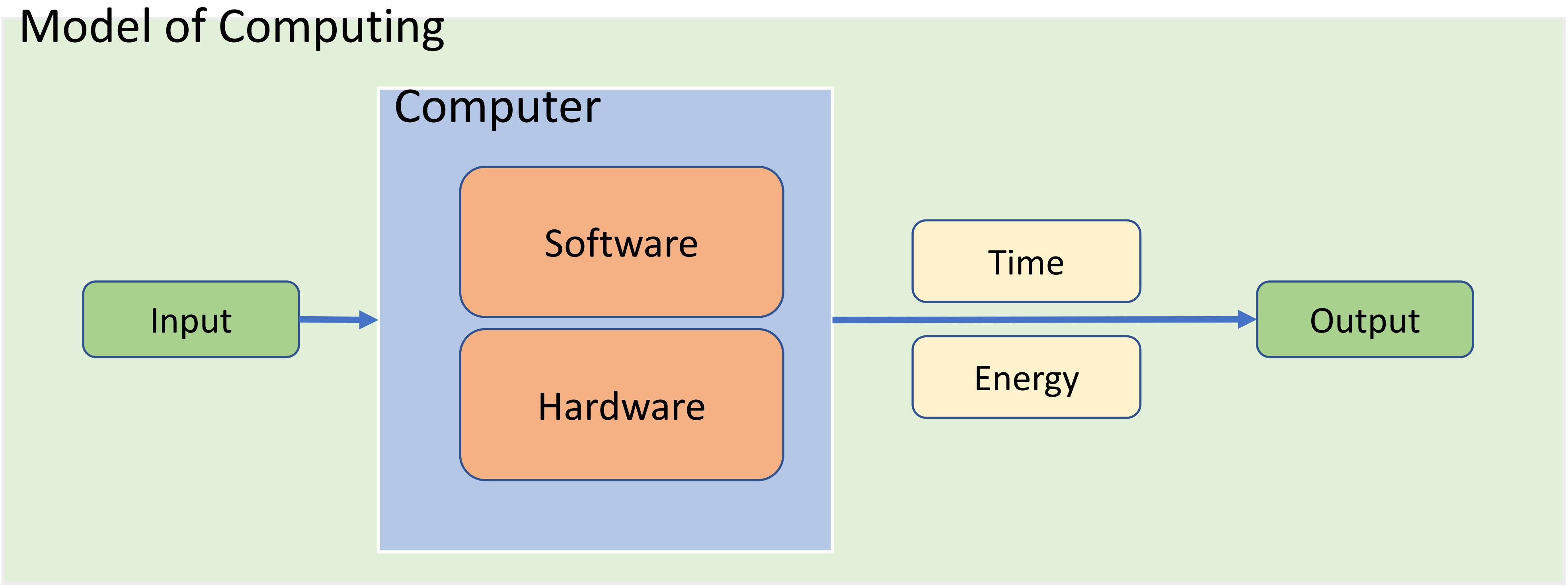
- Participants reduce energy consumption during
  - Emergency events
  - High electricity price period
- Emergency DR
  - Mandatory energy reduction to target level
- Economic DR
  - Voluntary participation based on economic incentives



Outline

- Background and motivation
- **Power and performance prediction modeling of HPC**
- Energy-efficient modeling of HPC
- Path forward

Performance Prediction of Computational Methods

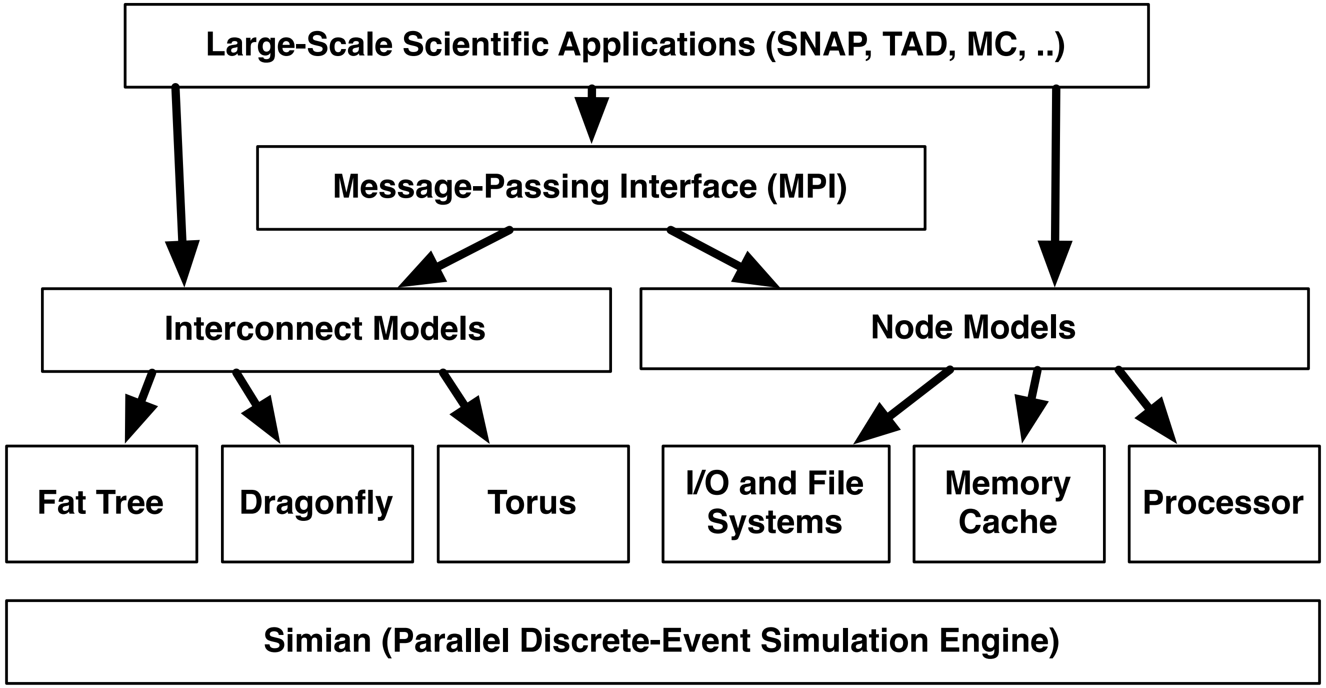


🟨 Predicted performance measured

🟧 Design spaces

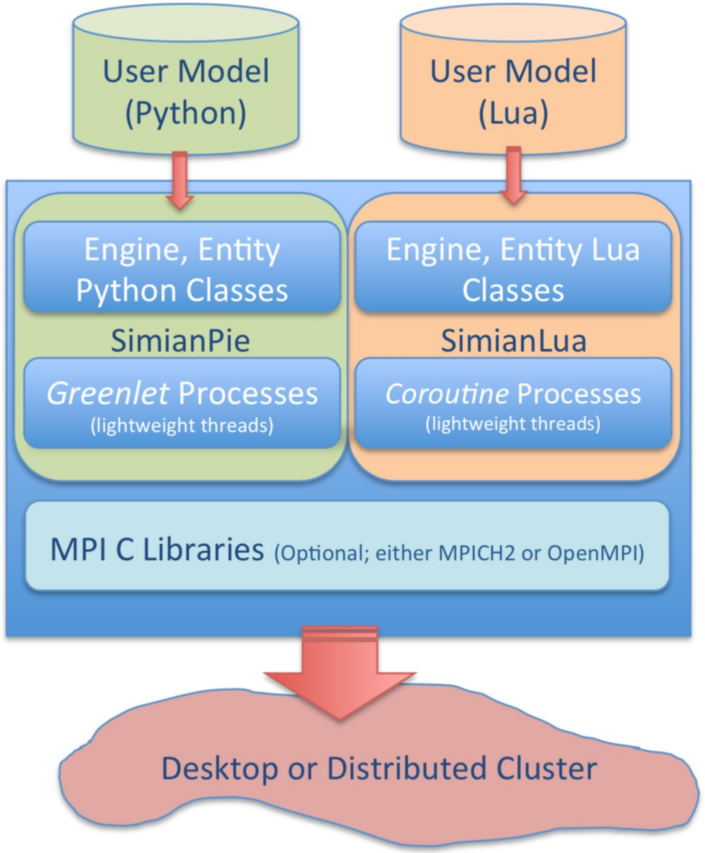
Performance Prediction Toolkit (PPT)

- Simian – parallel discrete event simulation engine
- Configurable hardware models
- Middleware models
- Application library


Large-Scale Scientific Applications (SNAP, TAD, MC, ..)
Message-Passing Interface (MPI)
Interconnect Models
Node Models
Fat Tree
Dragonfly
Torus
I/O and File Systems
Memory Cache
Processor
Simian (Parallel Discrete-Event Simulation Engine)

Simian

- Open source, general purpose parallel discrete-event library
- Independent implementation in two interpreted languages: Python and Lua, with optional C libraries (such as MPI)
- Minimalistic design: LOC=500 with 8 common methods
- Simulation code can be Just-In-Time (JIT) compiled to achieve very competitive event-rates, outperforming C++ implementation in some cases


User Model (Python)
User Model (Lua)
Engine, Entity Python Classes
SimianPie
Greenlet Processes (lightweight threads)
Engine, Entity Lua Classes
SimianLua
Coroutine Processes (lightweight threads)
MPI C Libraries (Optional; either MPICH2 or OpenMPI)
Desktop or Distributed Cluster

Interconnection network

- Interconnect is a critical component of extreme-scale HPC architectural design
- Interconnection network model is essential for performance evaluation studies
  - Need to be scalable, efficient, and accurate
- Common interconnect topologies
  - Torus (e.g., Cray’s Gemini)
  - Dragonfly (e.g., Cray’s Aries)
  - Fat-tree (e.g., Mellanox Infiniband)

Interconnection Network Simulators

- **BigSim (UIUC):** for performance prediction of large-scale parallel machines (with relatively simple interconnect models), implemented in Charm++ and MPI, shown to scale up to 64K ranks intiially

- **xSim (ORNL):** scale to 128M MPI ranks using PDES with lightweight threads, include various interconnect topologies (high-level models, e.g., network congestion omitted)

- **SST and SST Macro (SNL):** a comprehensive simulation framework, separate implementation, one intended with cycle-level accuracy and the other at coarser level for scale

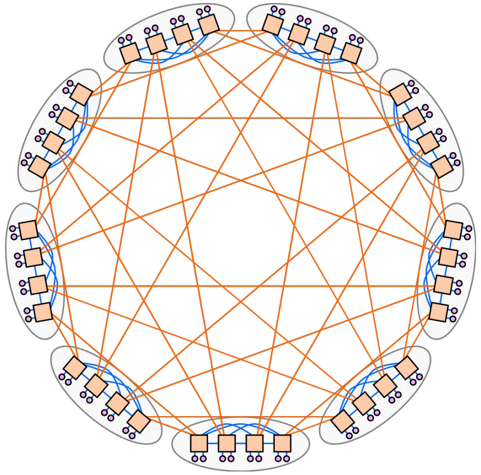
- **CODES (ANL):** contains interconnect models and storage systems, built on ROSS using reverse computation simulation that also scales well

