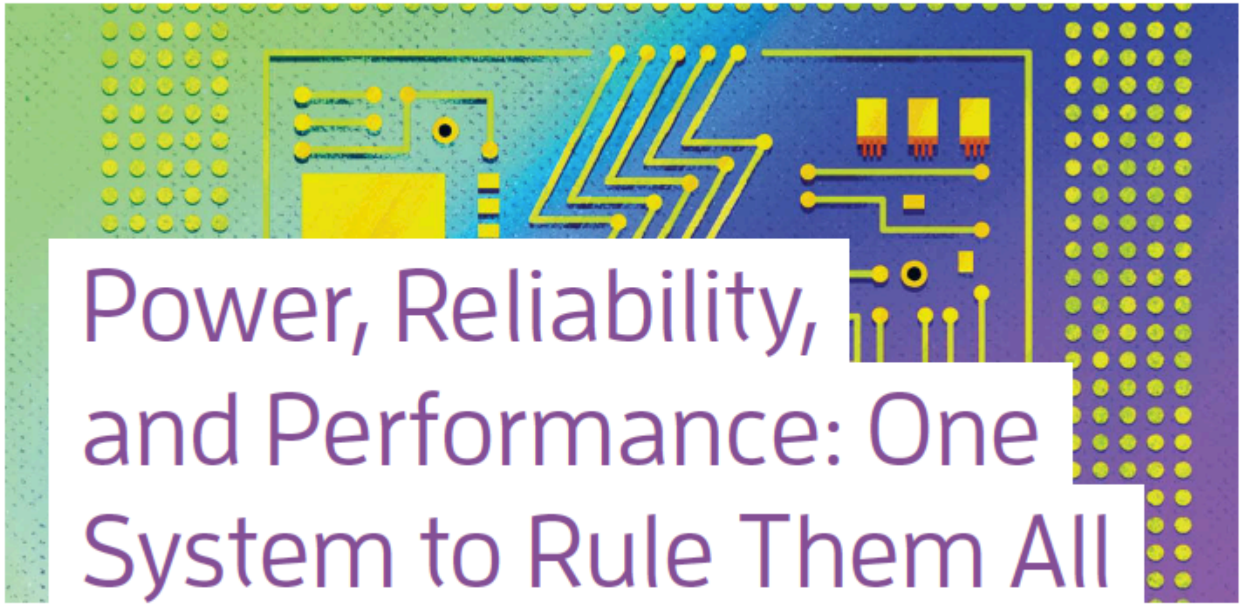
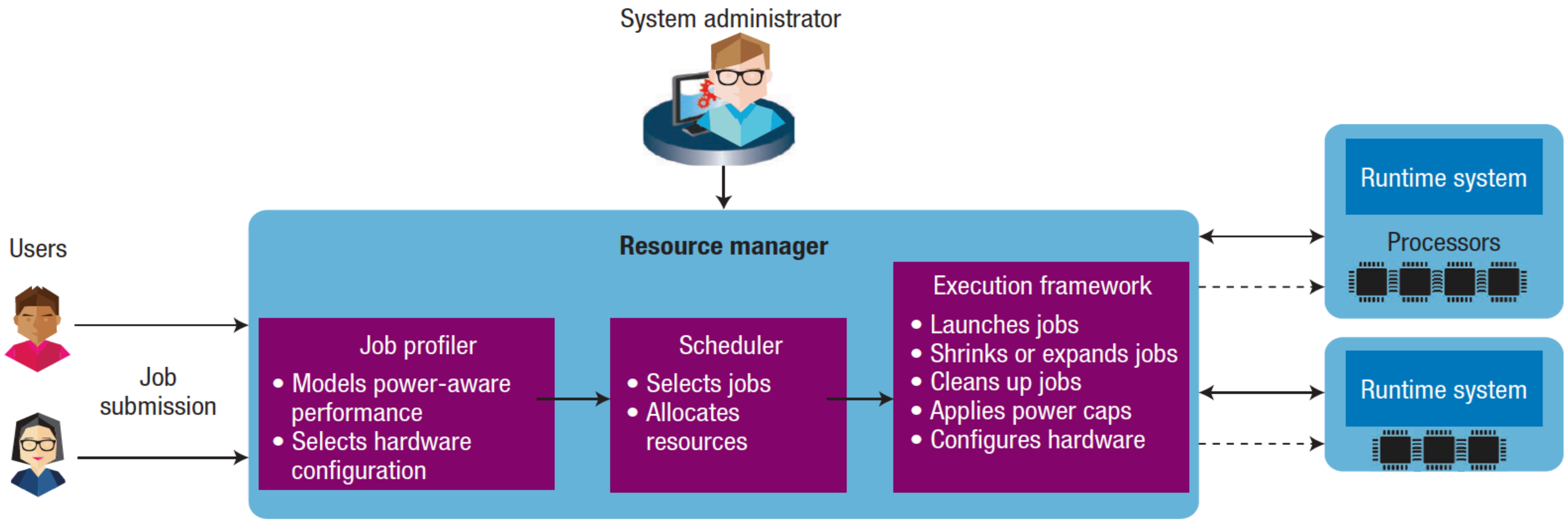
Charm++ as an Energy Efficient Runtime

COVER FEATURE **ENERGY-EFFICIENT COMPUTING**



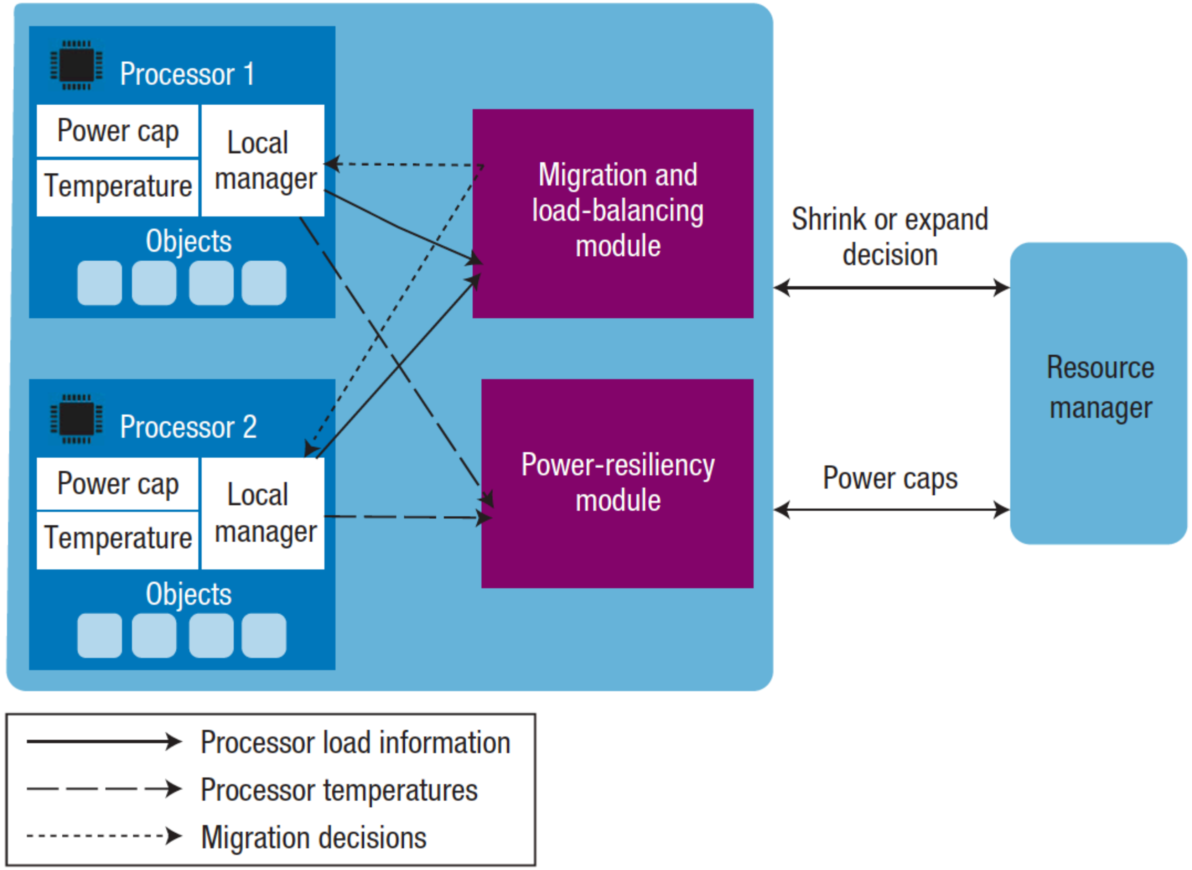
Bilge Acun, University of Illinois at Urbana–Champaign
Akhil Langer, Intel
Esteban Meneses, Costa Rica Institute of Technology and Costa Rica National High Technology Center
Harshitha Menon, University of Illinois at Urbana–Champaign
Osman Sarood, Yelp
Ehsan Totoni, Intel Labs
Laxmikant V. Kalé, University of Illinois at Urbana–Champaign

Interaction Between the Runtime System and the Resource Manager



- ✓ Allows dynamic interaction between the system resource manager or scheduler and the job runtime system
- ✓ Meets system-level constraints such as power caps and hardware configurations
- ✓ Achieves the objectives of both datacenter users and system administrators

Components of Charm++ with Its Interactions



Charm++ has three main components:

- **Local manager:** tracks local information such as object loads, CPU temperatures
- **Load-balancing module:** makes load-balancing decisions and redistributes load
- **Power-resiliency module:** ensures that the CPU temperatures remain below the temperature threshold, change the power cap

Support for Proactive Cooling Decisions with Neural Network-Based Temperature Prediction

BILGE ACUN[1], EUN KYUNG LEE[1], YOONHO PARK[1], LAXMIKANT V. KALE[2]

