Application-aware Energy Efficient High Performance Computing

Laura Carrington

University of California, San Diego

San Diego Supercomputer Center

Performance Modeling and Characterization Lab (PMaC)

SAIM MS231

Mar. 1st, 2013

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

PMaC
Performance Modeling and Characterization

Overview

- Motivation
- Description of framework for application-aware energy efficient HPC
- Strategies for energy efficiency
- Results & contributions

Current HPC systems

Today's largest systems have 1M+ cores requiring over ~10 MW of power

- **Constructed with simpler cores to reduce power draw (e.g. GPUs, BG, MIC, ARM, etc.)**
- **Simpler core = less logic also requires more of programmer to get efficiency**
- **Even with lower power cores – energy efficiency is still an issue**

Growing these systems/future systems

10 X cores = 10 X power = ~100 MW!

THIS IS THE POWER WALL

PMaC's Green Queue Framework (optimizing for performance & power)

Goal: Develop automated framework that uses power and performance models to make application-aware energy optimizations during execution (now:DVFS future: power gating)

DVFS: Reduce the speed (clock frequency) of CPU in exchange for reduced power consumption

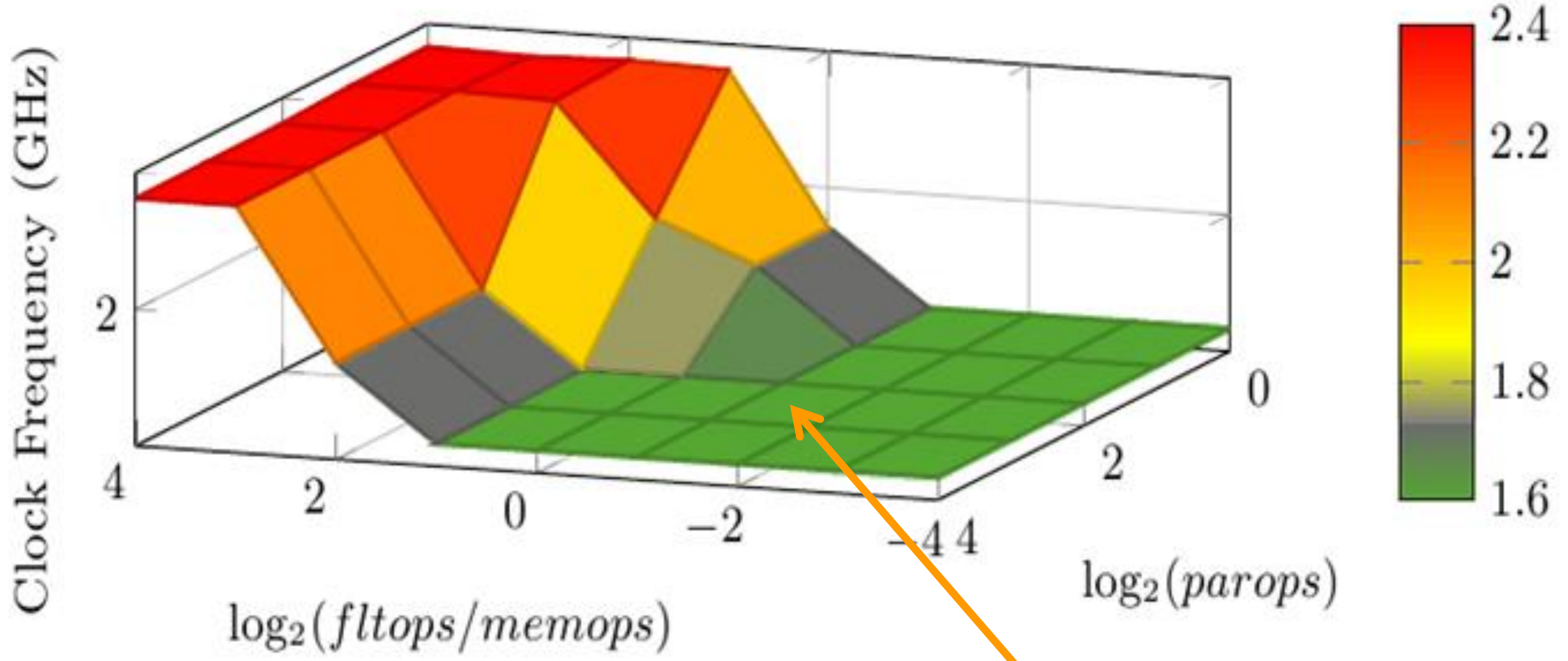
PMaC's Green Queue Framework
(optimizing for performance & power)

Goal: Develop automated framework that uses power and performance models to make application-aware energy optimizations during execution (now:DVFS future: power gating)

DVFS: Reduce the speed (clock frequency) of CPU in exchange for reduced power consumption

- **Different computations have different power requirements.**
- **For computations where the CPU is waiting for resources the frequency can be reduced to lower power with minimal performance impact.**