Our Focus on **Rapid** Performance Prediction

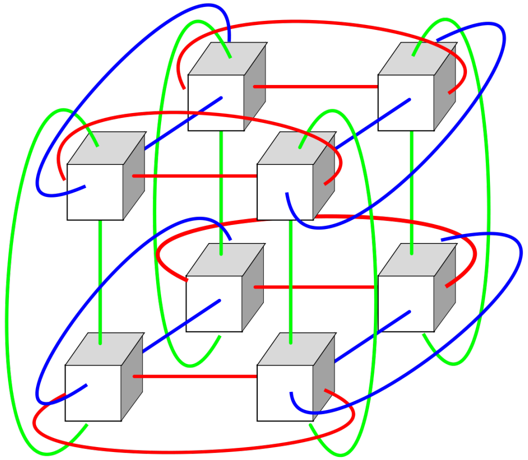
- Easy integration with selective models with varying abstraction
- Easy integration with physics applications
- Performance and scale
- Packet-level as opposed to phit-level
  - For performance and scale (speed advantage in several orders of magnitude, allow for full scale models, sufficient accuracy)
- Emphasis on production systems
  - Cielo, Darter, Edison, Hopper, Mira, Sequoia, Stampede, Titan, Vulcan, ...
- Seamlessly integrated with MPI
  - Implementation of all MPI common functions

Interconnect Models

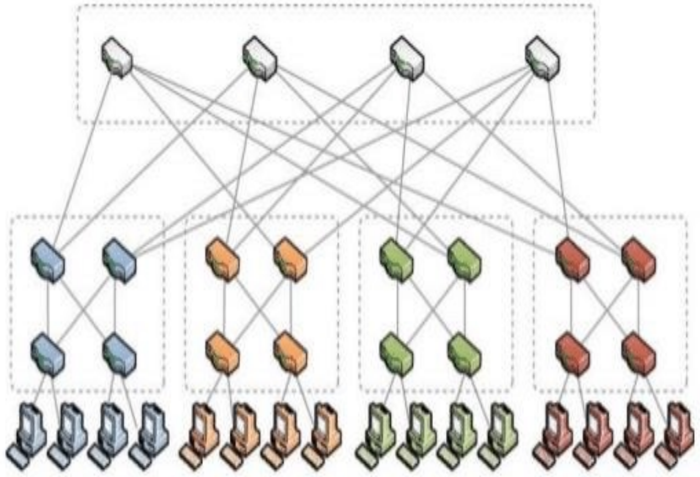
Three interconnect topologies



Dragonfly



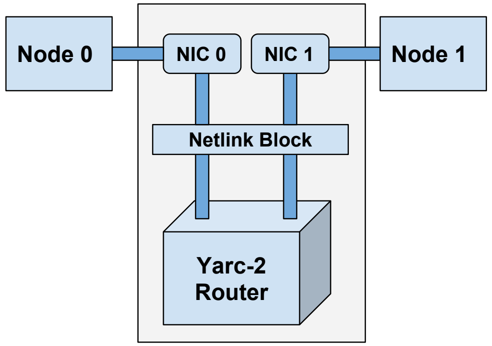
Torus

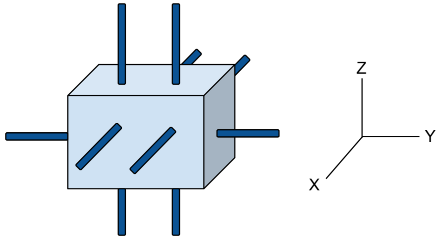


Fat-tree

Cray’s Gemini Interconnect

- 3D torus direct topology
- Each building block:
  - 2 compute nodes
  - 10 torus connections
    - ±X*2, ±Y, ±Z*2
- Routing
  - Adaptive dimension-order routing





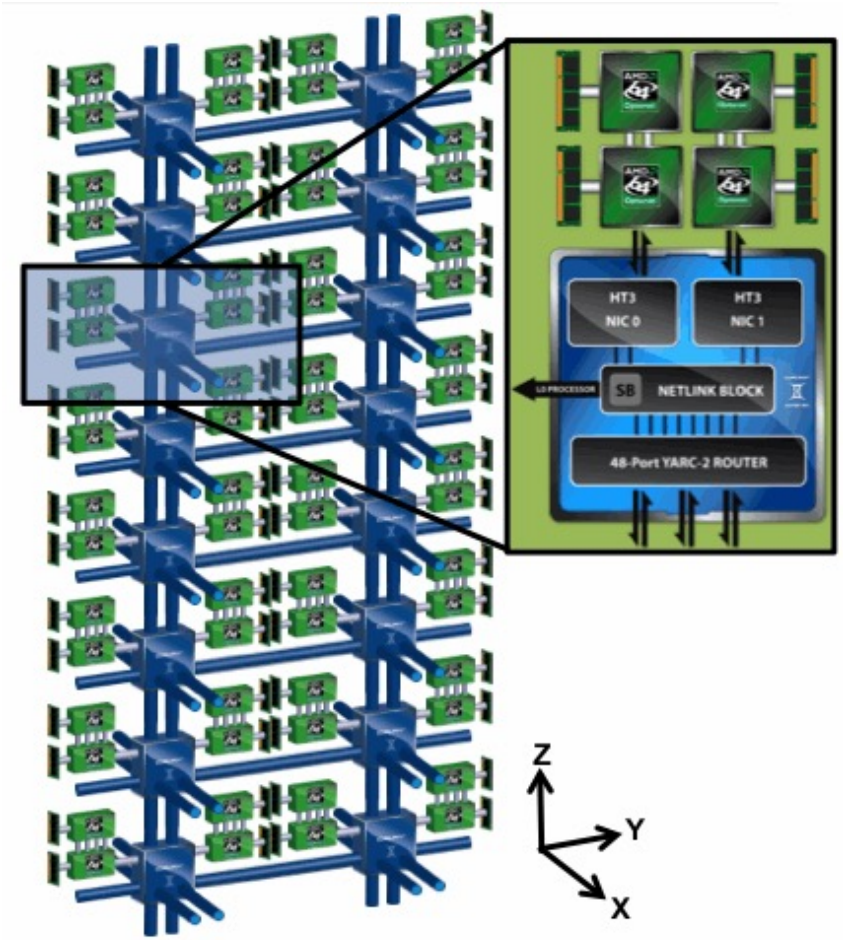
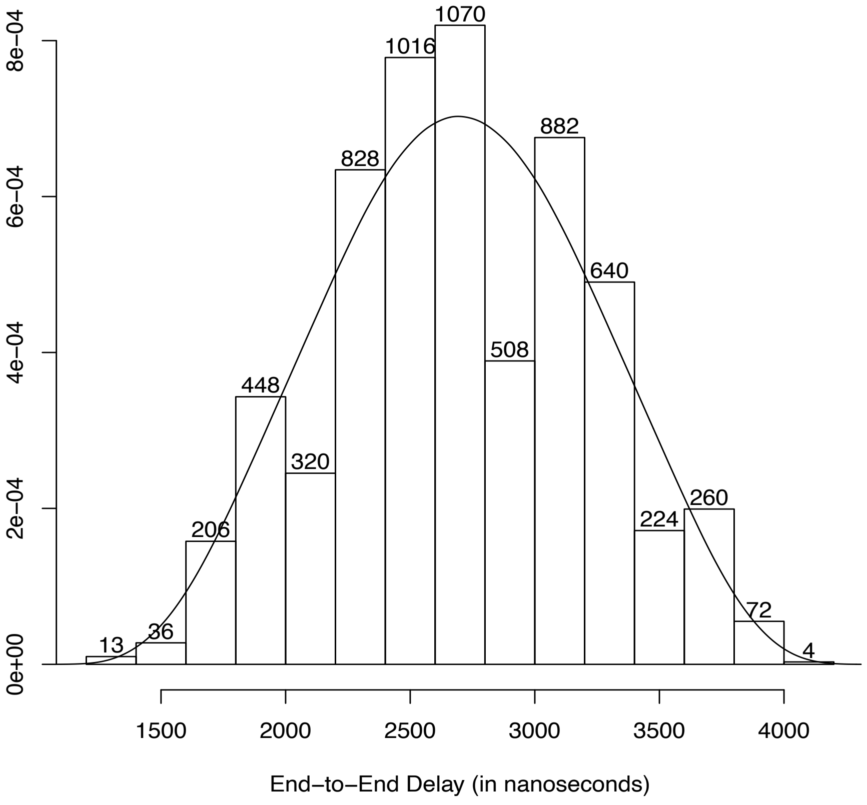
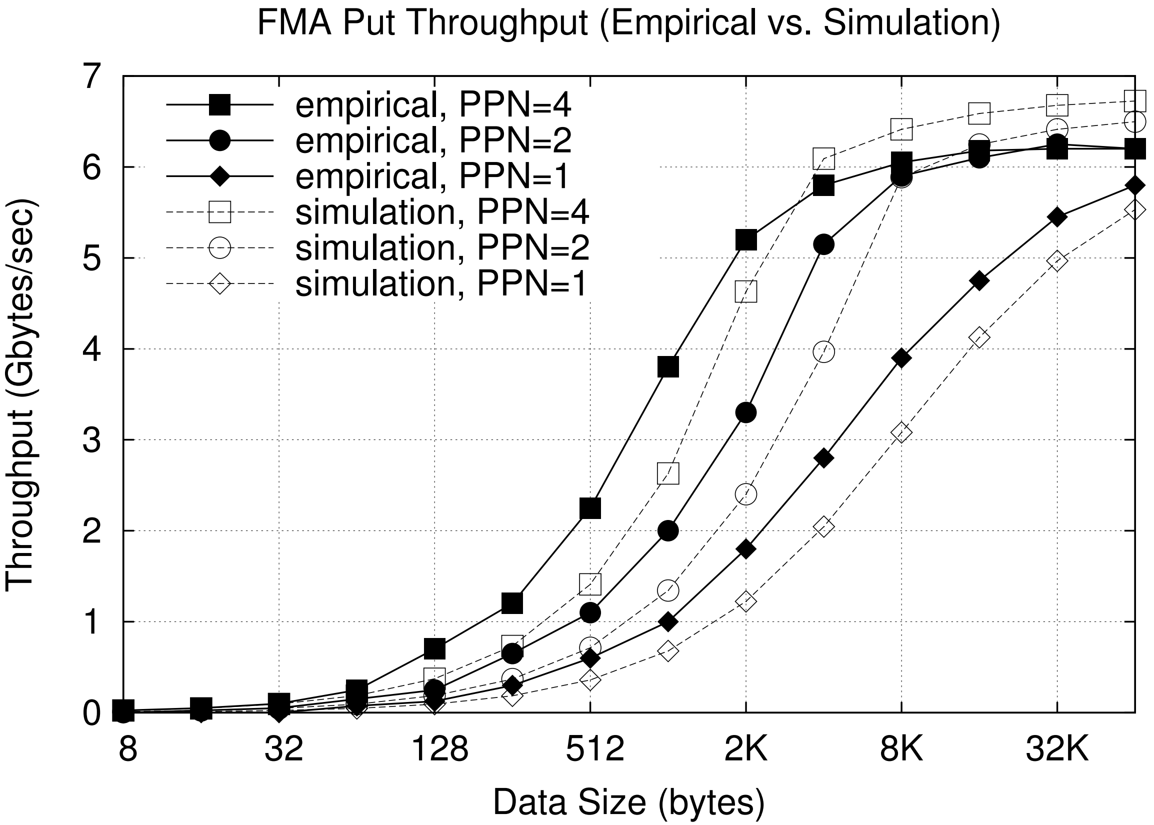


Image courtesy of Cray, Inc.