[1] IBM T.J. WATSON RESEARCH CENTER

[2] UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

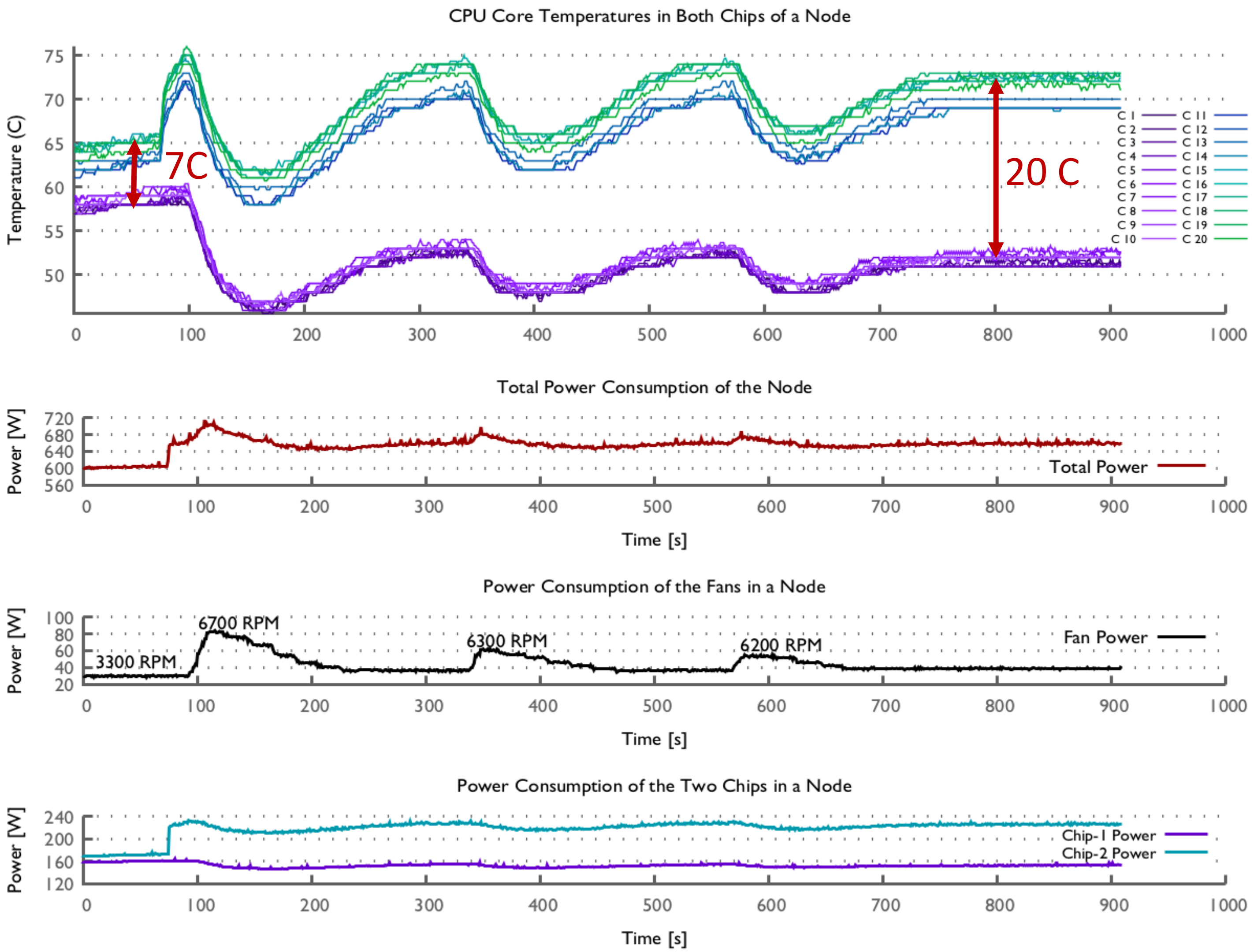
Motivation

1. Pressure of reducing the power consumption and carbon footprint of datacenters and supercomputers is increasing

2. Other expected problems include:
   - Larger process variations, temperature variations
   - More heat dissipation
   - Denser nodes with different components in the node such as GPUs, co-processors that have different temperature, cooling characteristics

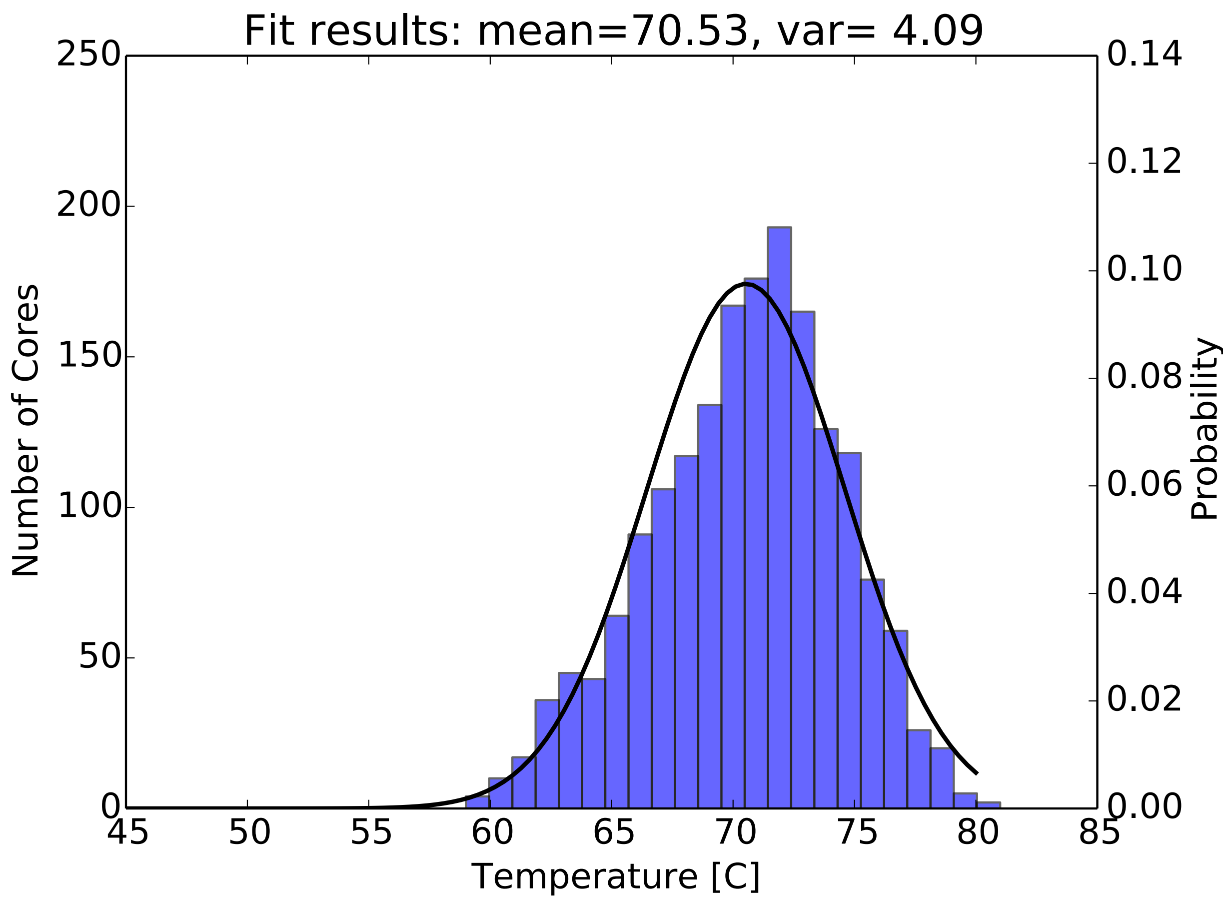
Motivation

- **Temperature variations among cores:**
  - 7 C in idle temperatures
  - 9 C in all active temperatures
  - 20 C idle/active mixed
- **Synchronous fan control:**
  - 4 independent fans in the node
  - Fans all act together and cause even further temperature variation
- **Reactive cooling behavior:**
  - 54 W jump in fan power
  - 10 minutes stabilization time with a regular workload