Identify the power and performance affects of different computational work



Energy savings via reduce processor frequency – minimal performance impact

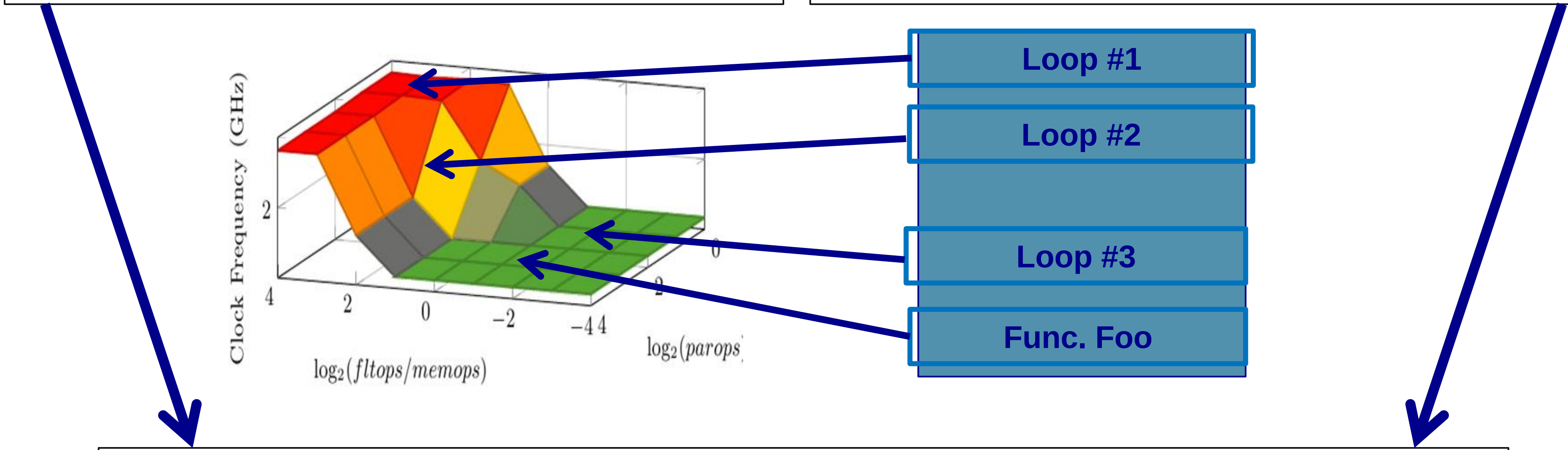
Application-aware Energy Efficient HPC

HPC System

Characterize the computational (& communication) patterns affect the overall power draw

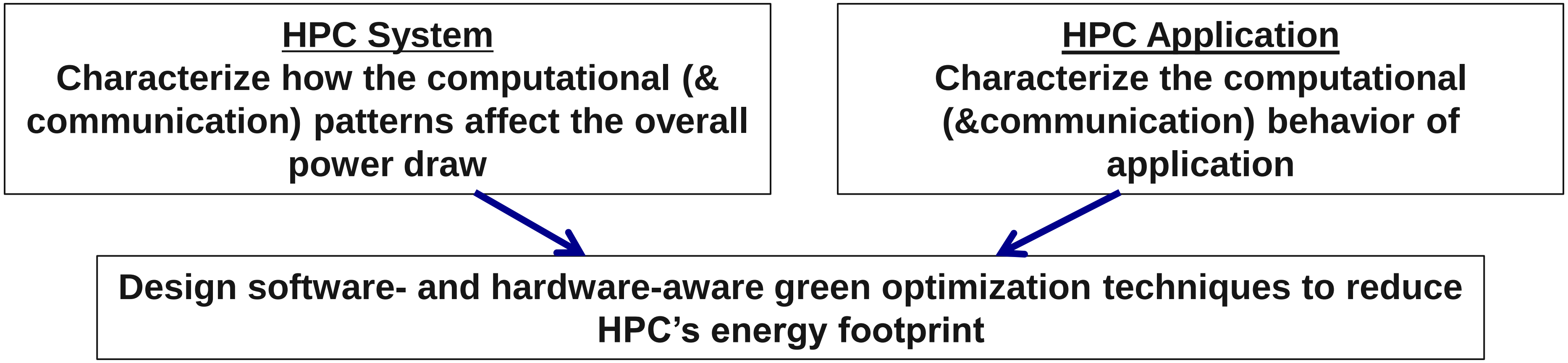
HPC Application

Characterize the computational (& communication) behavior of application



Design software- and hardware-aware green optimization techniques to reduce HPC's energy footprint

PMaC's Green Queue Framework
(fine-grained application-aware DVFS strategies)



HPC System
Characterize how the computational (& communication) patterns affect the overall power draw

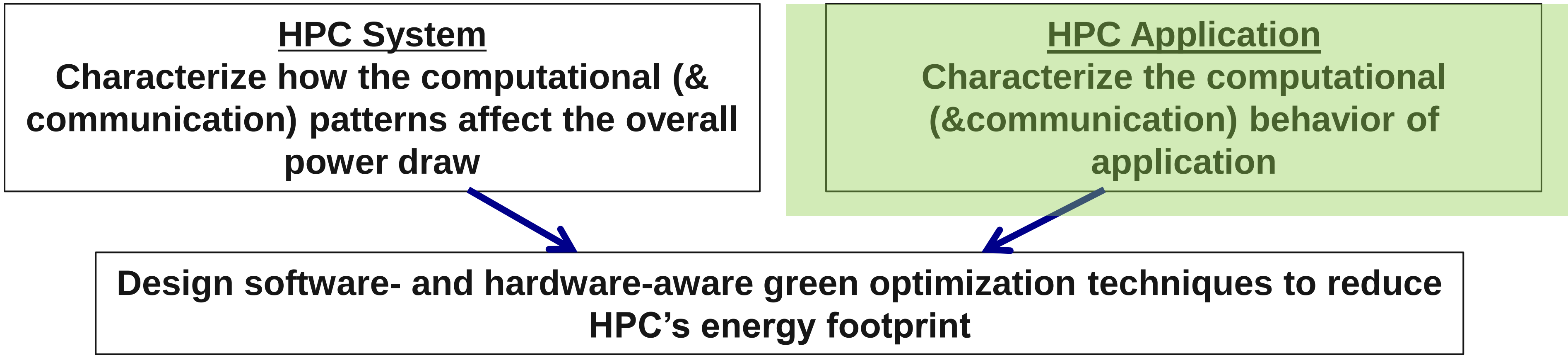
HPC Application
Characterize the computational (&communication) behavior of application