Gemini Validation

Compared against empirical results from Hopper @NERSC

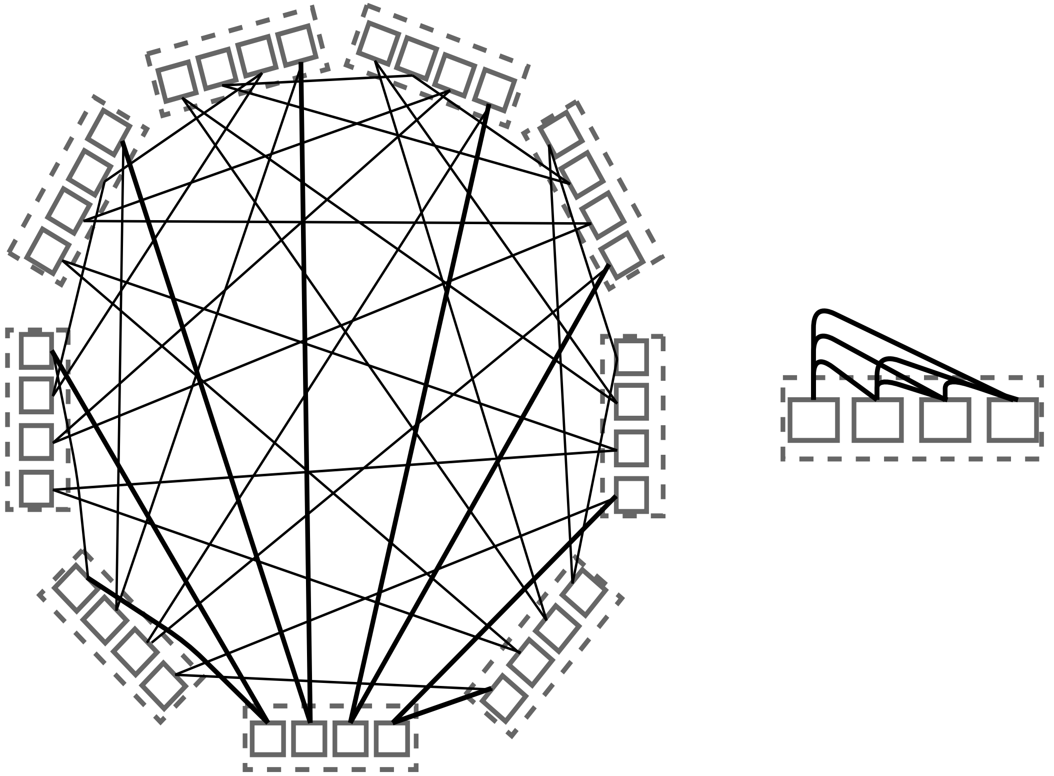




Inter-node latency: 1.27µs between nearest nodes
3.88µs between nearest nodes

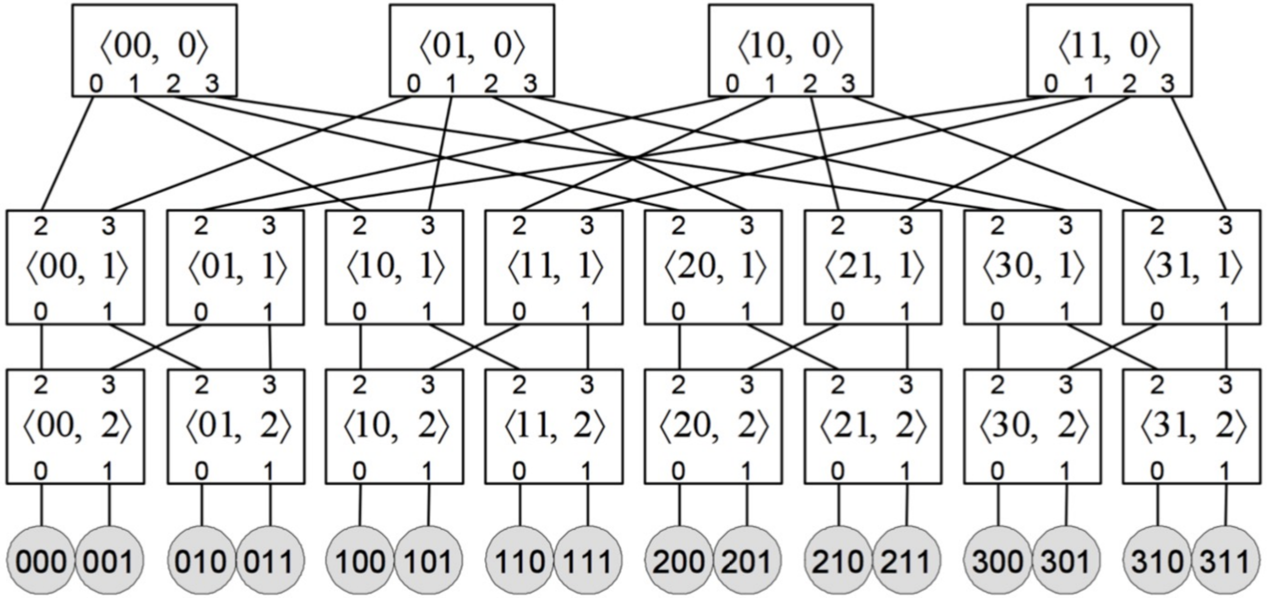
Cray’s Aries Interconnect

- Dragonfly: A cost-efficient topology
  - Nodes grouped together (high-radix router)
  - Economical, optical signaling technologies for distance routing
- Connections
  - Local link (completely connected)
  - Global link (consecutive connected)
- Routing
  - Minimal routing (benign traffic pattern)
  - Valiant routing (adversarial traffic pattern)



Fat-Tree Infiniband FDR

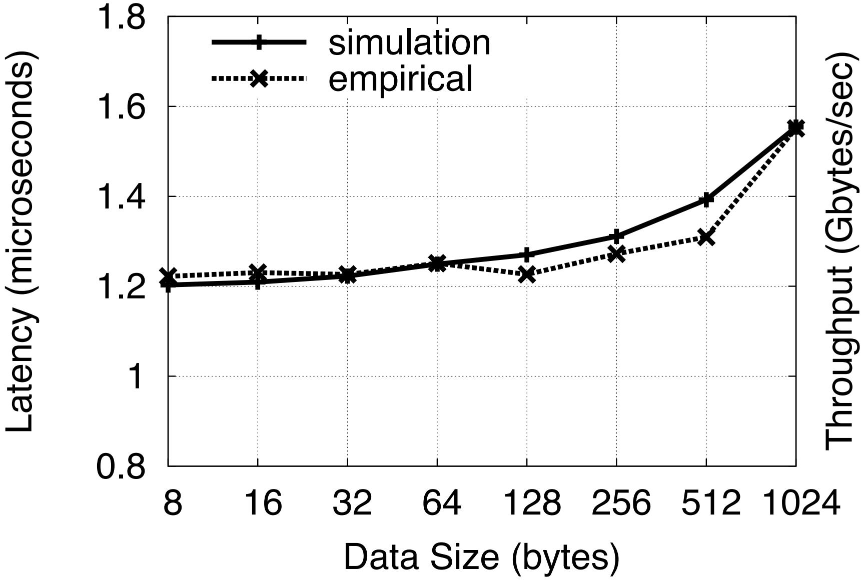
- An m-port n-tree
  - Height is (n+1)
  - $2(m/2)^n$ processing nodes
  - $(2n-1)(m/2)^{n-1}$ m-port switches
- Routing
  - Upward and downward phases
  - Valiant
- Examples: Stampede
  - 6400 nodes
  - 56Gbps Mellanox switches
  - 0.7 µs uplink and downlink latency

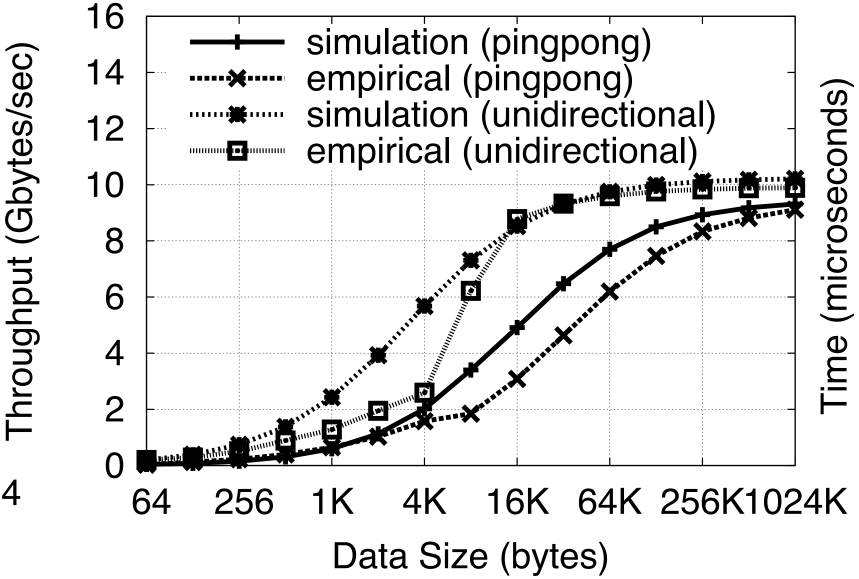


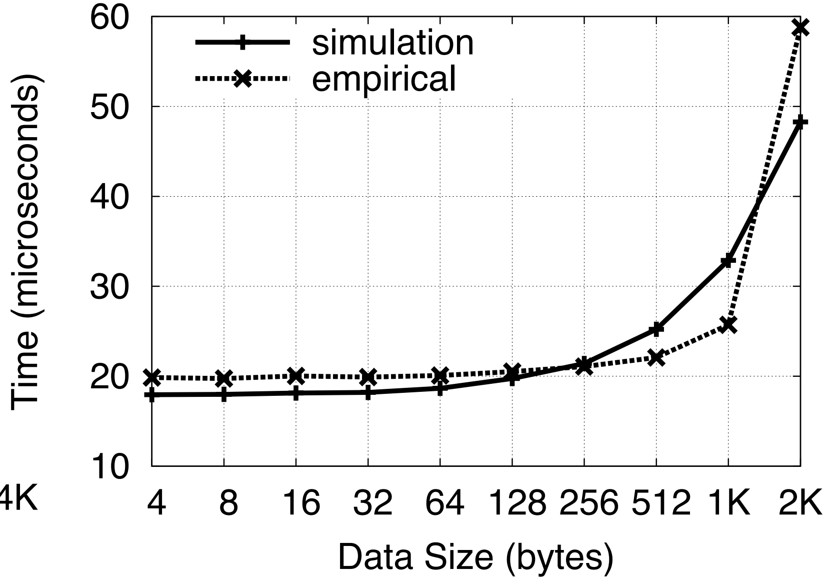
Dragonfly (Aries) and Fat-tree Validation

- Trinity@LANL
- 9436 nodes, uses Cray XC40 system
- Average end-to-end latency and throughput between nodes

- Darter@University of Tennessee
- 748 nodes, uses Cray XC30 system
- MPI_Allreduce time

