SAND2020-1537PE

# The Exascale Computing Project: Hardware Evaluation for Interconnects

ECP Annual Meeting

**Scott Hemmert (SNL), Jeremiah Wilke (SNL), Rob Ross (ANL), Sudheer Chunduri (ANL), Taylor Groves (LBNL), Ian Karlin (LLNL)**

Houston, TX
February 2020

EXASCALE COMPUTING PROJECT

www.ExascaleProject.org

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

# Interconnect evaluation via simulation answers system design questions that are *difficult or impossible* to answer on an existing, live production system

- Test designs before full expense of procurement/implementation
  - Performance of new topologies or routing algorithms
  - Value of switch architectures provided by different vendors
- Test system configurations without interrupting production system
  - Reconfigure network routing tables or QoS
  - Placement or allocation strategies
- Controlled environment to isolate individual design parameters
  - Sometimes difficult to isolate exact causes of performance in real system
  - More easily control aspects like job placement

## The session today should focus on expressing capabilities to and collecting requirements from facilities and other interested customers

- Introduction to simulator capabilities
  - Done through brief descriptions of some previous milestones

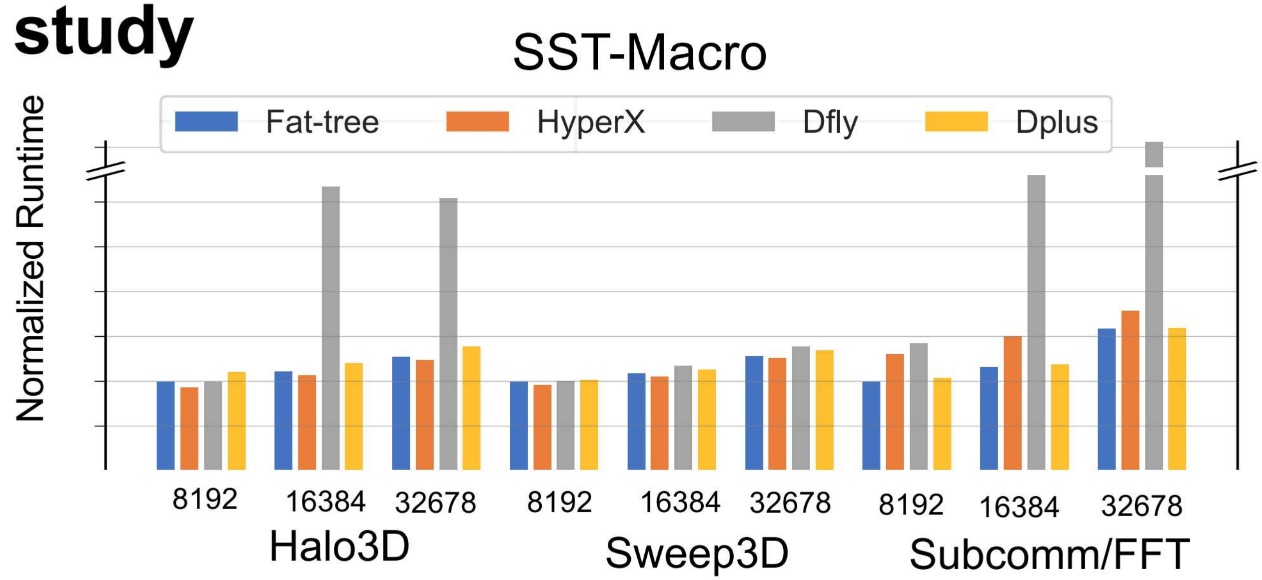
- Questions and insights from facilities representatives

- Discussion

# Previous milestones show range and type of questions being addressed

- **Milestone #1** (Q3 2018): Topology + routing design space survey
- **Milestone #2** (Q4 2018): Analysis and sensitivity to interconnect interference for A21
- Milestone #3 (Q2 2019): Detailed performance counter validation on production systems
- **Milestone #4** (Q4 2019): Ability of simple QoS strategies for alleviating multi-job interference