Design software- and hardware-aware green optimization techniques to reduce HPC's energy footprint

PMaC's Green Queue automated framework:

- Characterizes system's power draw behavior by running various computational work and uses to train models
- Characterizes computational work of HPC application
- Creates customize fine-grained DVFS policies for application
  - Intra-node: exploits application phases where CPU is stalled waiting for resources
  - Inter-node: exploits load imbalances in HPC applications

PMaC's Green Queue Framework
(fine-grained application-aware DVFS strategies)



<u>HPC System</u>
Characterize how the computational (& communication) patterns affect the overall power draw

<u>HPC Application</u>
Characterize the computational (&communication) behavior of application

Design software- and hardware-aware green optimization techniques to reduce HPC's energy footprint

PMaC's Green Queue automated framework:

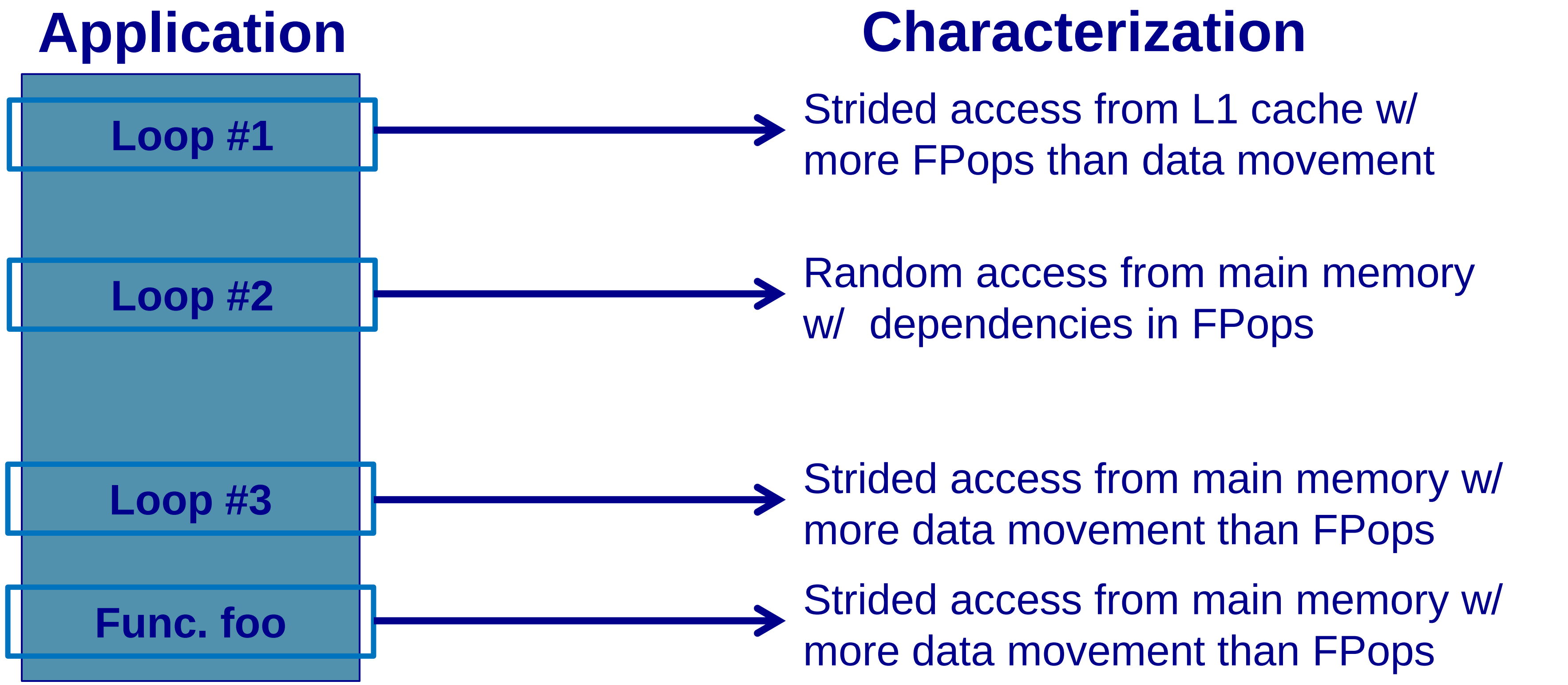
- Characterizes system's power draw behavior by running various computational work and uses to train models
- Characterizes computational work of HPC application
- Creates customize fine-grained DVFS policies for application
  - <u>Intra-node</u>: exploits application phases where CPU is stalled waiting for resources
  - <u>Inter-node</u>: exploits load imbalances in HPC applications

Application Characterization

Application characterization – fine-grained information about the communication & computation behavior of the application

– Low-level details that capture how application uses various hardware components
– Data movement on and off the processor and node
– Data locality and computational dependencies