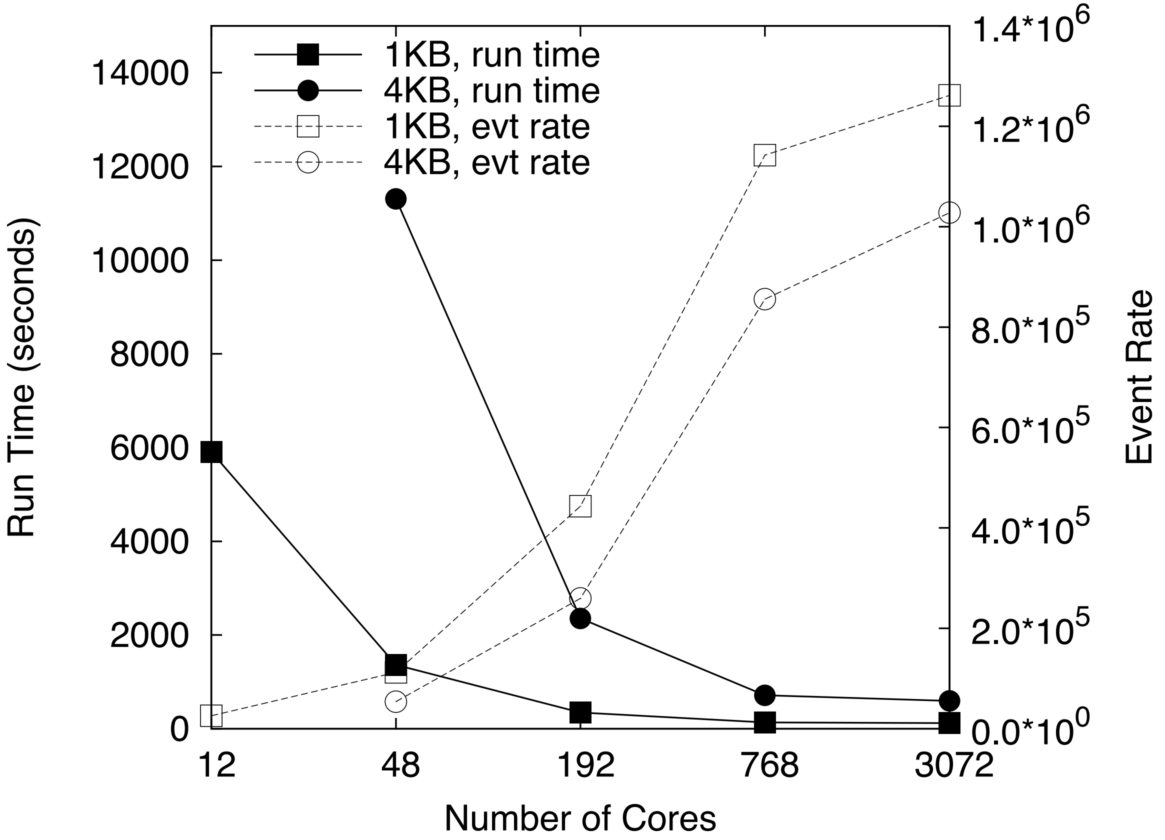






Parallel Performance

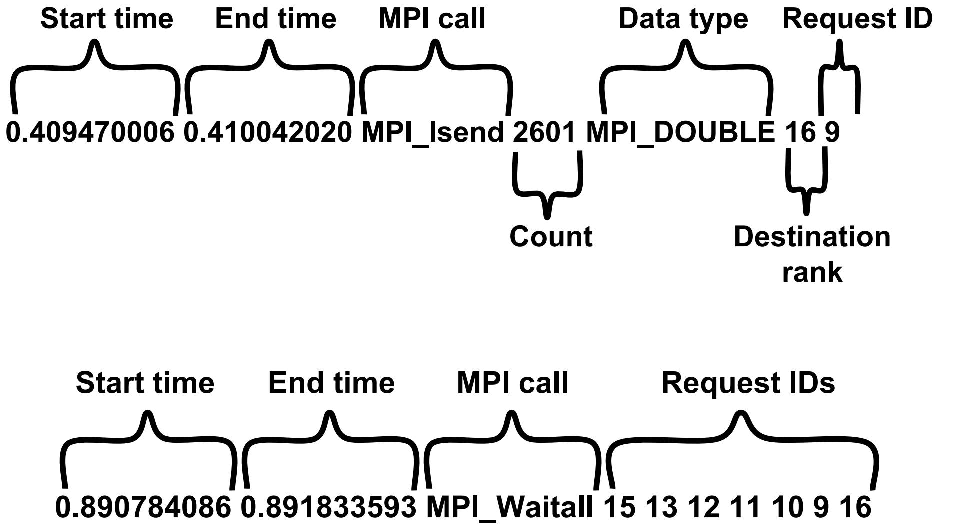
- A 1500-node cluster located at Los Alamos National Laboratory
- We varied number of compute nodes, from 1 (12 cores) to 256 nodes (3,072 cores)
- MPI_Allreduce, with different data size (1K or 4K bytes)



Three times event-rate of an optimized C++ simulator (MiniSSF)

Trace-Driven Simulation

- Use application communication traces for different DOE mini-apps (from NERSC)
- For this experiment, we use:
  - LULESH mini-app from ExMatEx
    - Approximates hydro-dynamic model and solves Sedov blast wave problem
  - 64 MPI processes
- Run trace for each MPI rank:
  - Start MPI call at exactly same time indicated in trace file
  - Store completion time of MPI call
  - Compare it with the completion time in trace file



Start time End time MPI call Data type Request ID

```
0.409470006 0.410042020 MPI_Isend 2601 MPI_DOUBLE 16 9
```

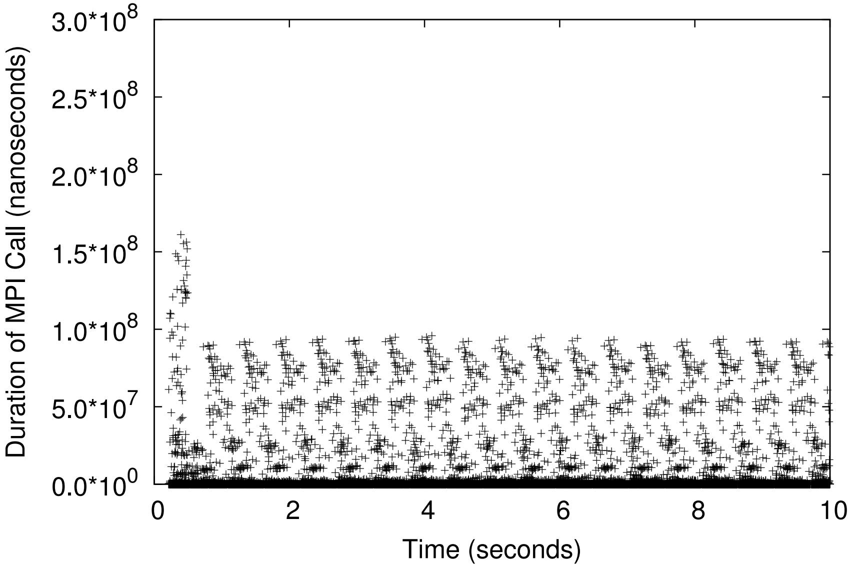
Count Destination rank

Start time End time MPI call Request IDs

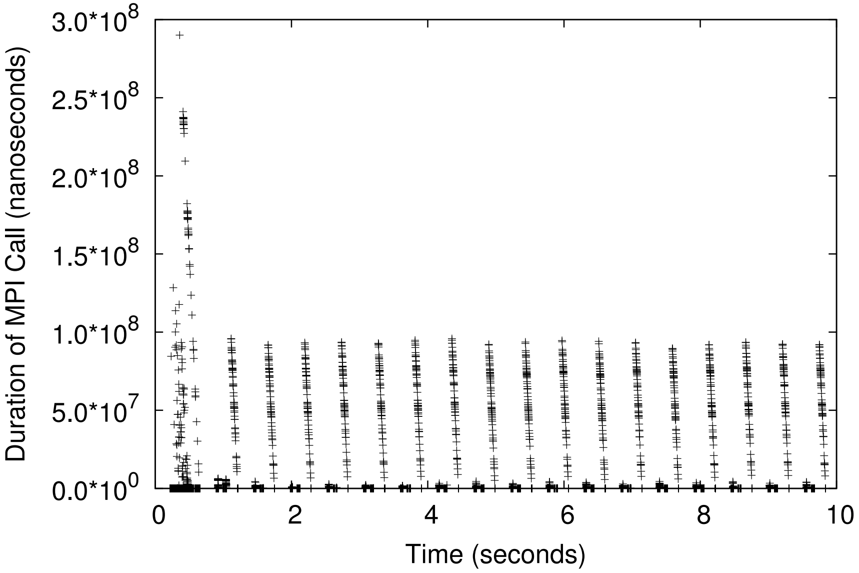
```
0.890784086 0.891833593 MPI_Waitall 15 13 12 11 10 9 16
```

Trace-Driven Simulation Results

- MPI calls:
  - MPI_Isend, MPI_Irecv, MPI_Wait (123,336 each)
  - MPI_Waitall (12,864)
  - MPI_Allreduce (6,336)
  - MPI_Barrier, MPI_Reduce (64 each)
- Simulation runtime 55 seconds



Trace output



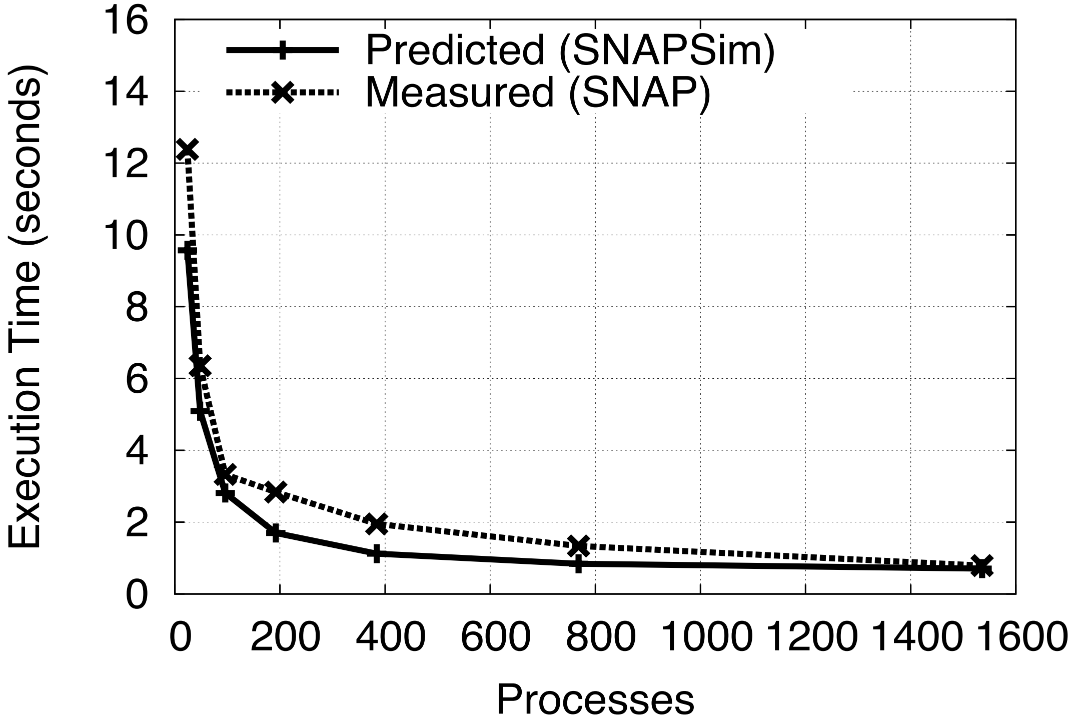
Simulation output

Strong Scaling Experiment

- SNAPSim
  - Stylized version of actual applications
- Use MPI to facilitate communication
- Use node model to do computation