# Simulator Capabilities from Milestone #1: General topology and routing study

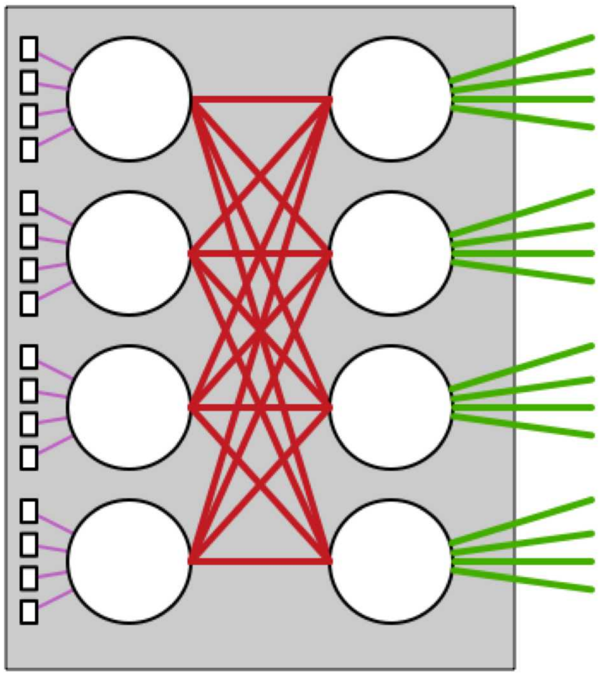



| Routing Algorithms | | | |
|---|---|---|---|
| Fat Tree | HyperX | Dragonfly | Dragonfly+ |
| Oblivious Adaptive (OA) | Progress Adaptive (PAR) | PAR | PAR |

# Rob Ross

# Simulator Capabilities from Milestone #2: Sensitivity of A21 interconnect to interference

- 8,448 node Dragonfly+ (a.k.a. Megafly) network
- Two service levels with bandwidth capping on fixed time interval
- SWMs to represent application patterns
  - Nekbone
  - LAMMPS
  - Nearest Neighbor (NN)
- Synthetic background traffic (e.g., uniform random)
- Two QoS scenarios
  - Priority to a single, latency-sensitive app
  - Priority to collective communication