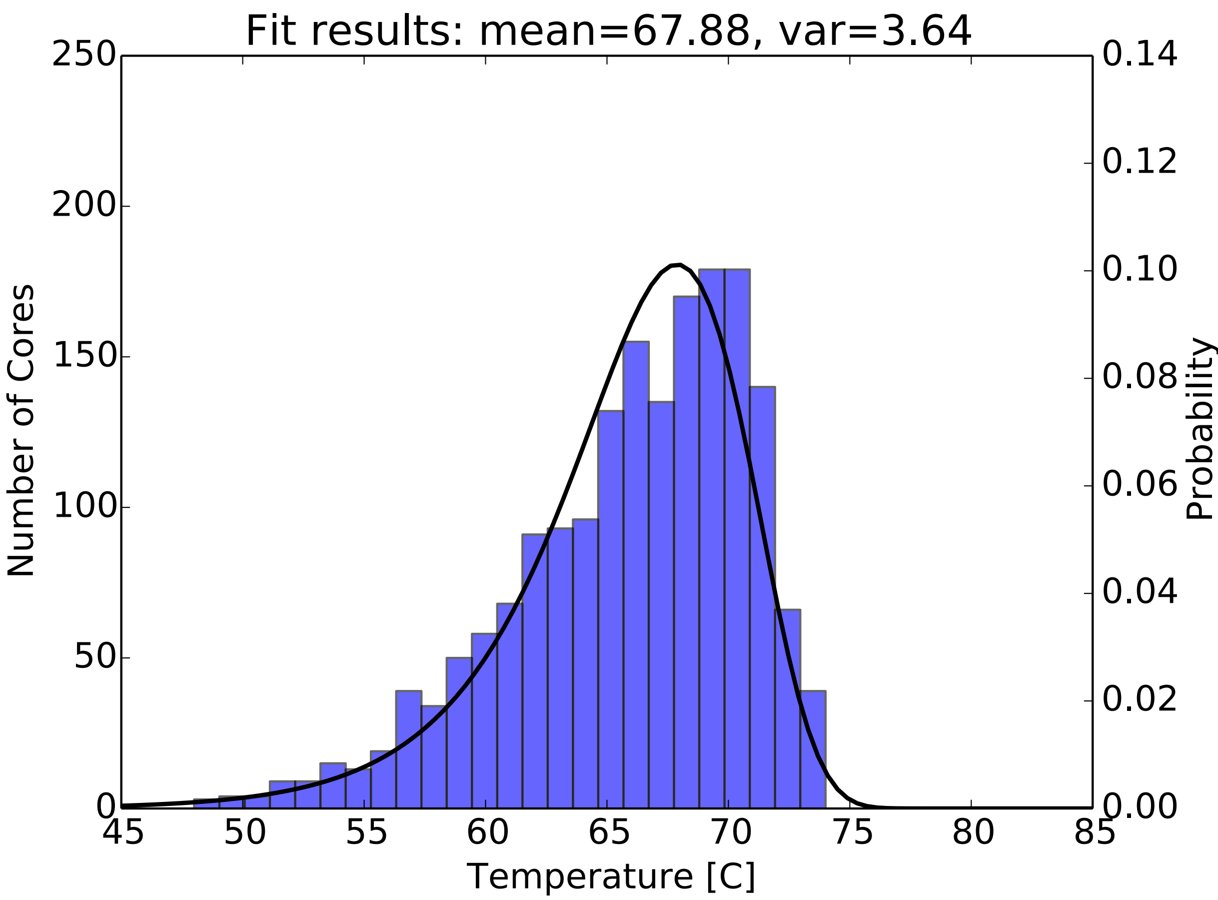


Temperature Variation in Large Scale

Temperature distribution of 1800 cores



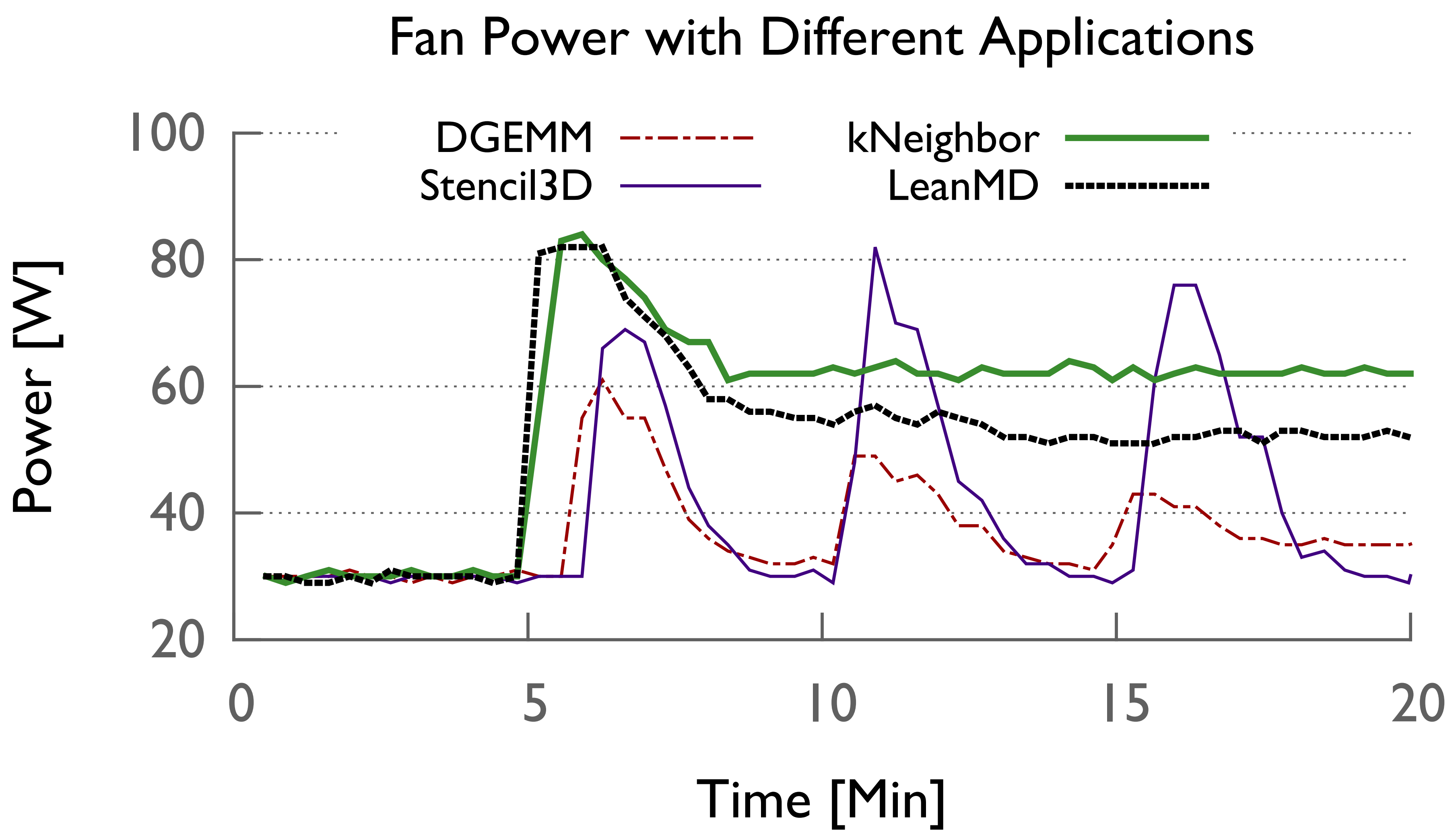
Cori at NERSC – Intel Haswell



Minsky at IBM POWER8

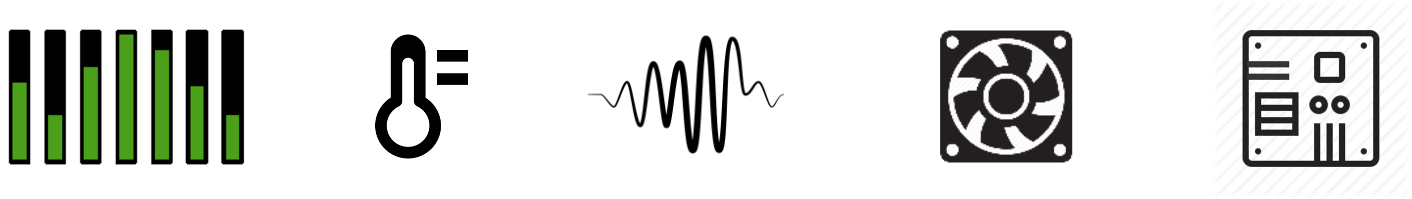
Oscillatory Cooling Behavior

Fan Behavior of Different Applications



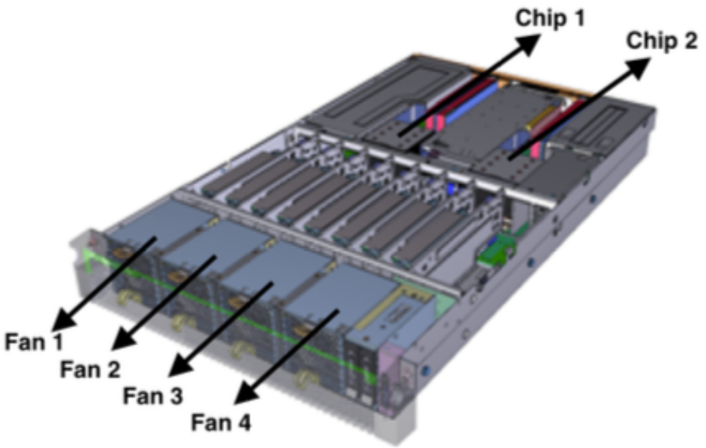
Why Temperature Modeling is Difficult?

- There are lots of parameters affecting the core temperatures:
  - Complex workloads
  - Ambient temperature
  - Core frequencies
  - Fan speed level
  - Physical layout
  - Hardware variations



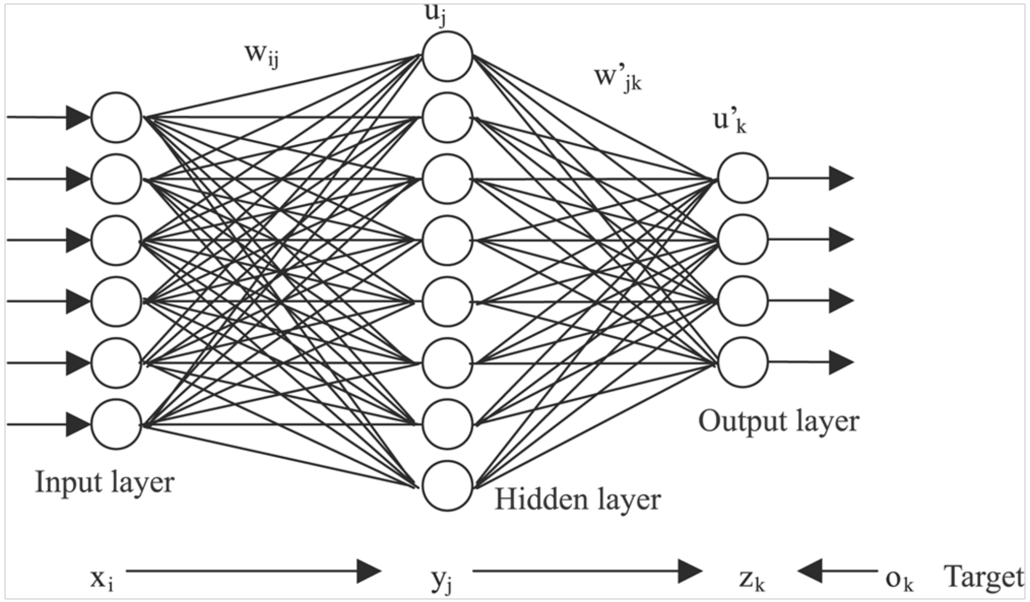
- Combination of these parameters create an exponential modeling space
  - 10 different cores
  - 0-100 CPU utilization levels
  - 44 different frequency levels
  - 3000 RPM-10000 RPM fan speed levels
  - 4 fans