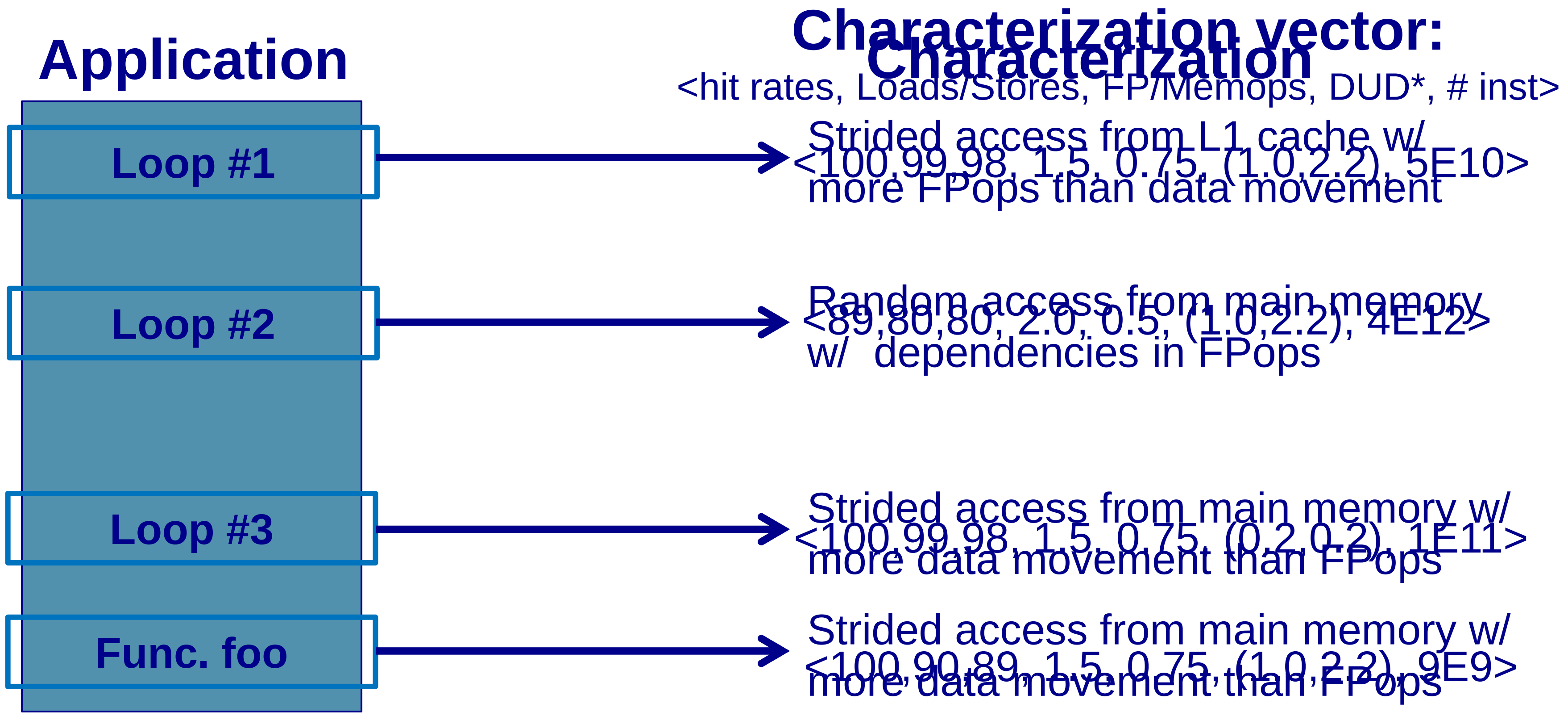
Application Characterization


Application
Characterization
Loop #1
Strided access from L1 cache w/ more FPops than data movement
Loop #2
Random access from main memory w/ dependencies in FPops
Loop #3
Strided access from main memory w/ more data movement than FPops
Func. foo
Strided access from main memory w/ more data movement than FPops

Computation characterization – collected with **PEBIL** (PMaC's Efficient Binary Instrumentor for Linux) static & dynamic analysis

Characterization vector =
<hit rates, Loads/Stores, FP/Memops, DUD*, # inst>

Application Characterization



Characterization vector:

Characterization

<hit rates, Loads/Stores, FP/Memops, DUD*, # inst>

<100,99,98, 1.5, 0.75, (1.0,2.2), 5E10>

Strided access from L1 cache w/ more FPops than data movement

<89,80,80, 2.0, 0.5, (1.0,2.2), 4E12>

Random access from main memory w/ dependencies in FPops

<100,99,98, 1.5, 0.75, (0.2,0.2), 1E11>

Strided access from main memory w/ more data movement than FPops

<100,90,89, 1.5, 0.75, (1.0,2.2), 9E9>

Strided access from main memory w/ more data movement than FPops

<u>**Computation characterization**</u> – collected with **PEBIL** (PMaC's Efficient Binary Instrumentor for Linux) static & dynamic analysis

Characterization vector =
<hit rates, Loads/Stores, FP/Memops, DUD*, # inst>

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

PMaC
Performance Modeling and Characterization

PMaC's Green Queue Framework
(fine-grained application-aware DVFS strategies)

HPC System
Characterize how the computational (& communication) patterns affect the overall power draw

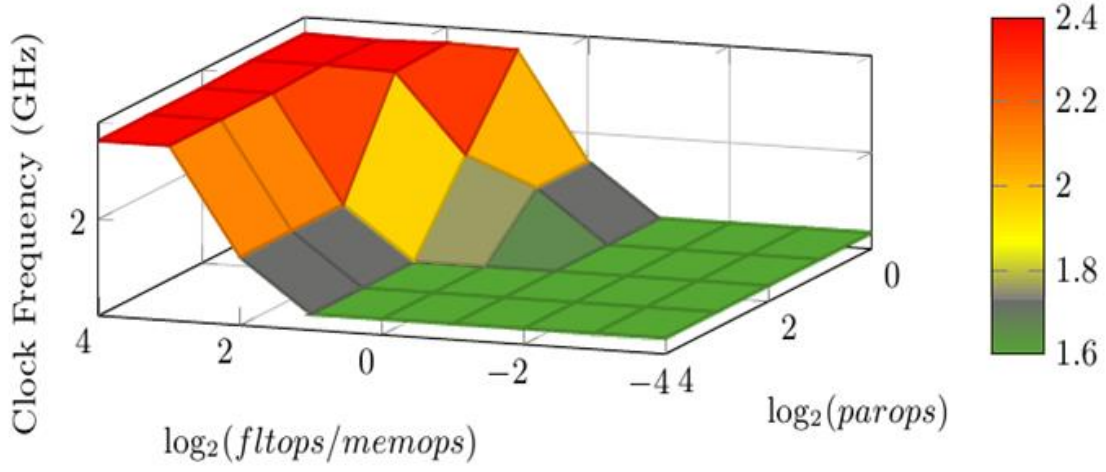
HPC Application
Characterize the computational (&communication) behavior of application