Edison Strong Scaling Study

64 × 32 × 48 Spatial Mesh 384 Angles, 42 Energy Groups



Outline

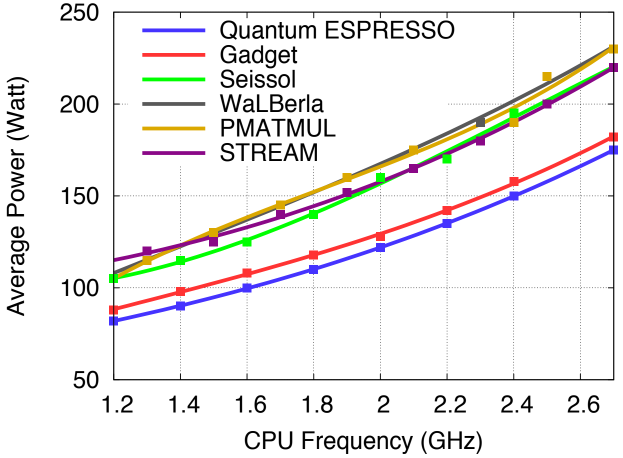
- Background and motivation
- Power and performance prediction modeling of HPC
- **Energy-efficient modeling of HPC**
- Path forward

Emergency DR Model

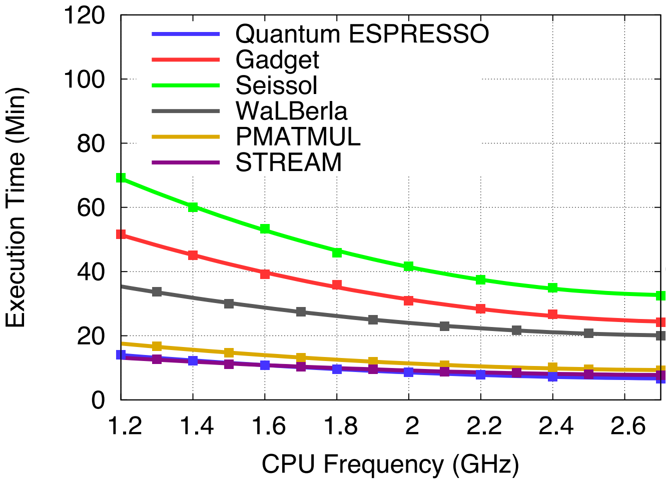
- Power/performance prediction model
  - Empirical data
  - Polynomial regression
- Demand response job scheduling
  - FCFS with possible job eviction
- Resource provisioning
  - Dynamic voltage frequency scaling (DVFS)

Power/Performance Prediction Model

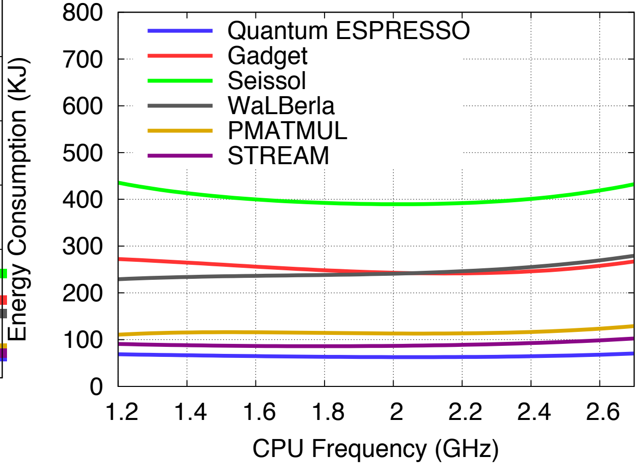
Apply regression on power and execution time



$$p(j,f) = a_j + b_j \cdot f + c_j \cdot f^2 + d_j \cdot f^3$$



$$t(j,f) = \alpha_j + \beta_j \cdot f + \gamma_j \cdot f^2$$



$$e(j,f) = n_j \cdot p(j,f) \cdot t(j,f)$$

Job Scheduling

● **During normal operation**

- Traditional job scheduling (FCFS)
- Optimized for best performance (max frequency)

● **During demand response period**

- Run jobs at optimal frequency
- May terminate some jobs to allocate reduced power limit

Resource Allocation

● **During demand response period**

- Reduce energy consumption (how?)

Exploit DVFS for DR participation

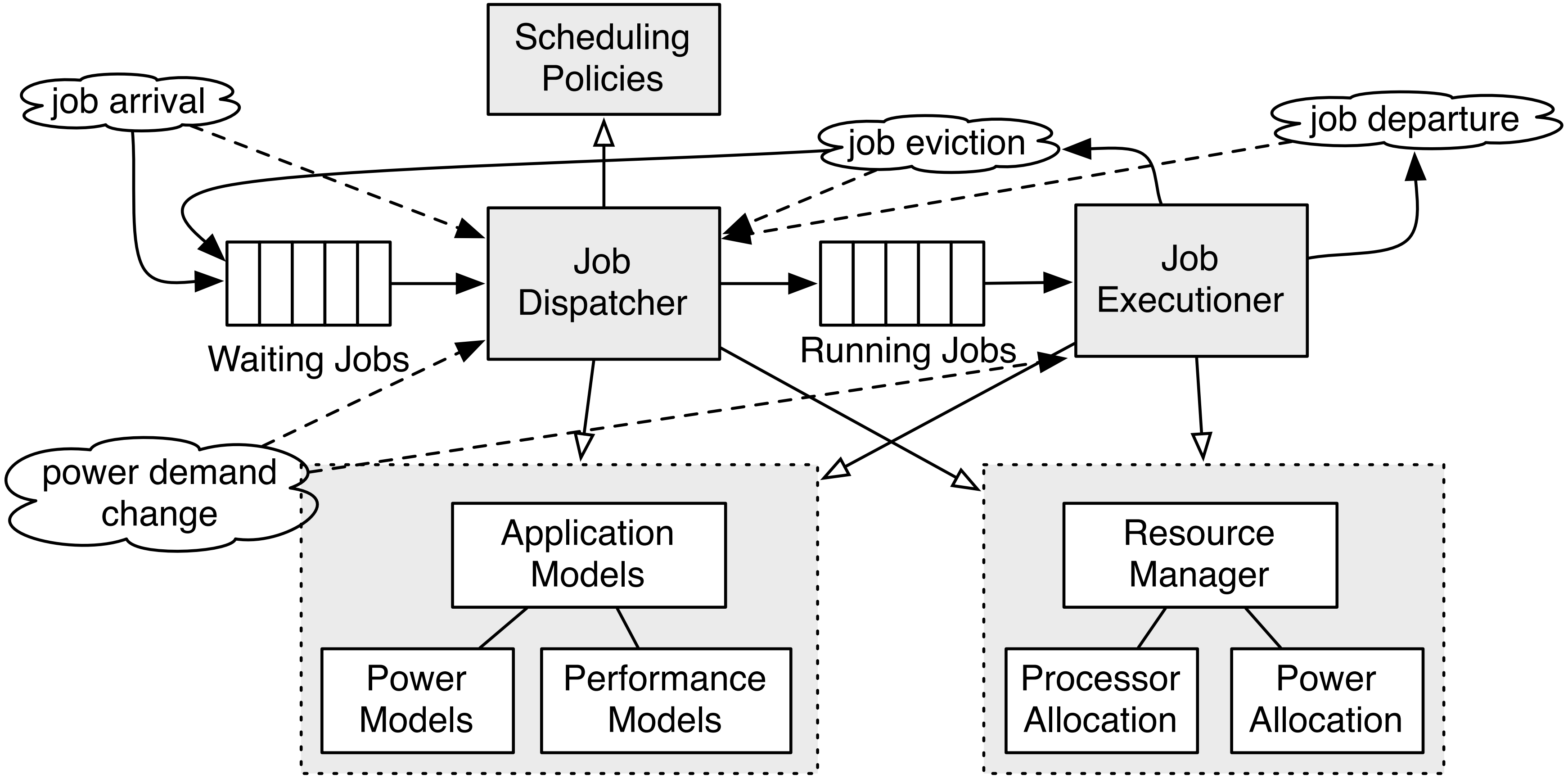
$$\text{Minimize: } \sum_{j \in R} e_R(j, f_j)$$

$$\text{subject to} \quad f_{min} \leq f_j \leq f_{max}$$

$$p_{run} = \sum_{j \in R} p(j, f_j) \leq \hat{p}$$

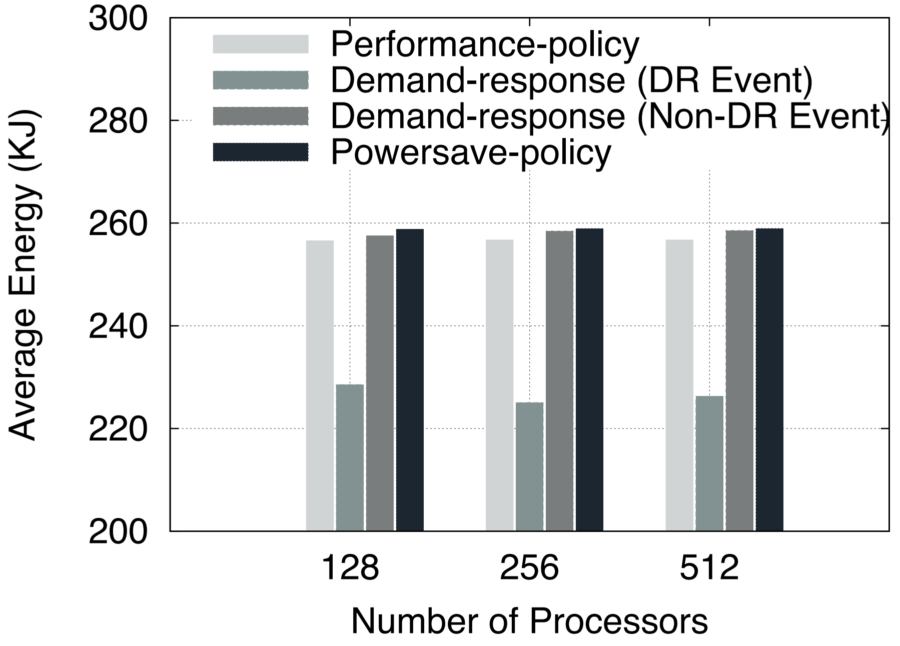
$$\text{where, } \; e_R(j, f_j) = (1 - \alpha_j) \cdot n_j \cdot p(j, f_j) \cdot t(j, f_j)$$

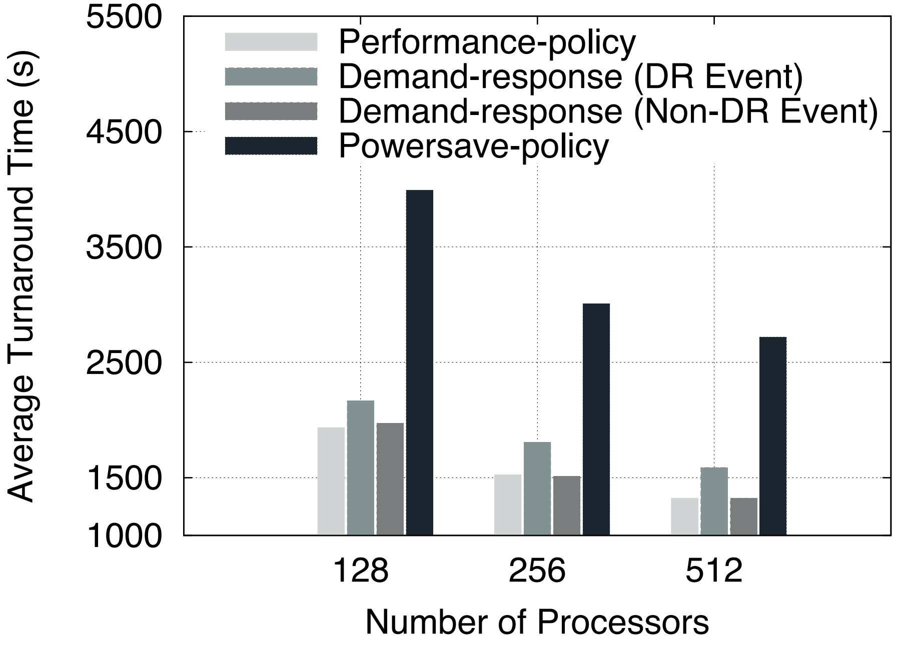
Scheduling Simulator



Model Evaluation

Vary system size: 128, 256 and 512 processors





Reduced energy consumption at moderate increase in turnaround time

Economic Demand Response

What is economic demand response?