❖ $(10^{10}) * 44 * (10^4) = \sim 2^{52}$



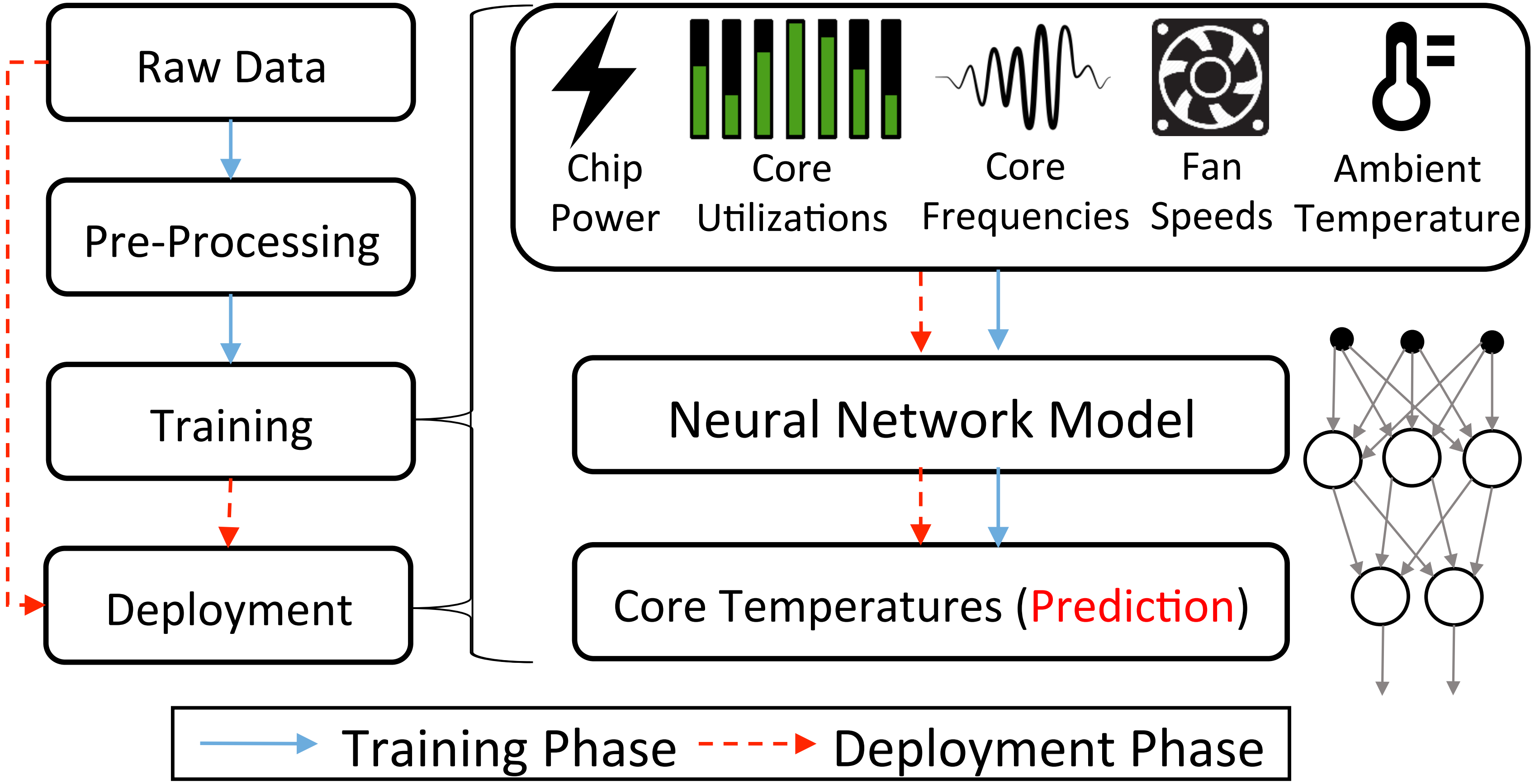
Neural Networks for Temperature Modeling

- Neural networks are good because:
  - They can capture linear and non-linear behavior between input and output parameters
  - They work well in noisy data
  - They do not need for formulation of an objective function
- Neural networks has been used in HPC for:
  - Energy and power modeling [1]
  - Performance modeling [2]
  - Temperature modeling
    - For GPU temperature modeling [3]
    - For coarse-grained data center level modeling [4]



1. A. Tiwari, M. A. Laurenzano, L. Carrington, and A. Snavely. Modeling power and energy usage of HPC kernels. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, IEEE, 2012.
2. B. C. Lee, D. M. Brooks, B. R. de Supinski, M. Schulz, K. Singh, and S. A. McKee. Methods of inference and learning for performance modeling of parallel applications. In *Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '07, 2007.
3. A. Sridhar, A. Vincenzi, M. Ruggiero, and D. Atienza. Neural network-based thermal simulation of integrated circuits on GPUs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31.
4. L. Wang, G. von Laszewski, F. Huang, J. Dayal, T. Frulani, and G. Fox. Task scheduling with ann-based temperature prediction in a data center: a simulation-based study. *Engineering with Computers*, 2011.

Neural Networks for Temperature Prediction

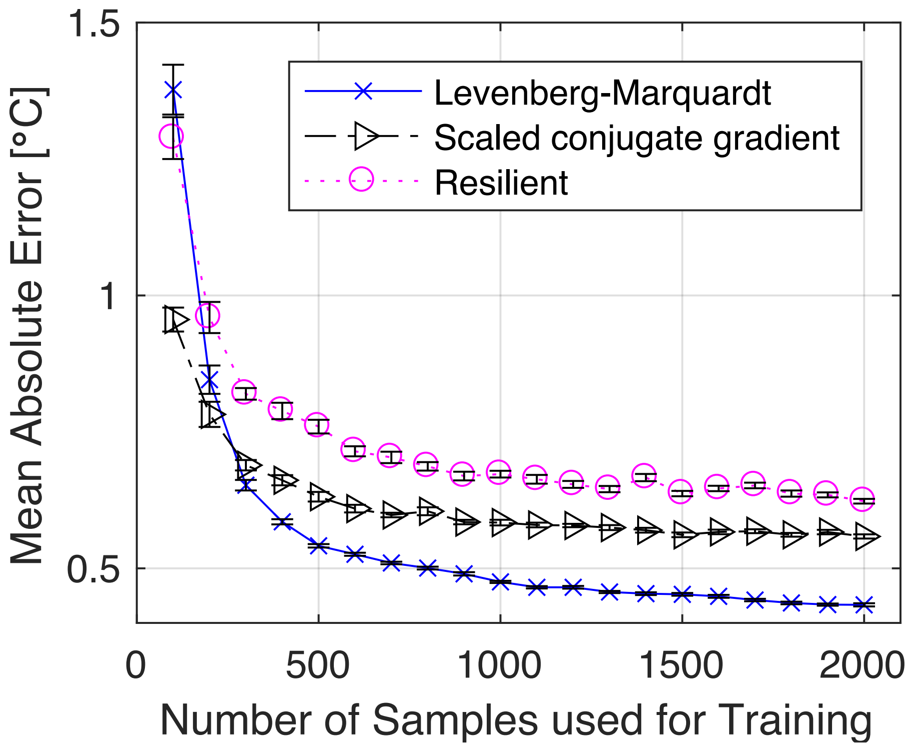


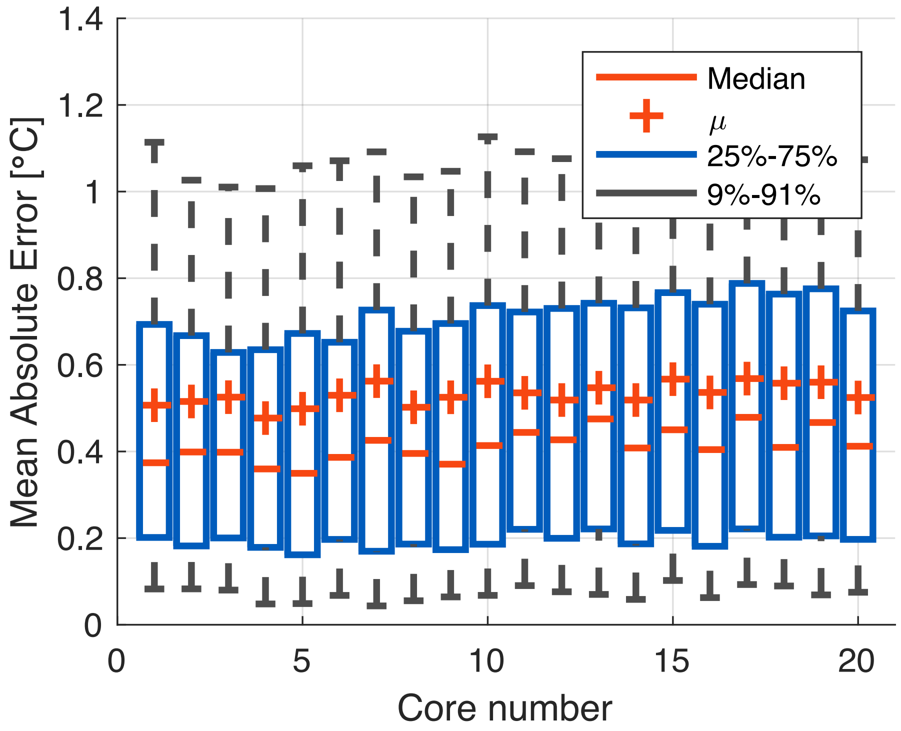
Experimental Setup:

- Firestone cluster at IBM with Power 8 processors
- 1 node = 2 sockets, 20 physical cores, 160 SMT cores
- OCC, and BMC for temperature, power readings

Neural Network Configuration and Validation

- We test different back-propagation algorithms with different time and memory requirements.