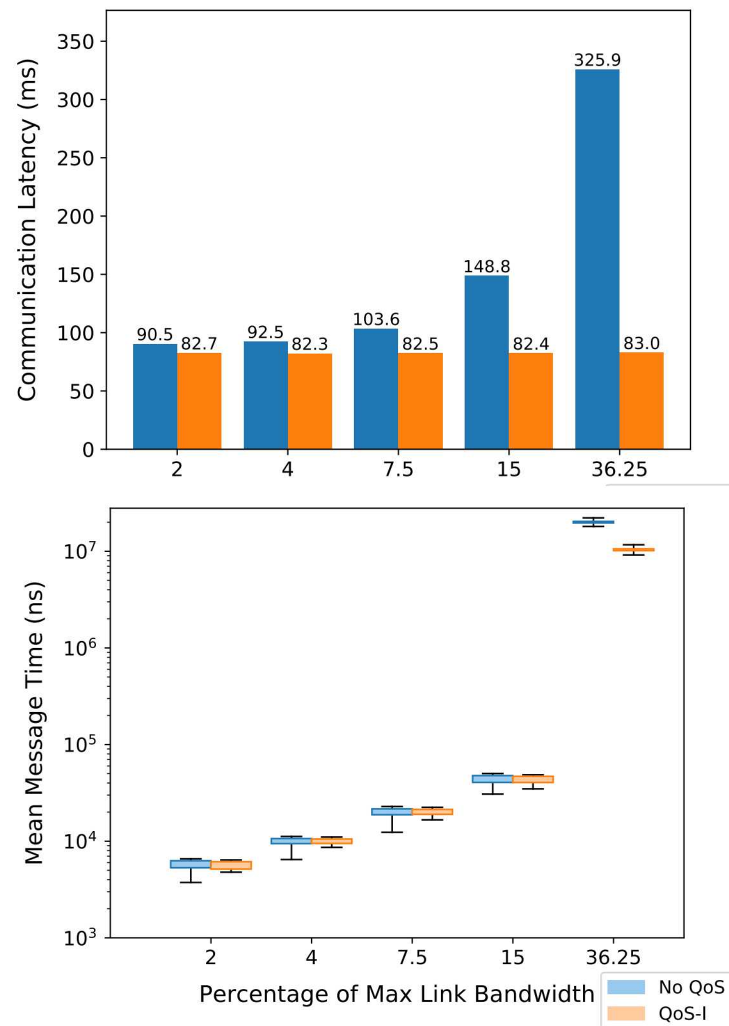


Dragonfly+ architecture: 2-level fat trees for local group connections, many options for spine connectivity.

# Prioritizing a select application (QoS-I)

- Prioritizing traffic from Nekbone (2,197 ranks)
  - Communication intensive and collective heavy
  - Given 70% of bandwidth as cap
- Uniform random background traffic on remaining ranks
  - Vary intensity as a percentage of link bandwidth

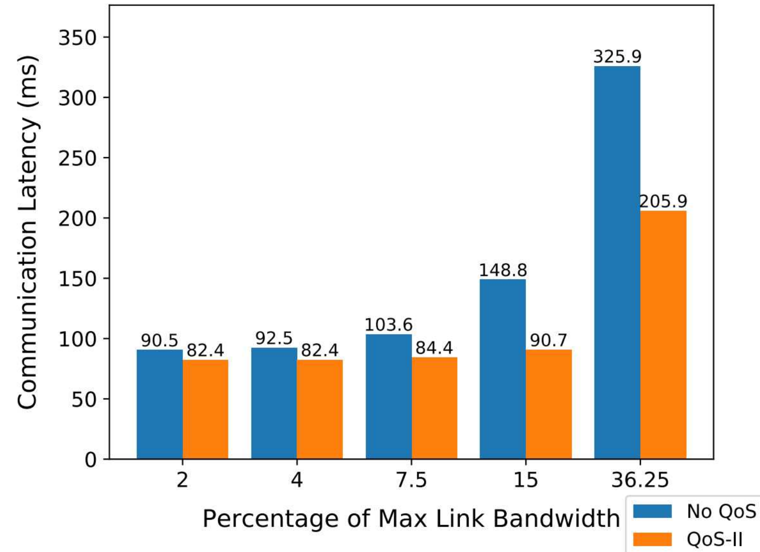
***Traffic differentiation with bandwidth shaping and prioritization can mitigate variability while causing minimal slowdown to background traffic.***