Design software- and hardware-aware green optimization techniques to reduce HPC's energy footprint

PMaC's Green Queue automated framework:

- Characterizes system's power draw behavior by running various computational work and uses to train models
- Characterizes computational work of HPC application
- Creates customize fine-grained DVFS policies for application
  - Intra-node: exploits application phases where CPU is stalled waiting for resources
  - Inter-node: exploits load imbalances in HPC applications

System Characterization



System characterization:

- Determine the most energy efficient frequency for range of computational work.
- Computational work focusing on-node.
- Computational work behavior that spans all HPC applications

Characterizing a system with PMaC's Performance & Power Benchmarking framework

PMaC's Performance Power benchmark (P^3)

- Generates computational test loops to measure performance and power for computational space of HPC application.
- Test loops measured at different frequencies
- Test loops designed to vary different characteristics of the loop (e.g. working set size or data locality)

Characterizing a system with PMaC's Performance & Power Benchmarking framework

PMaC's Performance Power benchmark (P^3)

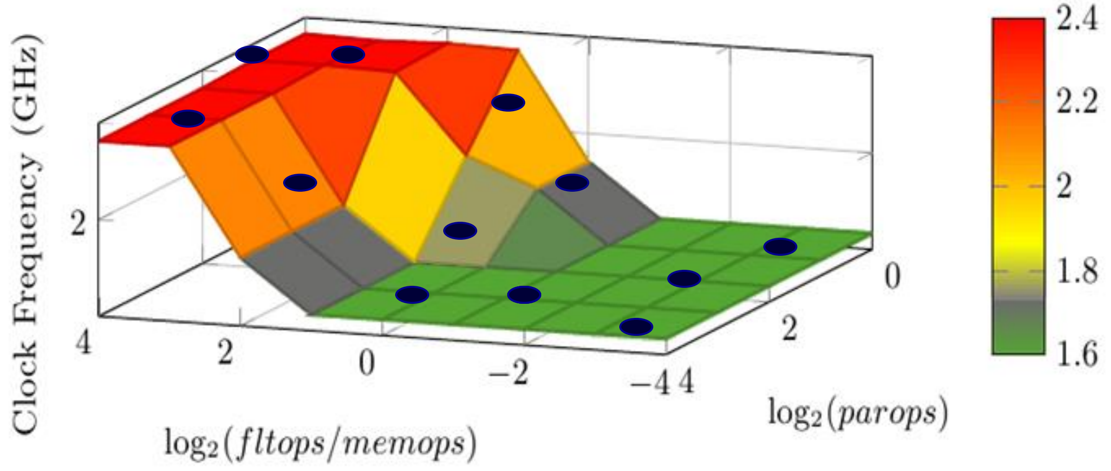
- Generates computational test loops to measure performance and power for computational space of HPC application.
- Test loops measured at different frequencies
- Test loops designed to vary different characteristics of the loop (e.g. working set size or data locality)

Testing space can grow to over 100K tests - weeks to run

Performance and Power models can save time

Power draw = func(computational behavior)