- Other configurations include number of layers, and number of neurons.

Model Guided Proactive Cooling Decisions

1. Fan control
   - This can reduce chip-to-chip temperature variations.
   - What should be the fan speed level to be able keep the chips at a certain temperature limit?

2. Load balancing
   - This can remove core-to-core, as well as chip-to-chip temperature variations.
   - What would the core temperatures become if a certain amount of data is moved from one core to another?

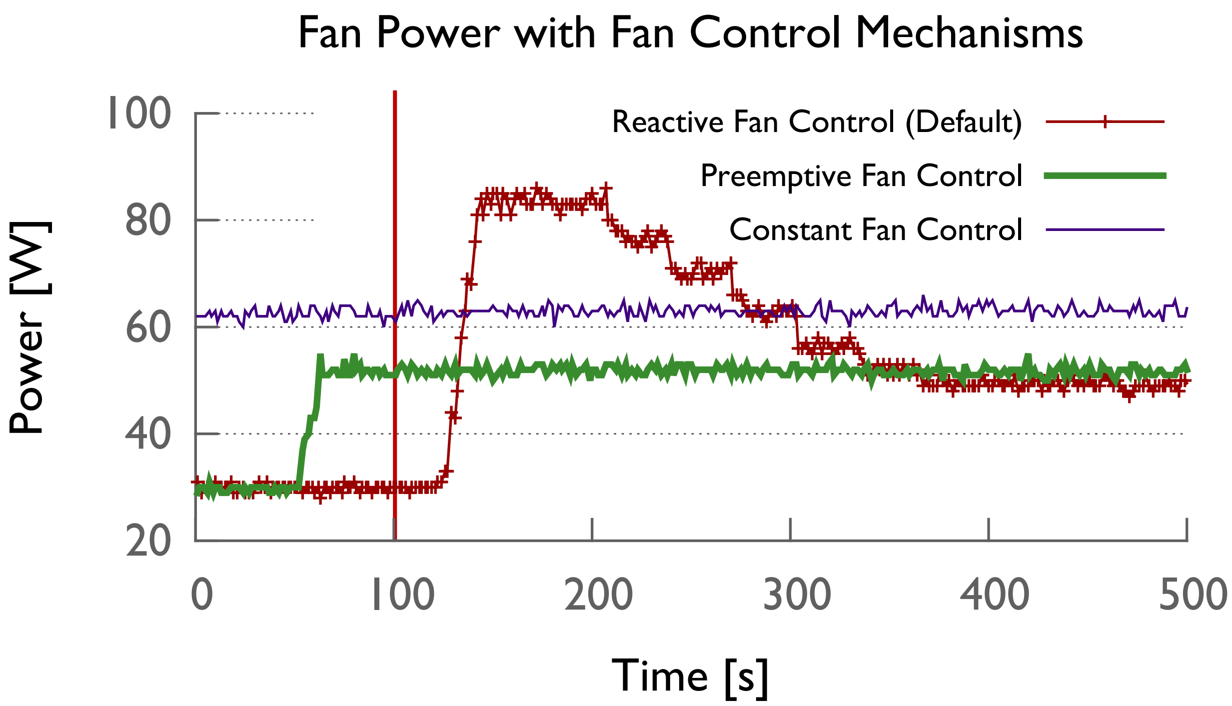
3. DVFS
   - Chip-level DVFS can reduce chip-to-chip, core level DVFS core-to-core temperature variations.
   - What frequency level we need to set for the cores to stay under a temperature limit for a workload?

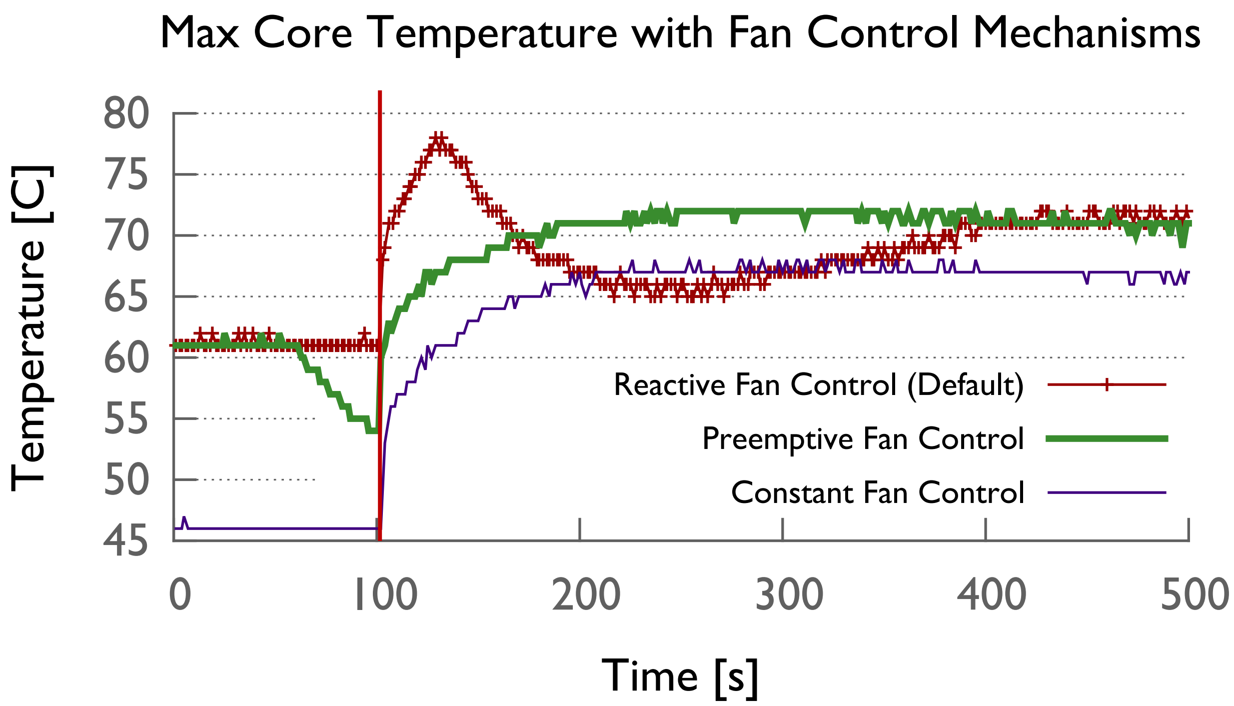
Model Guided Proactive Cooling Decisions

1. Fan control
   - This can reduce chip-to-chip temperature variations.
   - What should be the fan speed level to be able keep the chips at a certain temperature limit?

Proactive Fan Control Mechanism

- The key idea is cool the processor proactively, for example, before the application starts.

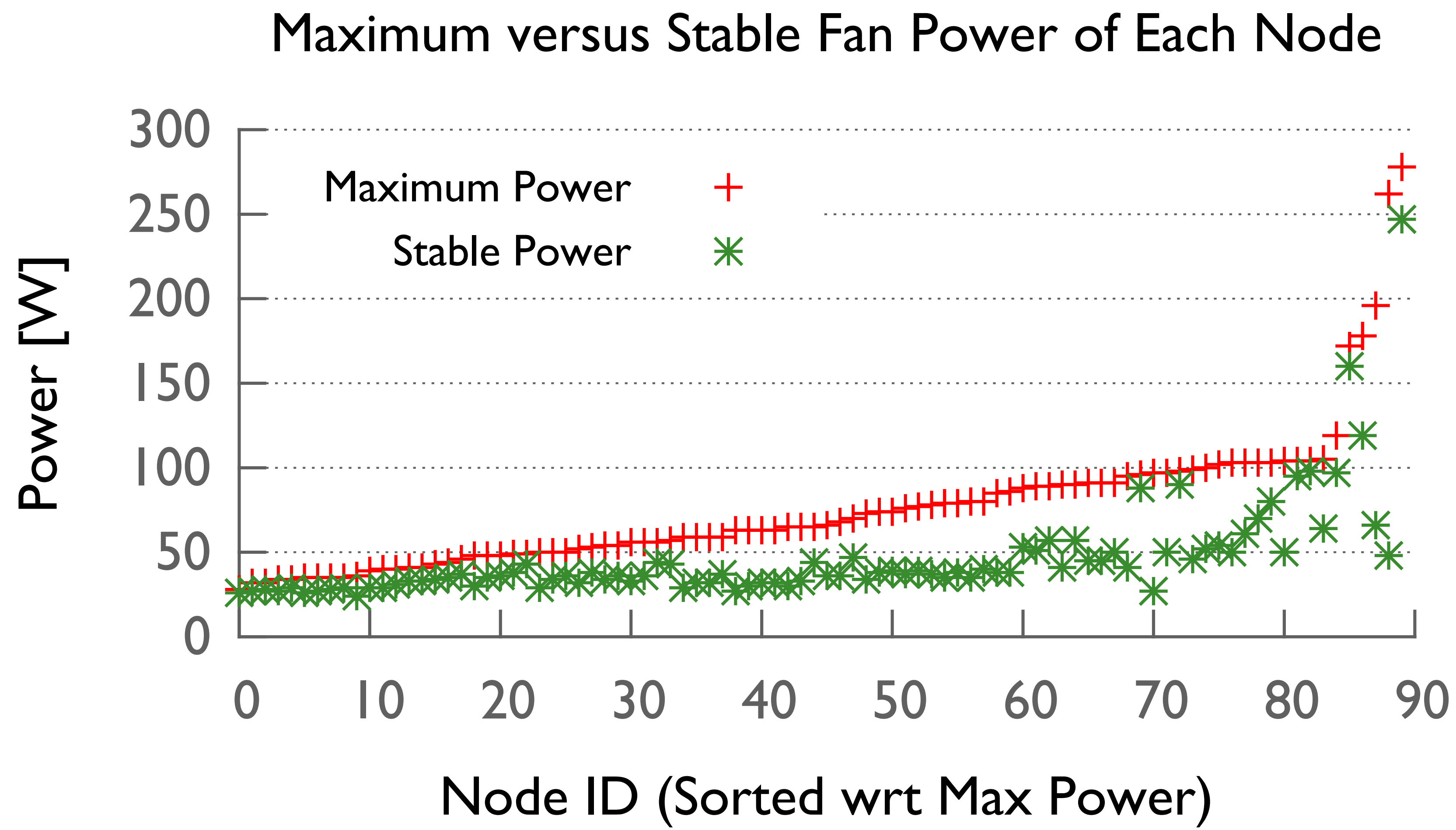




- Preemptive fan-control removes temperature peaks, and is able to keep the temperature as the same level as reactive fan control.