- Voluntary participation based on economic incentives

How to incentivize HPC users?

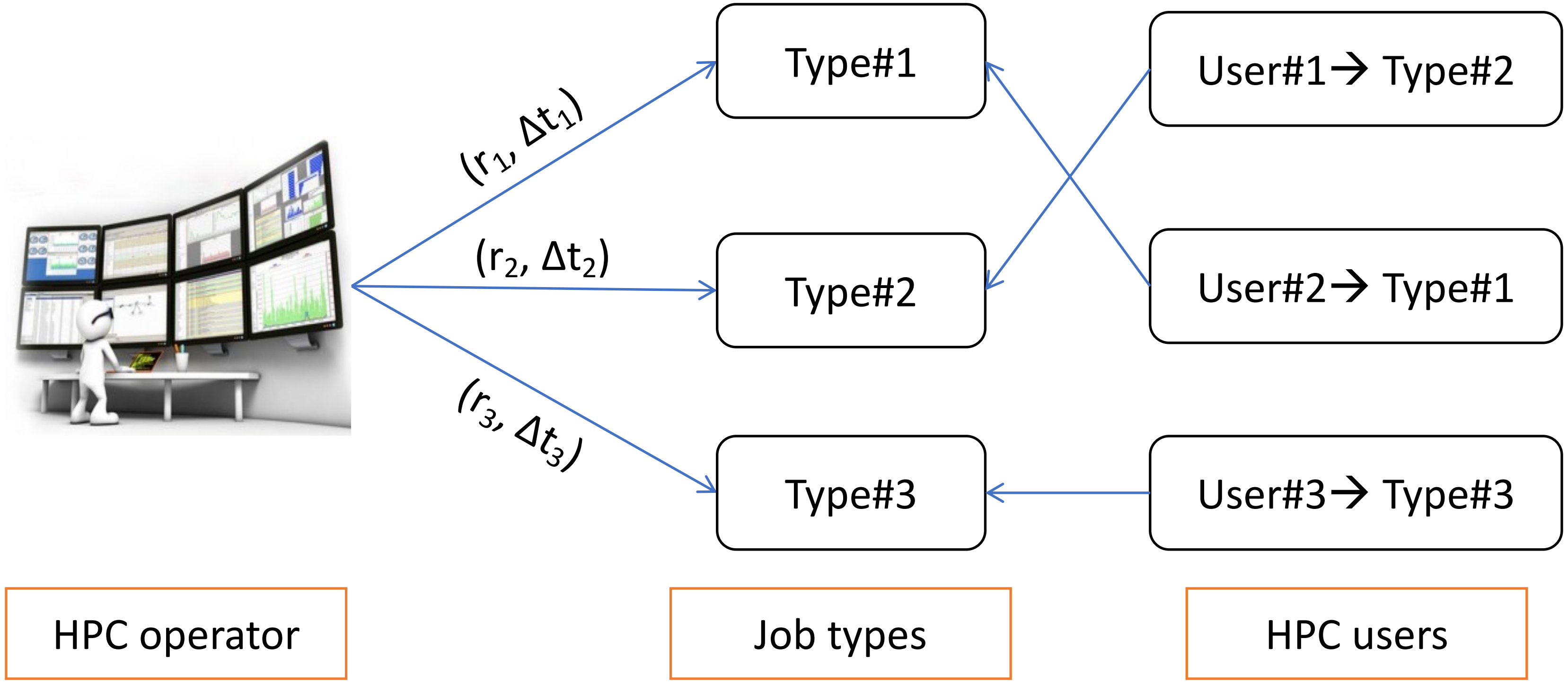
- Participation may introduce execution delays
- Need a proper rewarding mechanism

HPC users rewarded based on contract theory model

Contract Theory

- A formal (economic) study to develop contracts between parties
  - **Principal:** who offers the contracts (HPC operator)
  - **Agents:** who are offered the contracts (HPC users)
- Widely used in theory and practice
  - Economics (e.g., managerial compensation)
  - Communication (e.g., cellular network)

An Example



$(r_1, \Delta t_1)$

$(r_2, \Delta t_2)$

$(r_3, \Delta t_3)$

Type#1

Type#2

Type#3

User#1 → Type#2

User#2 → Type#1

User#3 → Type#3

HPC operator

Job types

HPC users

User’s utility maximized when selects own type’s contract

Resource Allocation Model

Maximize: $$\sum_{i=1}^{N} m_i \cdot (\phi \cdot \gamma \cdot \Delta e_i - \ r_i)$$

subject to: $f_{min} \leq f_{\text{i}} \leq f_{max}$, IR, and IC constraints

Individual rationality (IR) constraint:

$$r_i - \theta_i \cdot c(\Delta t_i) \geq 0$$

Incentive compatibility (IC) constraint:

$$r_i - \theta_i \cdot c(\Delta t_i) \geq r_{i'} - \theta_i \cdot c(\Delta t_{i'})$$

Power Capping and Job Size

● Power capping

- A common property in modern processors
- Dynamic setting of power budget to processors

● Job size

- Number of nodes allocated to a job

Exploit power-capping and job size for DR participation

Evaluation

HPC application power and performance data collected at an HPC cluster

CORAL COLLABORATION
OAK RIDGE • ARGONNE • LIVERMORE

NASA (NAS Devision)

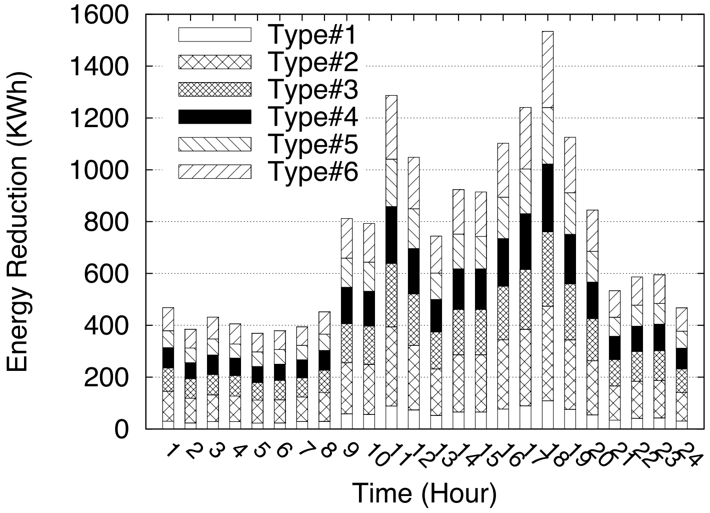
NAS Parallel Benchmarks

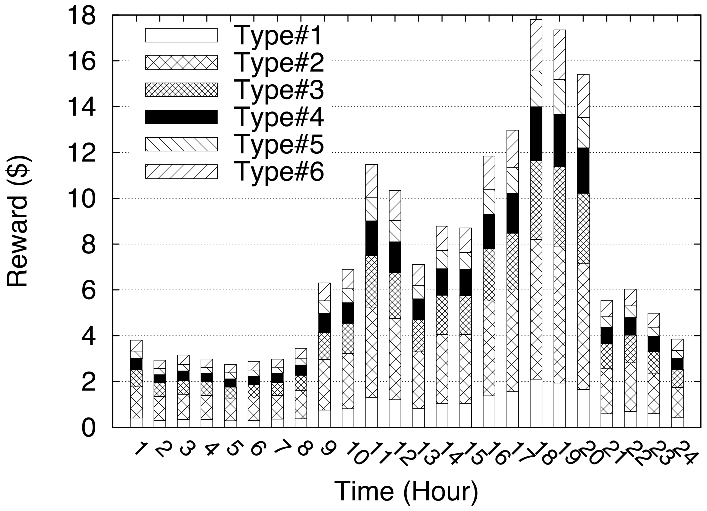
HPC CHALLENGE

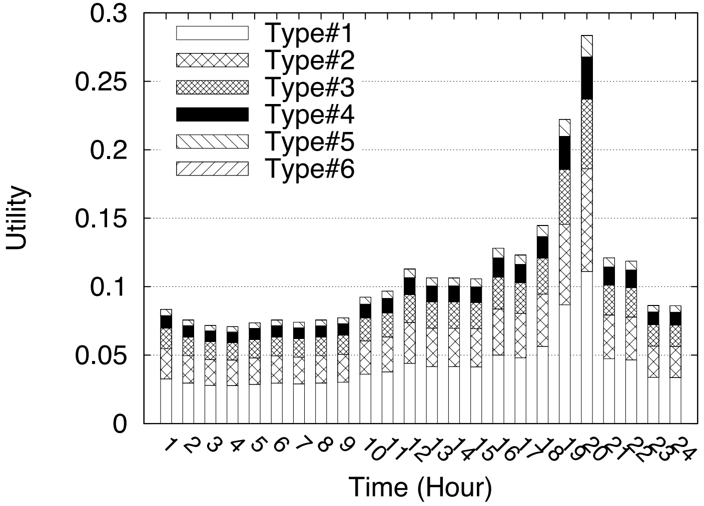
Chameleon

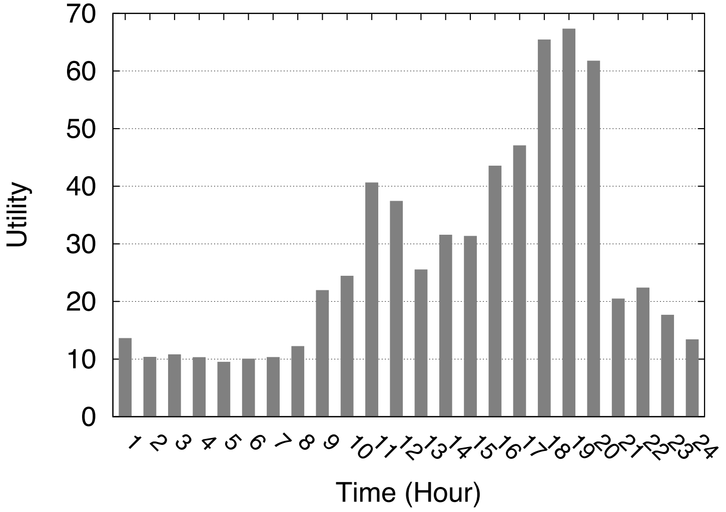
Argonne Joint Laboratory for System Evaluation