Using Performance and Power Models to fill in the Pcubed space

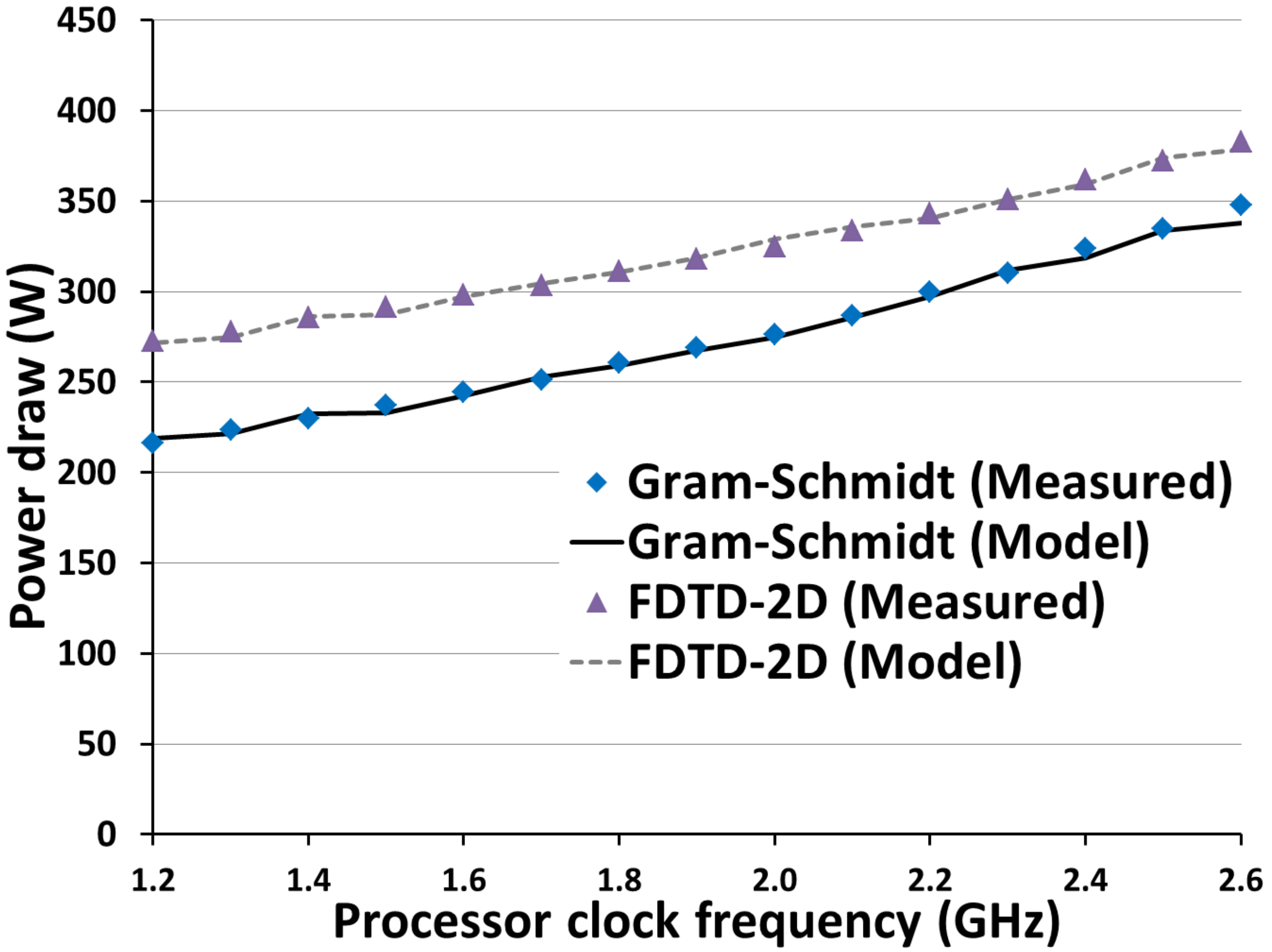


- Reduce the number of pcubed benchmark tests that we need to run: >100K → 3K
  - Reduces runtime from weeks to hours
- Use sampling of test runs to model remaining computation space.

Performance = func1(computational behavior)
Power draw = func2(computational behavior)

Power Models

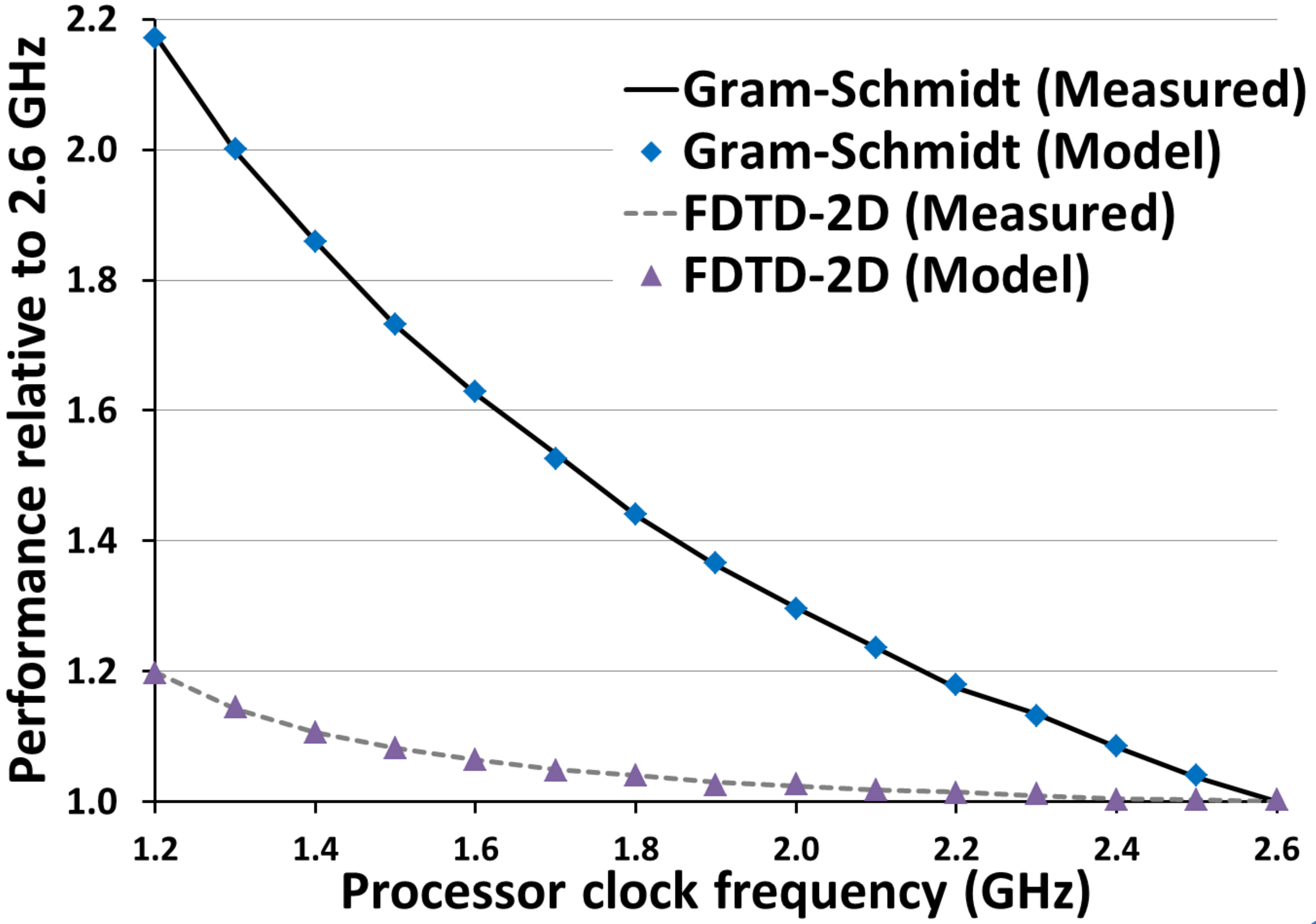
Model of power impact of frequency reduction



Power Model accuracy: 2.2% avg. absolute error on sampling of pcubed space of ~10,000 tests