- It can be done via job scheduler, and/or runtime without taking over the total control of the fan.

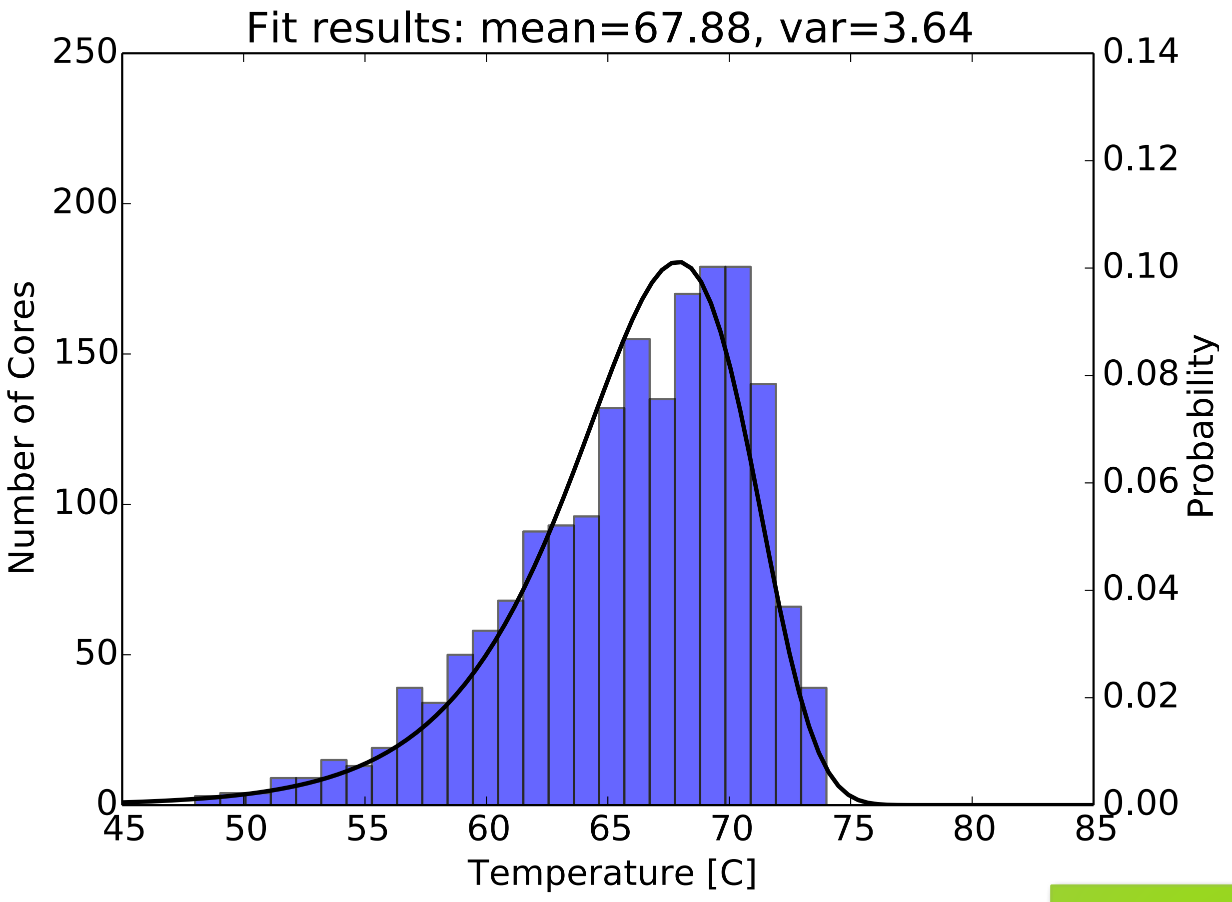
Power Reductions With Proactive Cooling



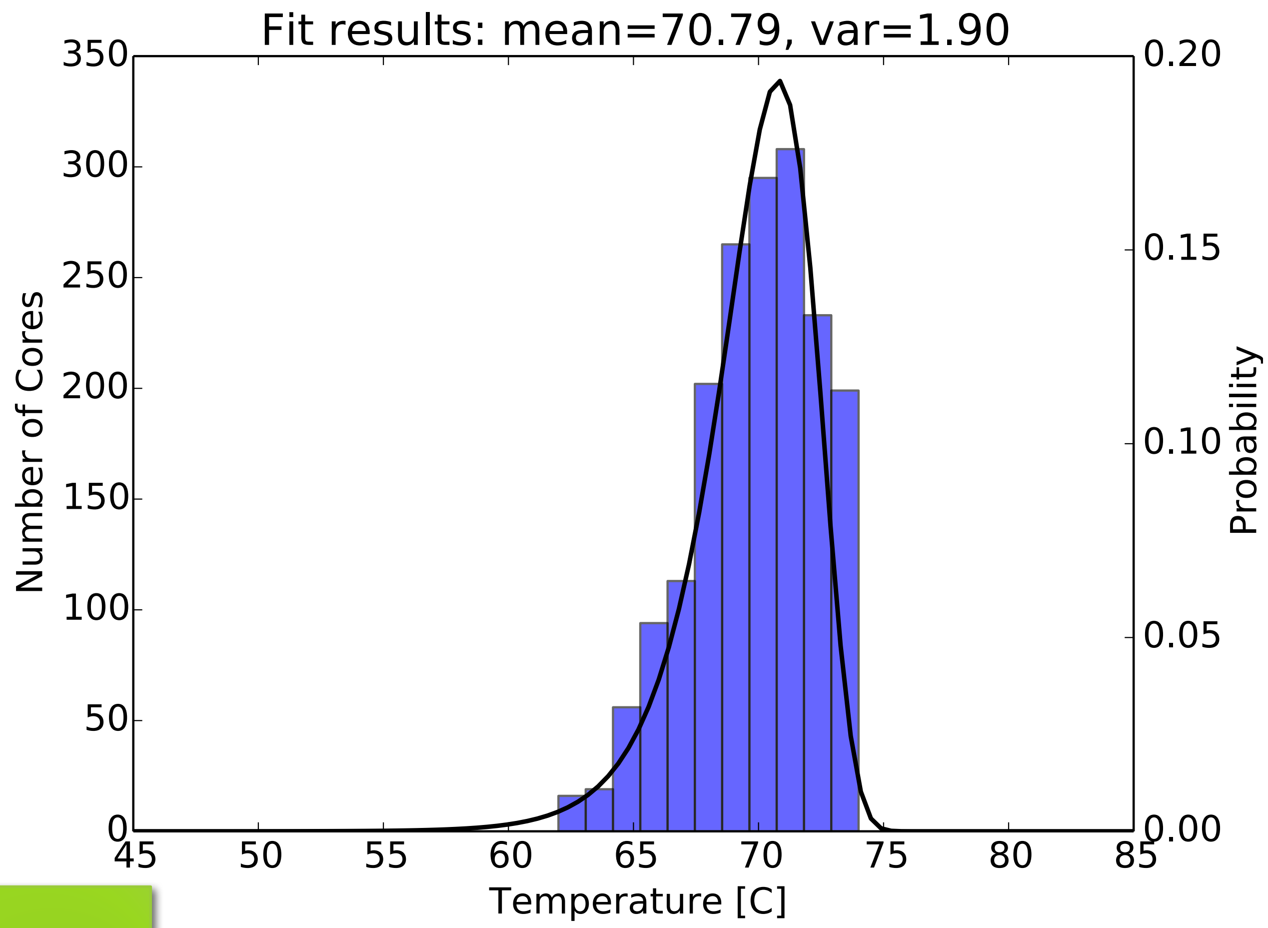
35% reduction in fan power

Power Reduction = Maximum Power – Stable Power

Decoupling the Fans



BEFORE



AFTER

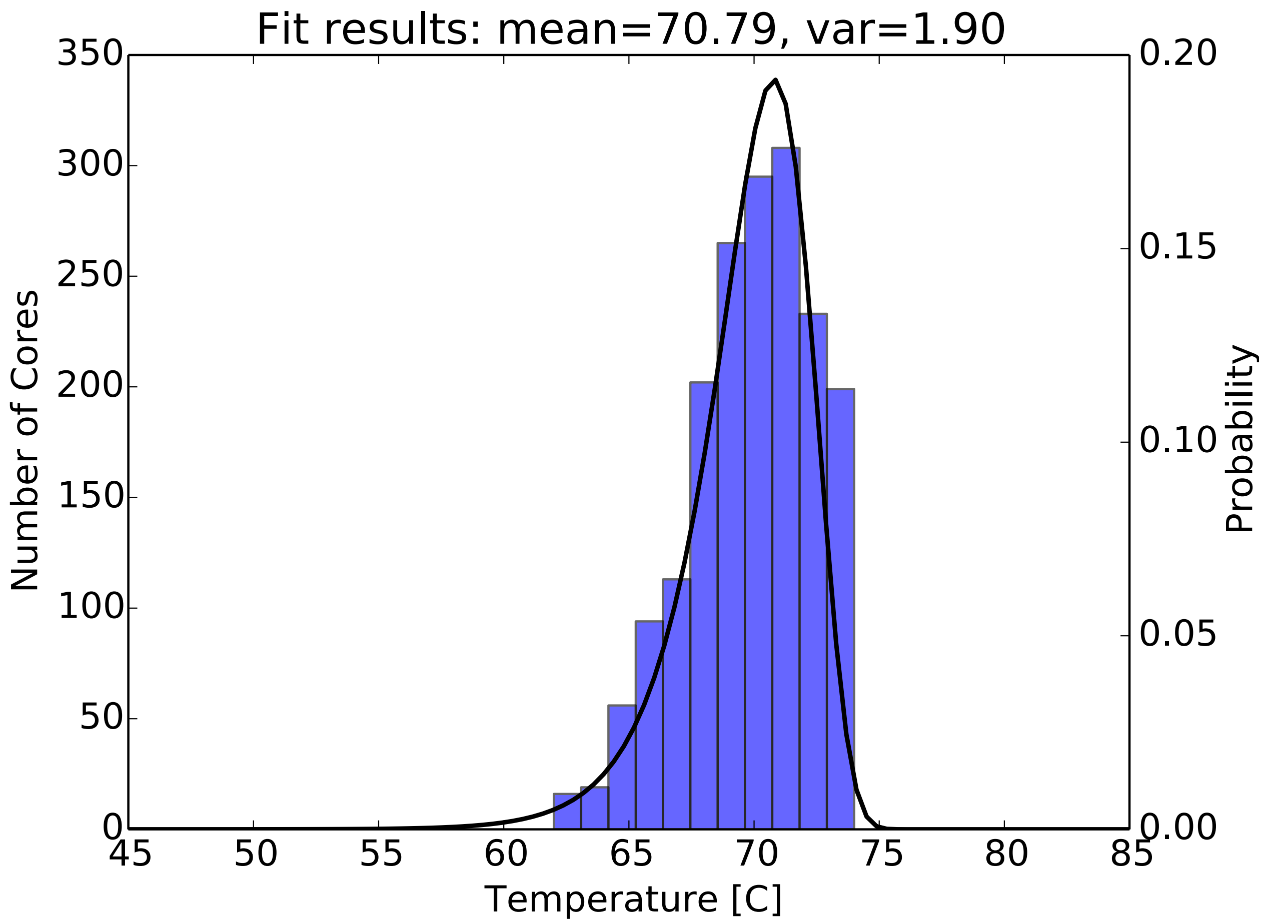
18% reduction in fan power

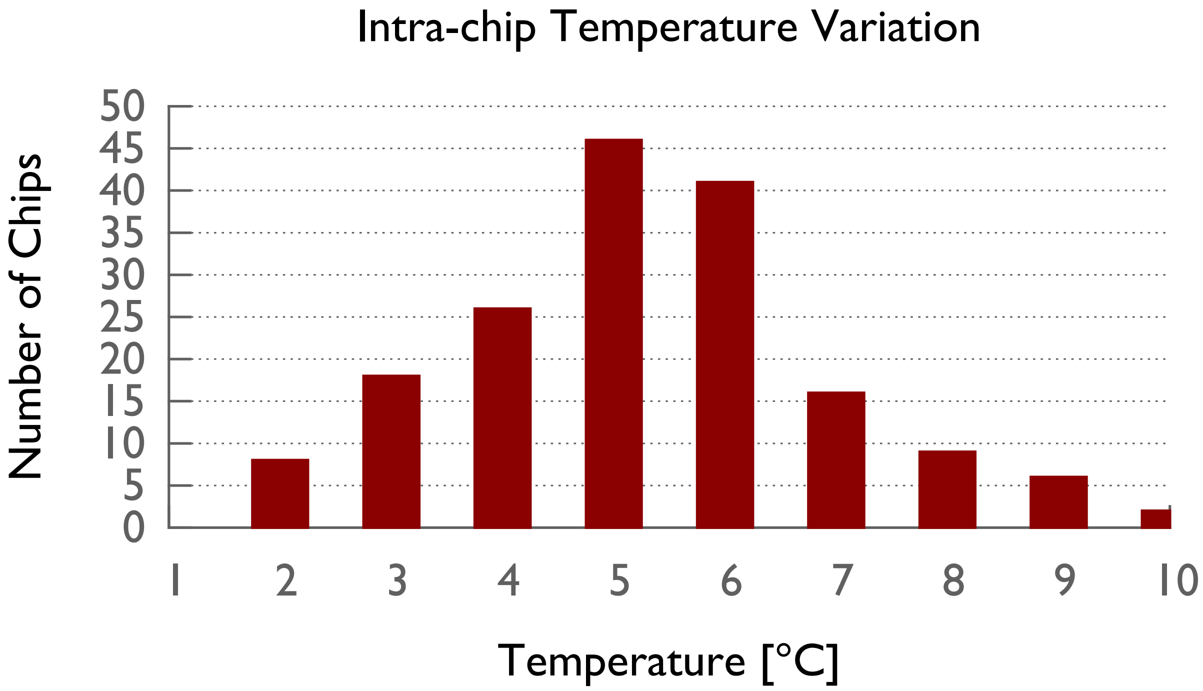
Total Reduction in Fan Power

Optimization \ Benchmarks	DGEMM	Stencil3D	kNeighbor	LeanMD	Average
Reactive Fan Control	5868 W	13433 W	6769 W	6770 W	8210 W
Preemptive Fan Control	3893 W	8526 W	4381 W	4224 W	5256 W
Preemptive and Decoupled Fan Control	3179 W	7972 W	3765 W	3569 W	4621 W
Total Power Reduction (%)	45.8	59.3	55.6	52.7	53.3

53% reduction
in fan power on average