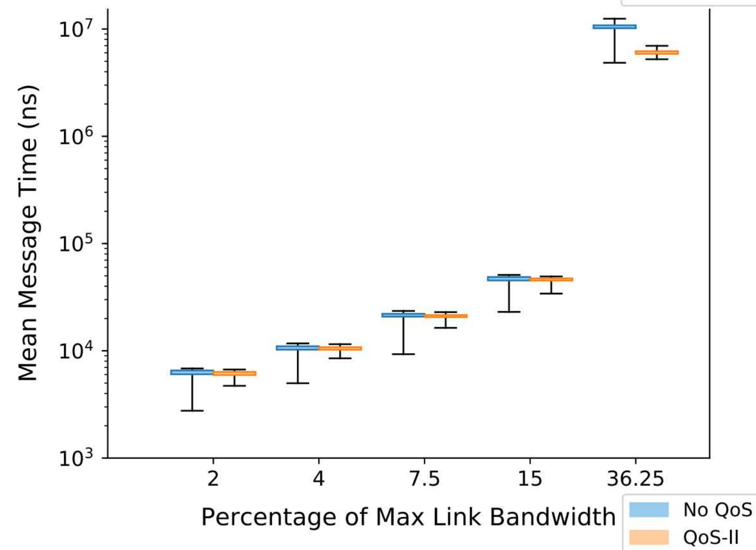


# Prioritizing latency-sensitive operations (QoS-II)

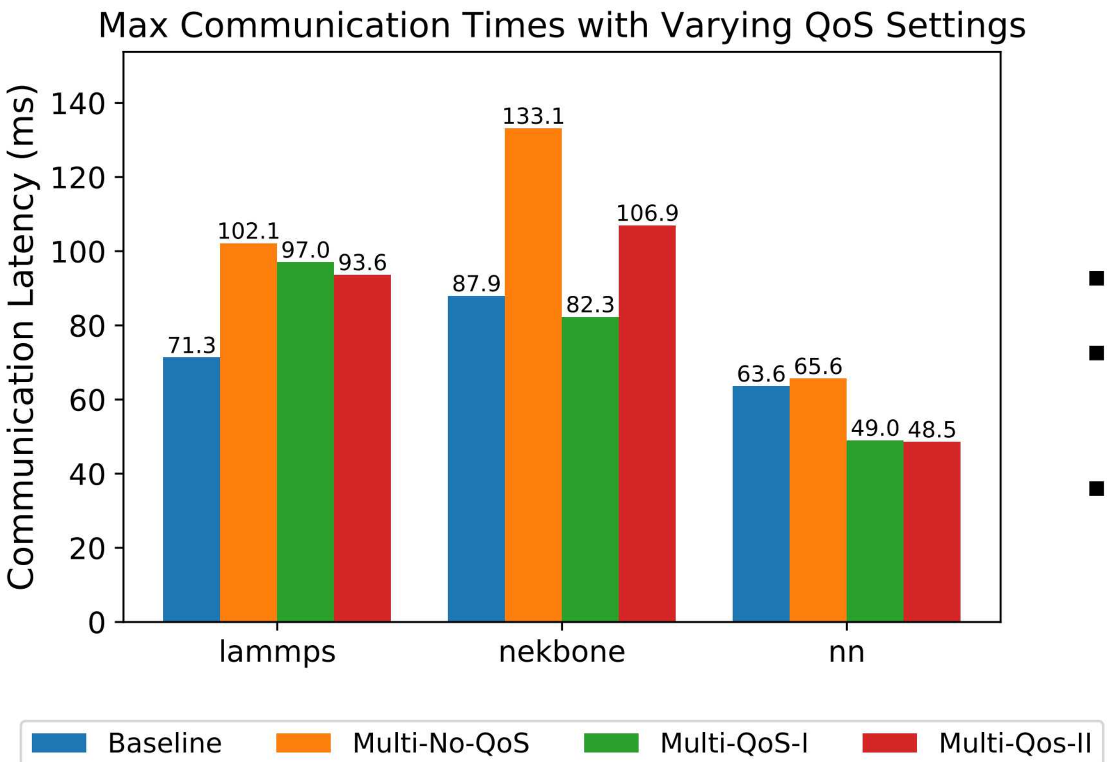
- Again examining Nekbone (2,197 ranks)
  – Given 10% of bandwidth as cap, **only for collectives**
- Uniform random background traffic on remaining ranks
  – Vary intensity as a percentage of link bandwidth

***Traffic differentiation focused on collectives can bring up to 60% speed up in communication time with collective-intensive applications such as Nekbone, with a modest bandwidth allocation.***



# Multiple applications running in parallel




- 3 SWMs in parallel:
  - Nekbone (2,197 nodes)
  - Nearest Neighbor (2,197 nodes)
  - LAMMPS (2,048 nodes)
  - Nekbone is most comm. intensive
- Baseline: Single SWM in isolation
- Multi-QoS-I: Prioritizing Nekbone and guaranteeing 1/3 BW
- Multi-QoS-II: Prioritizing collectives

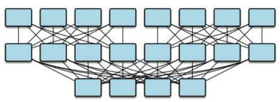
**Adding bandwidth cap on Nekbone helps improve the performance of other skeleton applications as well.**