Contract Theory Mechanism Simulation

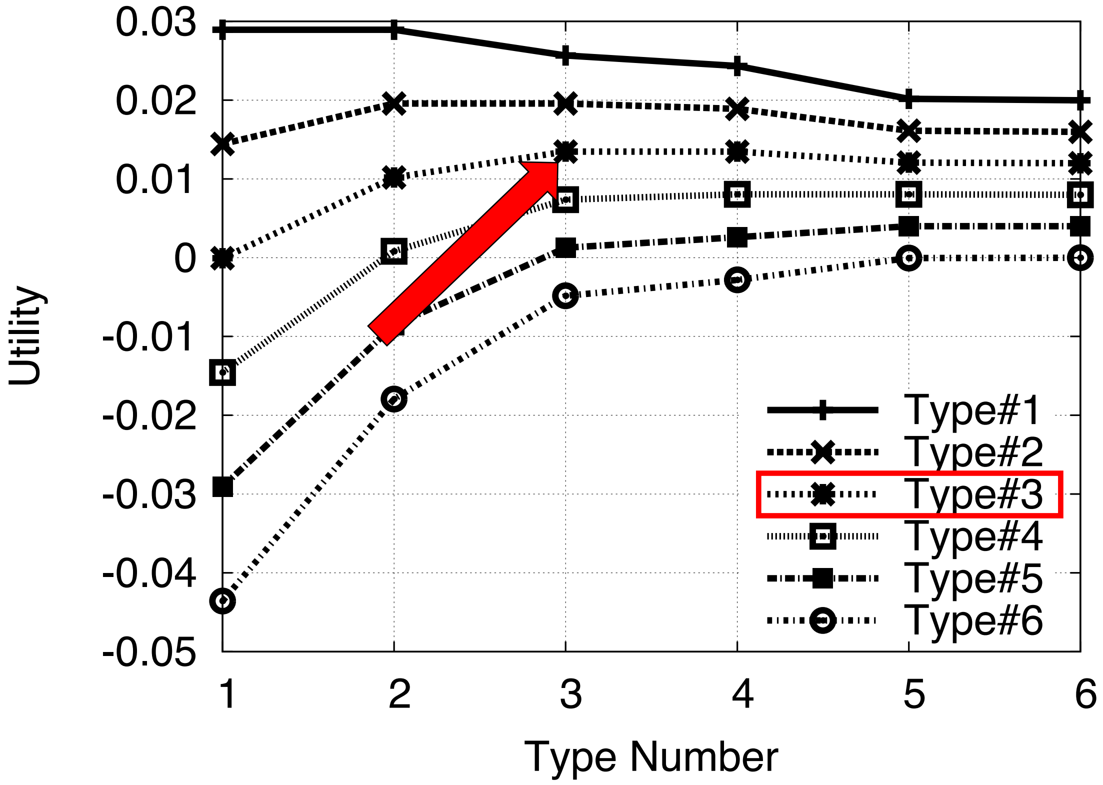








Contract Theory Mechanism Simulation (Contd.)

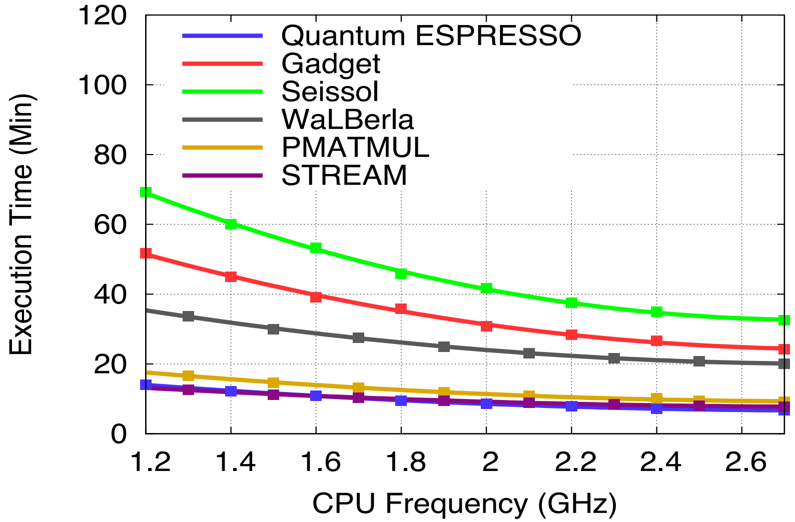


Each type of user achieves maximum utility, when users selects contract intended to its own type

Auction Mechanism Model

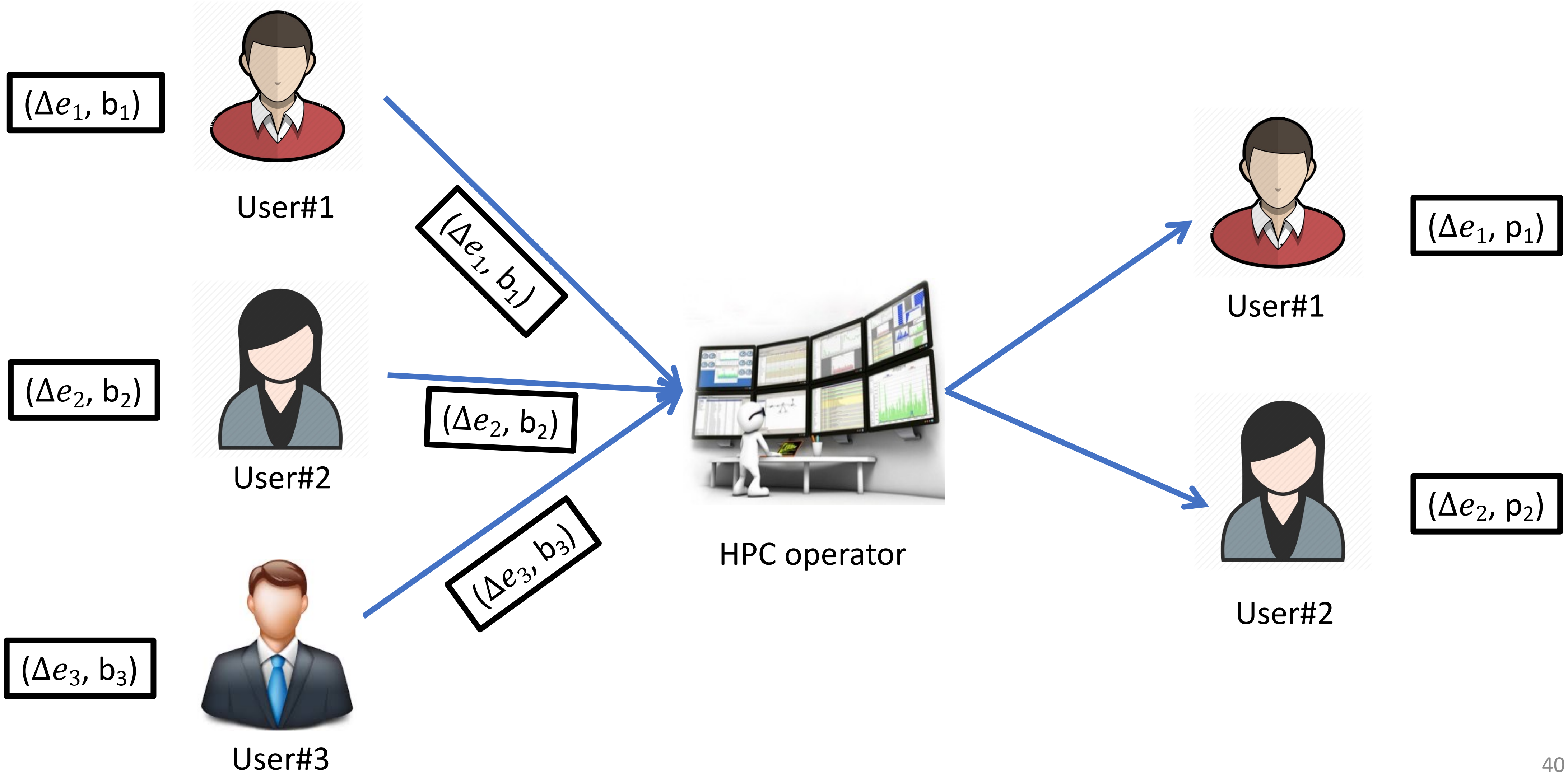
HPC users incur application performance loss

- How to ensure their participation?



HPC users rewarded based on a mechanism model (VCG-based auction)

An Example Scenario



$(\Delta e_1, \text{b}_1)$

User#1

$(\Delta e_1, \text{b}_1)$

$(\Delta e_2, \text{b}_2)$

User#2

$(\Delta e_2, \text{b}_2)$

$(\Delta e_3, \text{b}_3)$

$(\Delta e_3, \text{b}_3)$

User#3

HPC operator

User#1

$(\Delta e_1, \text{p}_1)$

User#2

$(\Delta e_2, \text{p}_2)$

Determining Bids and Winning Bidders

Inconvenience cost

$$\mathrm{c}_i = \beta \cdot \Delta t_i$$

- β converts time change to monetary value

Bid determined by user *i*

$$\mathrm{b}_i = \tau_i \cdot c_i$$

- τ_i is a truthfulness parameter

Operator solves the following optimization to determine the winning bidders

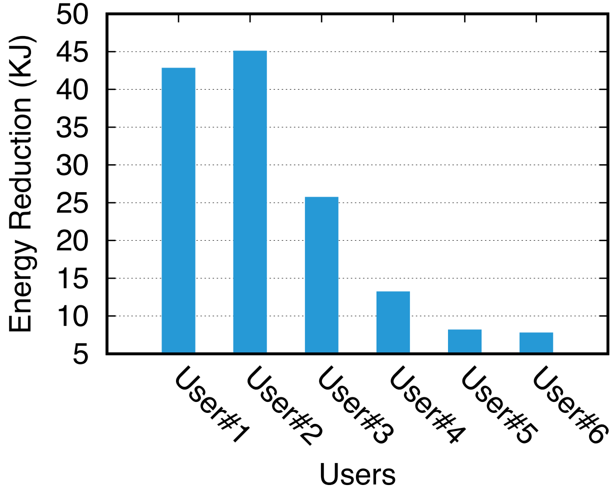
$$\text{Minimize: } \sum_{i=1}^{N} b_i$$

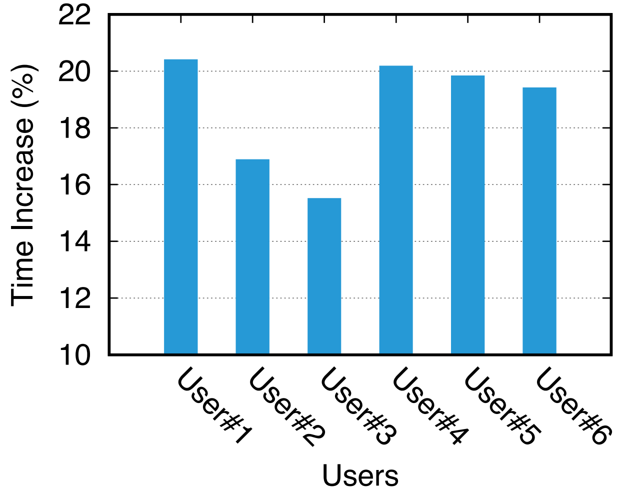
$$\text{subject to: } \sum_{i=1}^{N} \Delta e_i \geq e_{th}$$