Remaining Temperature Variation





- DVFS?
- Load Balancing?

Temperature-Aware Load Balancing With Charm++

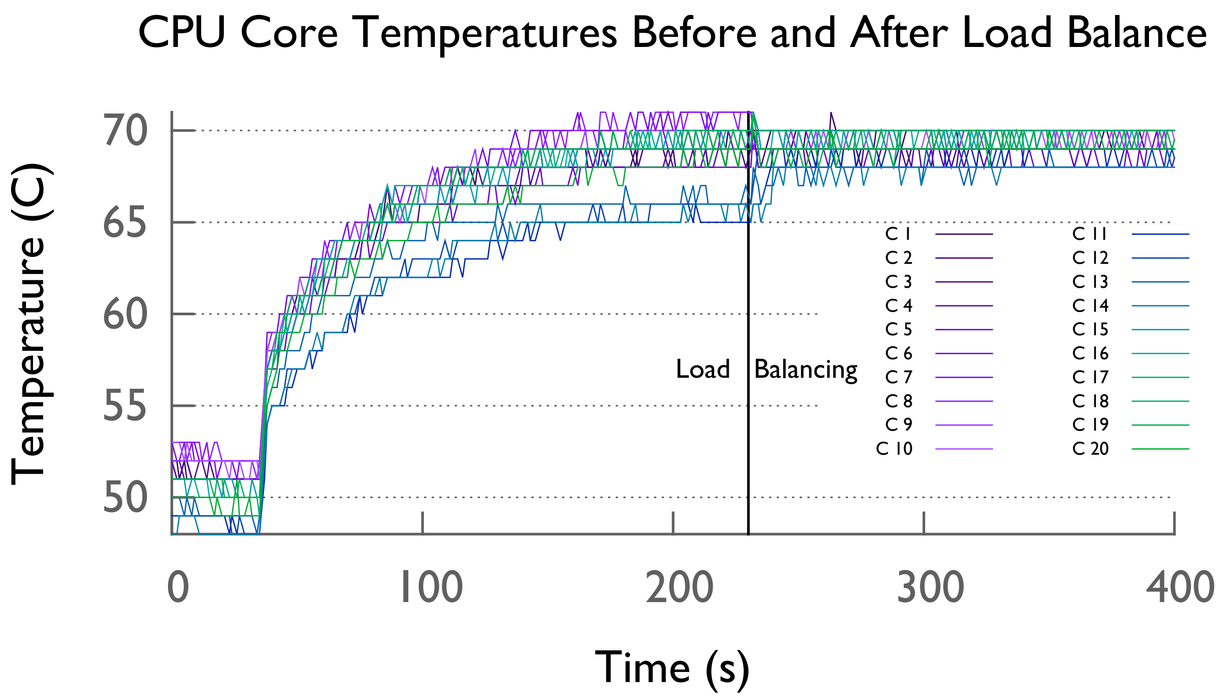
- Load balancing can help reduce the temperature variations, but how do we decide how much load to move?

- Charm++ [1] has an runtime database which stores:
  - Number of tasks per process
  - Load of each object (in terms of execution time)
  - Communication load of each object

- Load balancing is triggered periodically with customizable periods

- We implement our temperature-aware model guided load balancing algorithm.

- Load balancing has potential to remove both chip and core level variations.



1. B. Acun, et al. Parallel programming with migratable objects: charm++ in practice. *In SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 647-658. IEEE, 2014.