# Jeremy Wilke

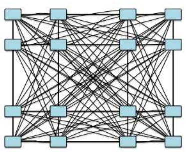
EXASCALE COMPUTING PROJECT

# Simulator Capabilities #4: Effectiveness of simple QoS strategies to alleviate interference effects

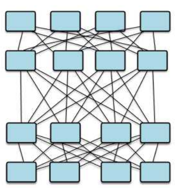
4 Topologies



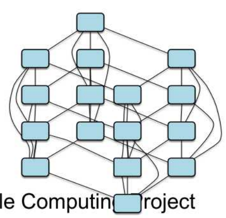
Fat tree



Dragonfly



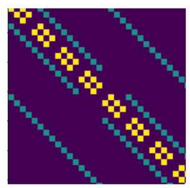
Dragonfly+



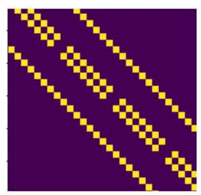
HyperX

3 Workloads x
2 Placements

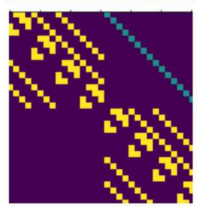
Random + Linear Placement



Halo



Swee
p



Subcomm (FFT)

2 Scales

- "Small" ATS: 8K nodes e.g. Summit
- "Big" ATS: 16K nodes e.g. Trinity

3 "Environments"