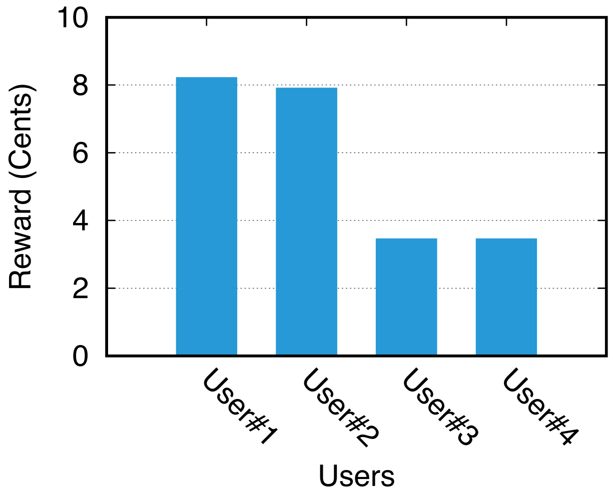
Payment of user-i*:*

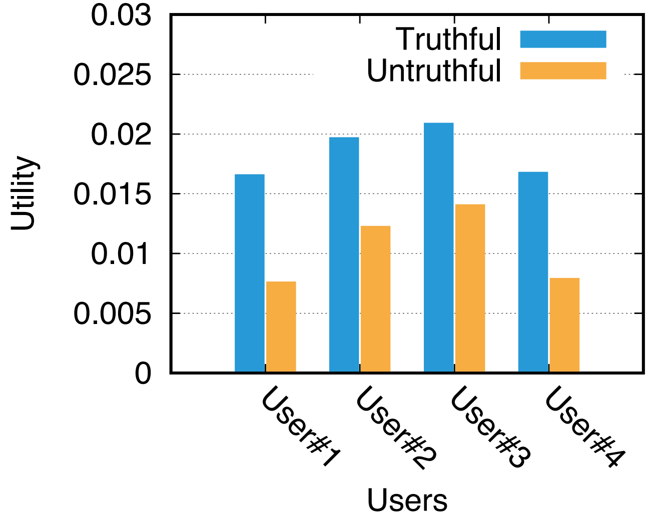
$$p_i = \ C^*_{B-b_i} \text{-} [C^*_B - b_i]$$

Auction Mechanism Simulation





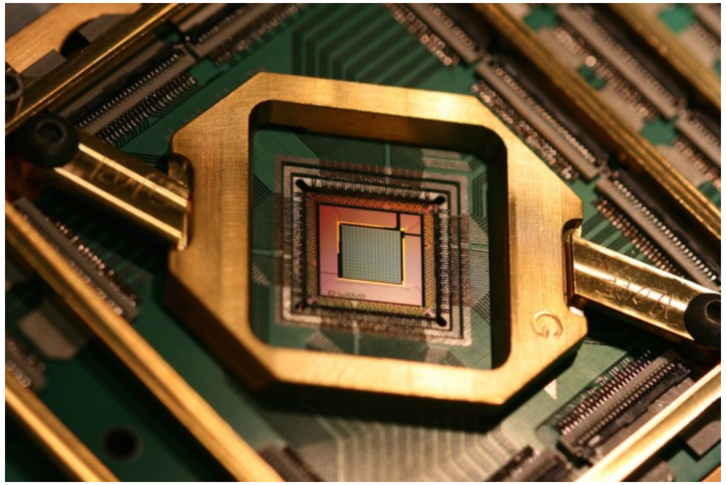




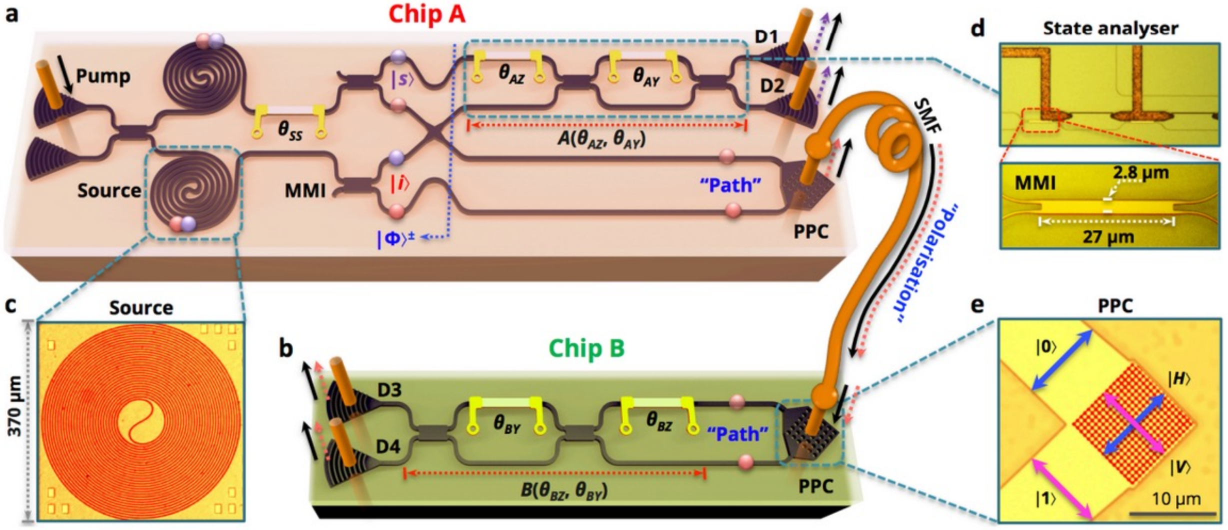
Outline

- Background and motivation
- Power and performance prediction modeling of HPC
- Energy-efficient modeling of HPC
- **Path forward**

Path Forward: System Simulation

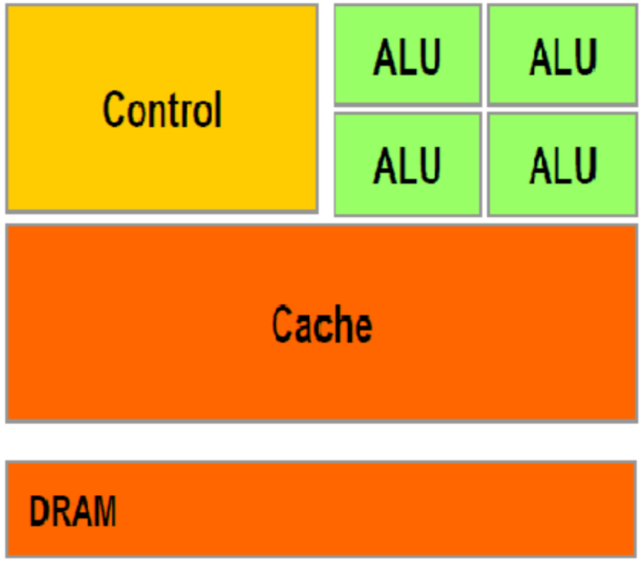


Quantum processor unit (qpu)

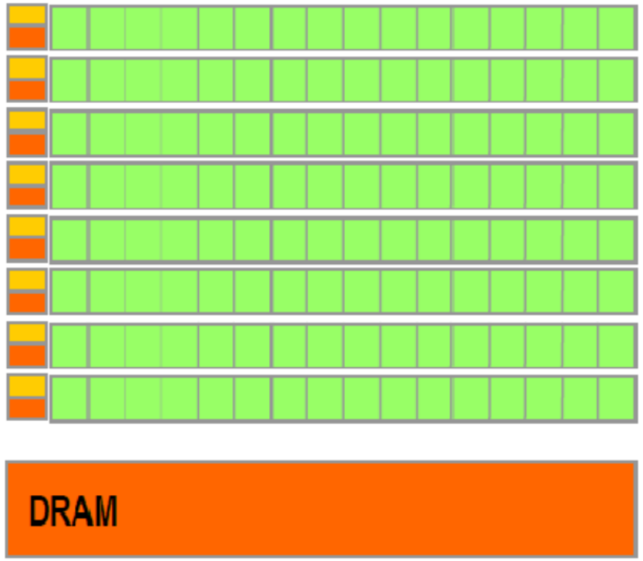


Quantum interconnect

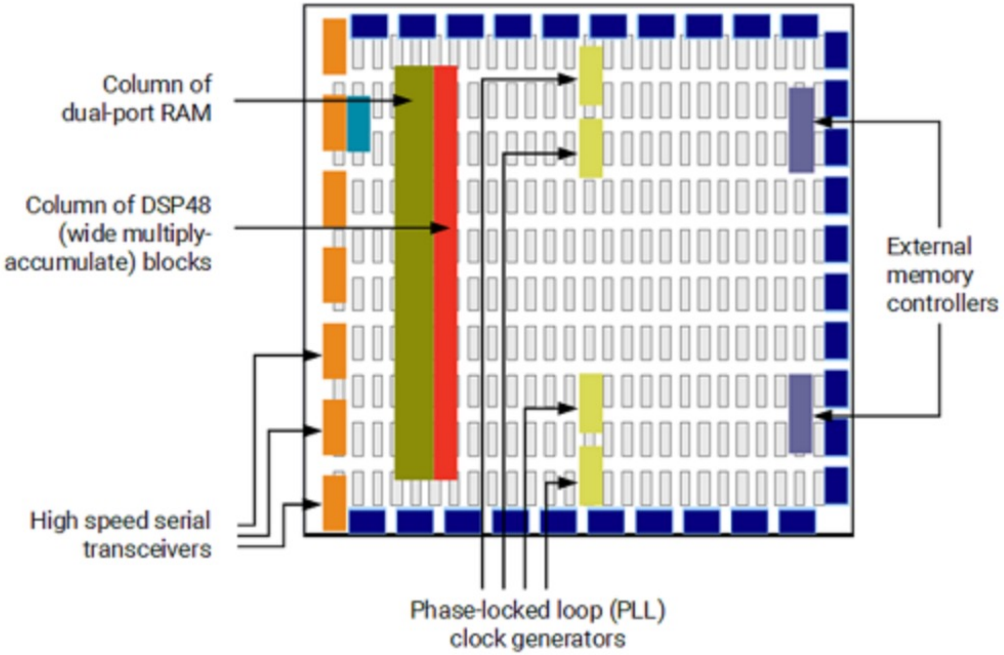
Path Forward: Energy-Efficient Computing



CPU



GPU


Column of dual-port RAM
Column of DSP48 (wide multiply-accumulate) blocks
High speed serial transceivers
Phase-locked loop (PLL) clock generators
External memory controllers

FPGA

Image source: Nvidia and Xilinx

Related Publications

- Kishwar Ahmed**,** Samia Tasnim, and Kazutomo Yoshii, ”Simulation of Auction Mechanism Model for Energy-Efficient High Performance Computing,” ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (**PADS**), 2020.
- Mohammad A. Islam, Kishwar Ahmed, Hong Xu, Nguyen Tran, Gang Quan, and Shaolei Ren. Exploiting Spatio-Temporal Diversity for Water Saving in Geo-Distributed Data Centers. **IEEE Transactions on Cloud Computing (TCC)**, 2018.
- Kishwar Ahmed, Jason Liu, and Kazutomo Yoshii. Enabling Demand Response for HPC Systems Through Power Capping and Node Scaling. Submitted to IEEE International Conferene on **High Performance Computing and Communications (HPCC)**, 2018
- Kishwar Ahmed**,** Jason Liu, and Xingfu Wu, “An Energy Efficient Demand-Response Model for High Performance Computing Systems,” IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (**MASCOTS**), 2017.
- Kishwar Ahmed, Mohammad Obaida, Jason Liu, Stephan Eidenbenz, Nandakishore Santhi, and Guillaume Chapuis. “An integrated interconnection network model for large-scale performance prediction,” ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (**PADS**), 2016.
- Kishwar Ahmed, Mohammad A. Islam, and Shaolei Ren. “A Contract Design Approach for Colocation Data Center Demand Response,” IEEE International Conference on Computer-Aided Design (**ICCAD**), 2015.

Thank You!
Questions?

Acknowledgements:

NSF

UNIVERSITY OF
South Carolina
Office of Research

Argonne
NATIONAL LABORATORY