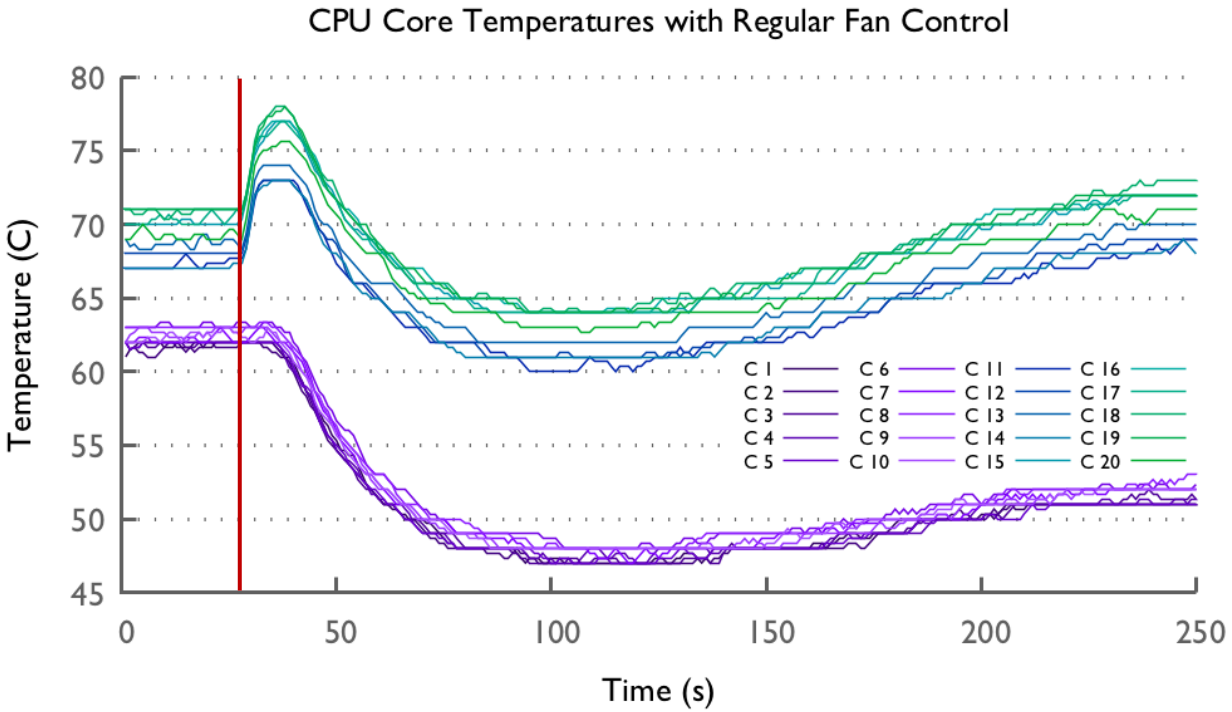
Conclusion

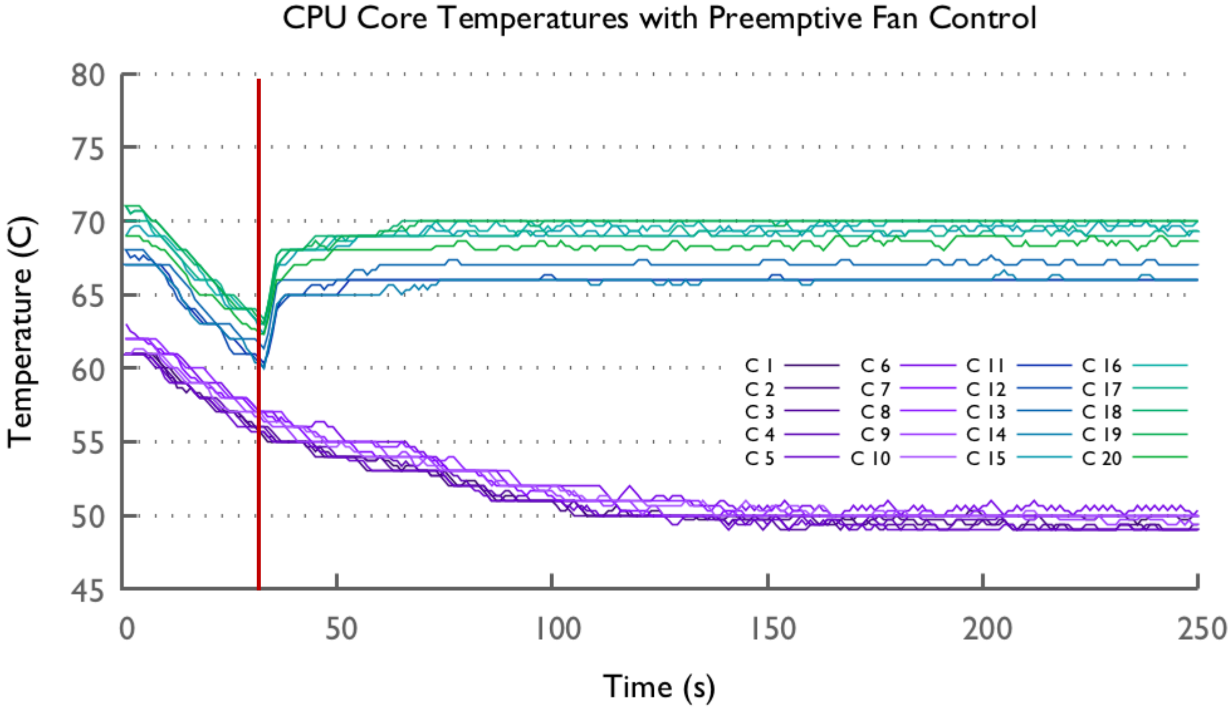
- In summary, we propose:
  - A neural-network based temperature prediction model
  - Proactive cooling mechanisms:
    - Fan control
    - Load balancing
- Our results shows:
  - We can accurately predict core temperatures
  - Peak fan power can be reduced by 53%
  - Air cooling systems can be made more efficient

Thank you!

Comparison of Reactive vs Preemptive Fan Control

- The key idea is cool the processor proactively, for example, before the application starts.





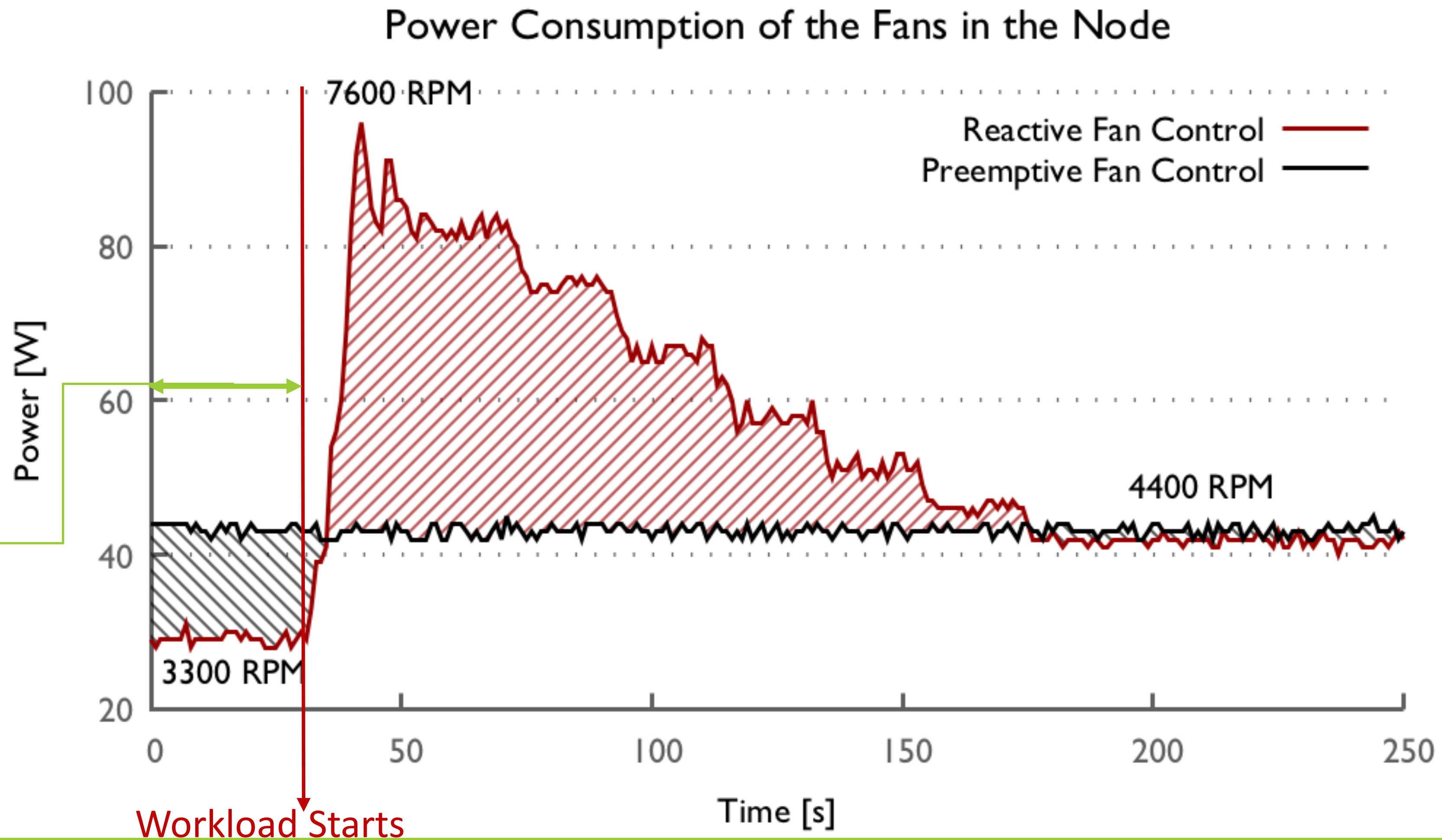
- Preemptive fan-control removes temperature peaks, and is able to keep the temperature as the same level as reactive fan control.

- It can be done via job scheduler, and/or runtime without taking over the total control of the fan.

Power Reductions in Preemptive Fan Control

- Peak fan power can be reduced by 54 Watts = 58% reduction in cooling power.
- 2790 Joules of energy is saved = Red area – black area

How early to set the cooling speed?



Workload Starts