¼, No Background

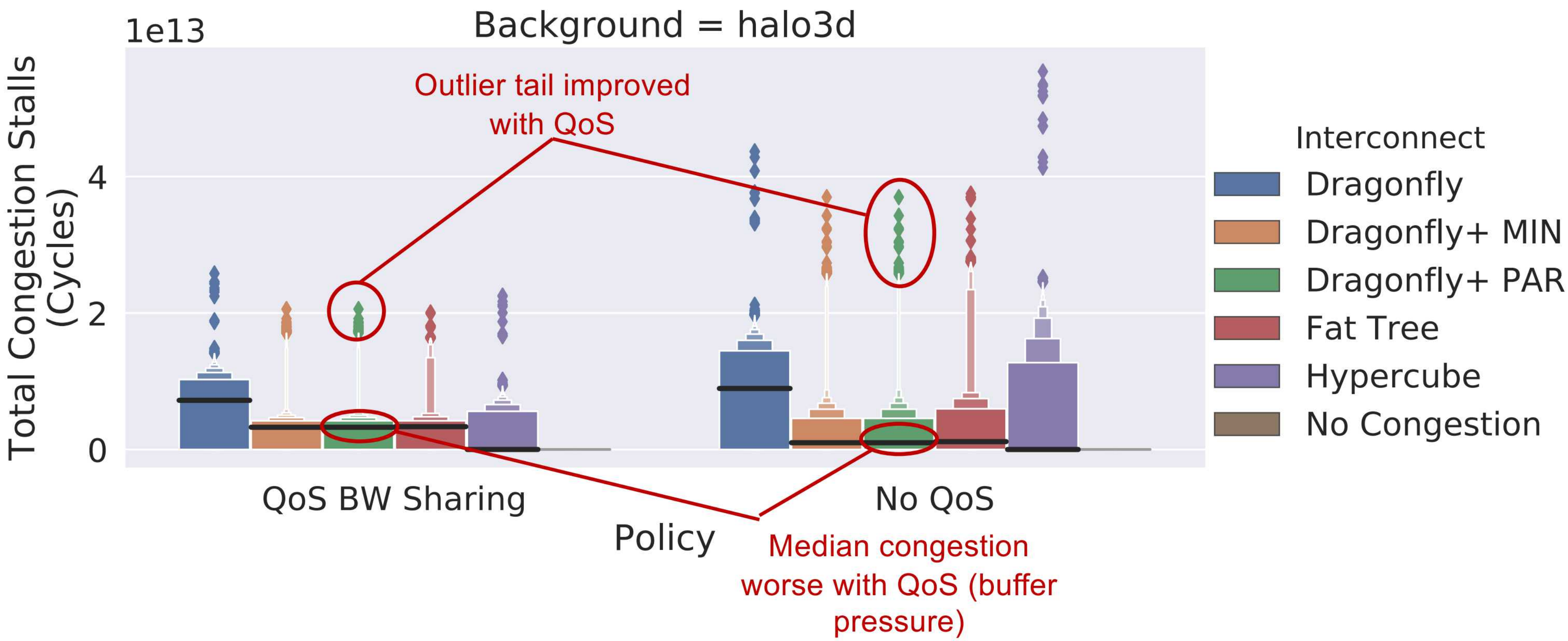
¼ + Halo Background

¼ + FFT Background

For each milestone, goal is broad survey over the interconnect design space covers different workloads and scales

# Subcom heavily affected by halo background traffic, QoS smooths performance

# QoS not only helps with fair BW sharing, but also seems to improve congestion outliers



# Taylor Groves

# Where are simulations needed

- Broadly speaking if we can study and adjust a parameter on a real system after delivery, simulation is less important.
  - However, simulation may be able to whittle down a large parameter space to the most important parameters to study on a live system
- If we need design or architectural insights that are hard to change in place we need to invest in simulation/models.

# What network simulation can help facilities with today

- Understanding bandwidth provisioning (particularly for new topologies)
  - lots of experience with dragonfly topologies, but no experience with MegaFly or HyperX
  - In some cases we don't have extra switches or cable ports available to expand the system if not designed in early
- Resilience studies: how are different topologies and routing strategies performing under link failures?
  - impractical to study this on production systems (has been done at the end of system life), but easy to simulate
  - this information could help influence the overall architecture we choose

# Where simulators need additional development

- Tying into network endpoints: complex hierarchies of memory and accelerators make end-point simulation difficult to incorporate (e.g. kernel launch/synchronization overheads b/n GPU communication)

- Provisioning the right number of NICs for a specific CPU/Accelerator and network topology

- Do accelerators drastically change the communication patterns of our motifs and traces that we are using today?

# Sudheer Chunduri

EXASCALE COMPUTING PROJECT

# Simulation Use Cases

- Use cases
  - Optimal network operating parameters
    - QoS/Traffic classes configuration
    - Routing biases
  - Feature interplay
    - Congestion management, routing and QoS
  - System design decisions
    - E.g.: How much injection bandwidth is good enough?
- Simulation is the only way (most of the times) to study these
  - No hardware available
  - Too costly or not feasible to evaluate on real machines

# Where do we go from here

- Where things could be improved
  - Fidelity of the simulations
    - Representation of the real application characteristics
      - Often missing computation component and communication-computation overlap aspect
    - Validation against the real systems
  - Translation of simulation observations into actionable insights useful for real systems
    - Scale issues

# Ian Karlin

EXASCALE COMPUTING PROJECT

# High-level Simulation Use Cases

- Machine configuration decisions
- Non-Recurring Engineering (NRE) Engagements
- Co-design and Pre-RFP Discussions

# Detailed Questions Simulations Can Help With

- How job placement and task mapping impact performance and how sensitive a given network is to these factors? Are there cases when the next job should not be scheduled to help defragment a system?
- What application or system level data can we measure to help us understand how well the network is performing? For example, switch counters, link utilization, etc.
- What the tradeoffs of configuring the network are to applications? For example, tapering, global links, etc.

# Current Issues With Using Simulators for Our Work

- Hard to understand their accuracy/error bars
- Have not been shown to be predictive onto future networks
- Validation is often not through or well documented
- Need to have someone on the simulator project to perform work fast enough

# Discussion

# Acknowledgments

*This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations – the Office of Science and the National Nuclear Security Administration, responsible for preparing a capable exascale ecosystem, including software, applications, hardware, and early testbed platforms, to support the nation's exascale computing imperative.*

EXASCALE COMPUTING PROJECT

www.ExascaleProject.org

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science