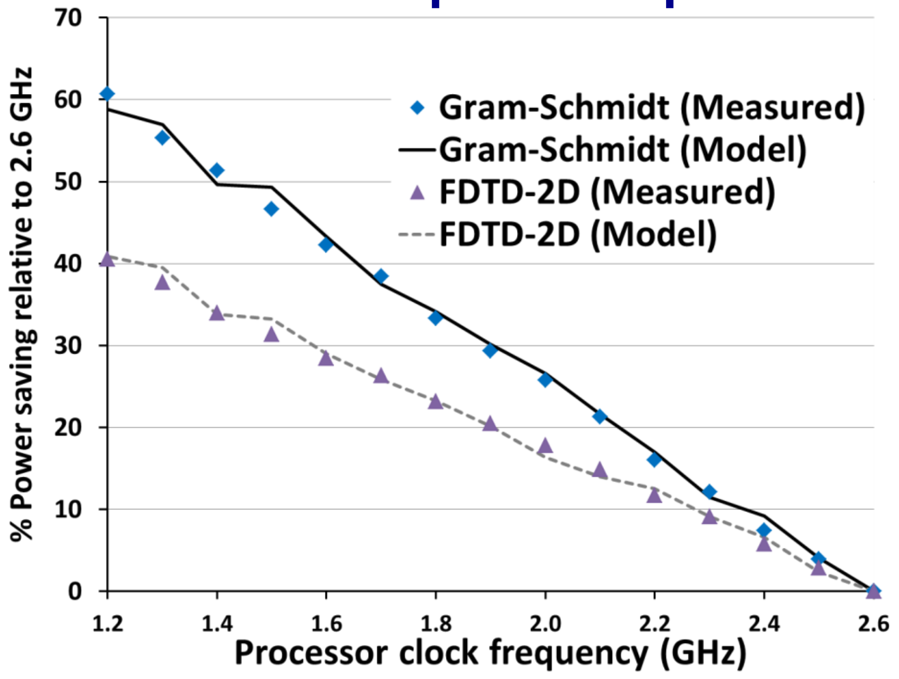
Performance Models

Model of performance impact of frequency reduction

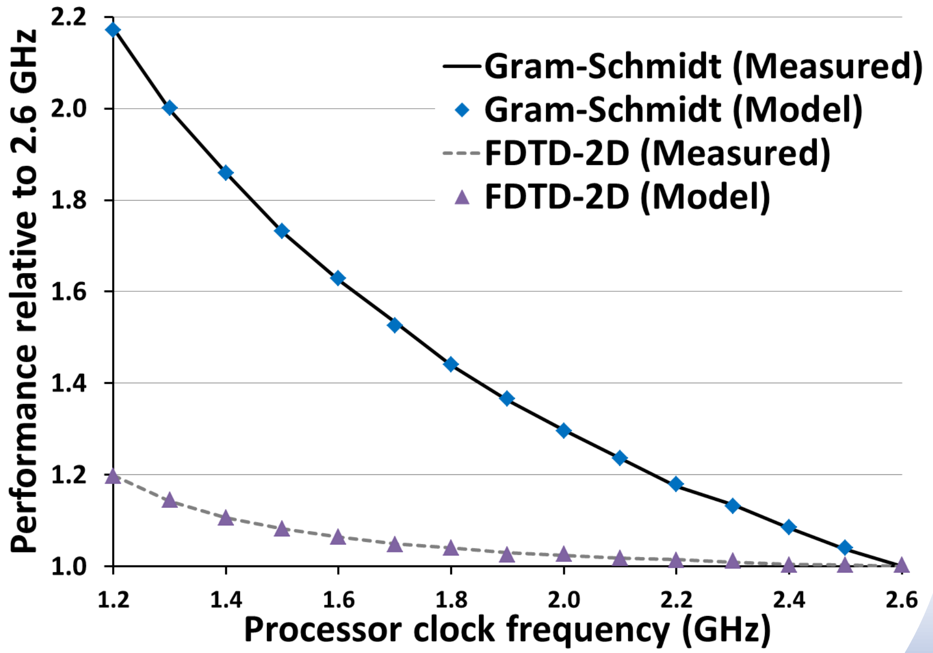


Combining Power & Performance Models for optimal energy efficiency

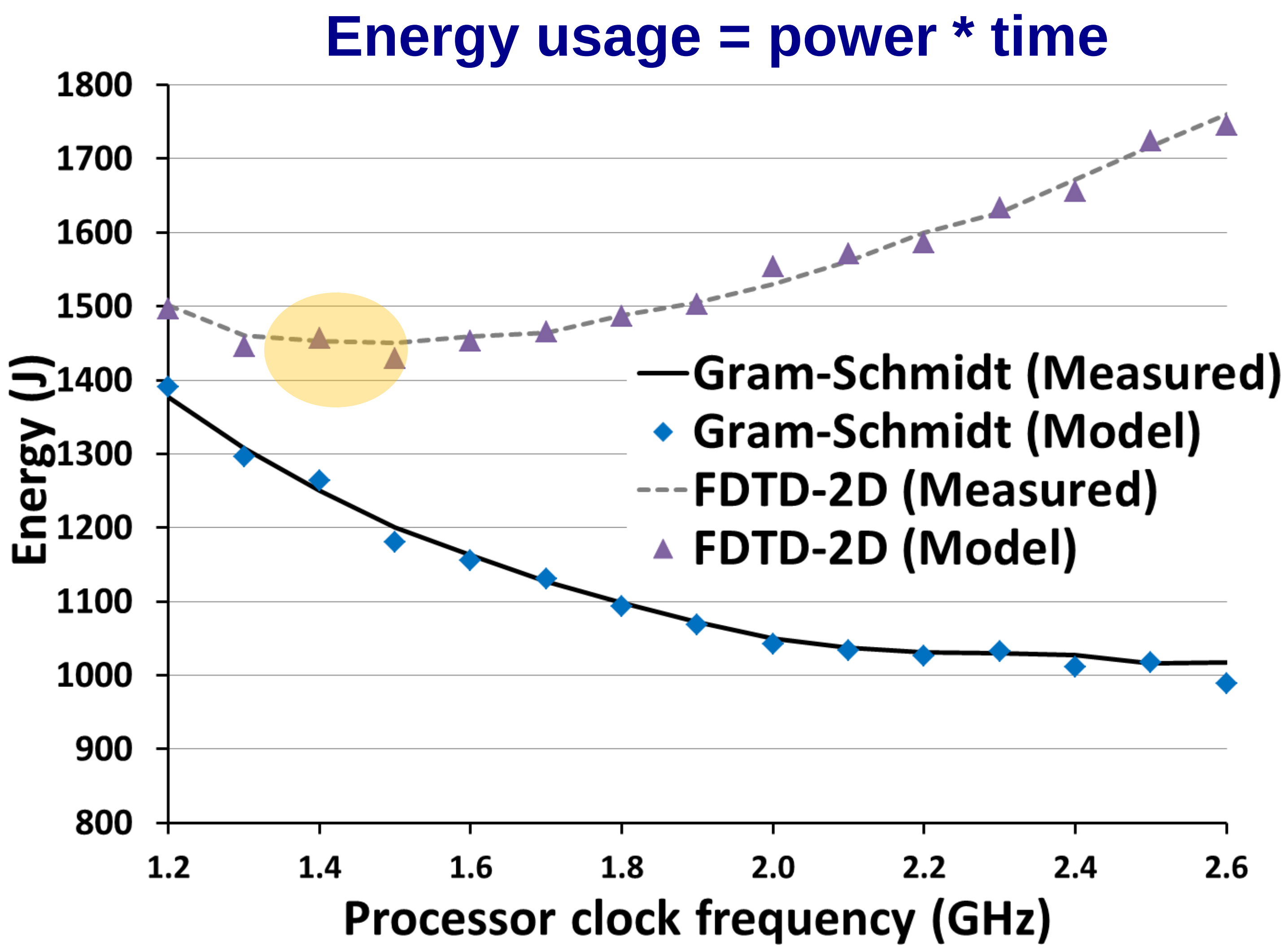
Model of power impact



Model of performance impact



Combining Power & Performance Models for optimal energy efficiency



PMaC's Green Queue Framework (fine-grained application-aware DVFS strategies)

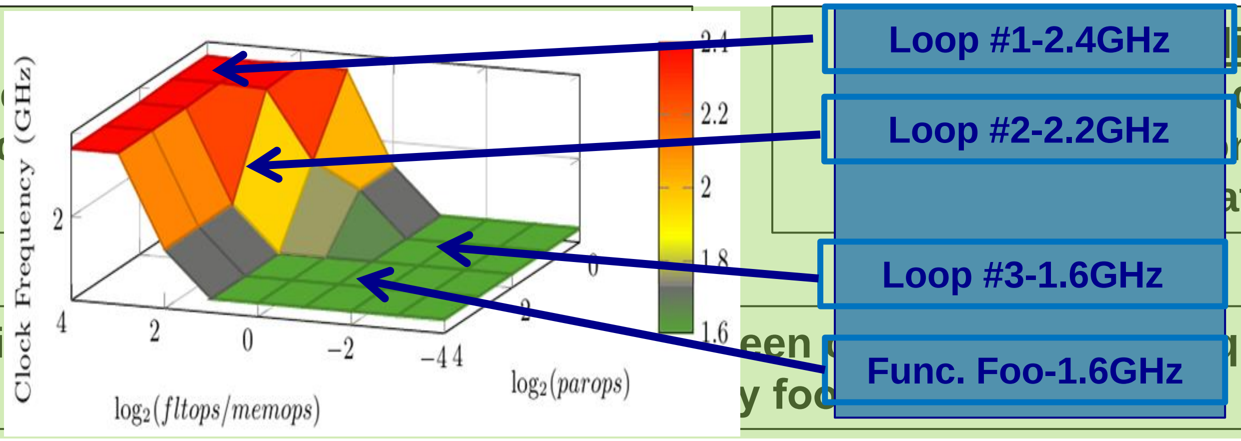
Characte
communic

lication
computational
n) behavior of
ation

Desi

een
y foc

ques to reduce



PMaC's Green Queue automated framework:

- Characterize systems power draw behavior when running various computational work using models
- Characterizes computational work of HPC application
- Creates customize fine-grained DVFS policies for application
  - Intra-node: exploits application phases where CPU is stalled waiting for resources
  - Inter-node: exploits load imbalances in HPC applications

SDSC SAN DIEGO SUPERCOMPUTER CENTER

PMaC Performance Modeling and Characterization

Intra-node Technique

(Focusing on work done on processor in between communication events)

- Memory subsystem’s performance is often the bottleneck for node-level performance
  - CPU may stall while the hardware satisfies memory requests from off-chip (e.g., L3 cache or main memory)
  - Lower the clock frequency during the phases where these stalls are significant

Intra-node Technique

(Focusing on work done on processor in between communication events)

- Memory subsystem's performance is often the bottleneck for node-level performance
  - CPU may stall while the hardware satisfies memory requests from off-chip (e.g., L3 cache or main memory)
  - Lower the clock frequency during the phases where these stalls are significant
- *Phase* is a path through the program's control flow graph which exhibits uniform runtime behavior while on that path
- Green Queue uses the **structure of the application** to identify *all phases*
  - Phase detection mechanism crosses loop and function boundaries

PMaC's Green Queue Framework (fine-grained application-aware DVFS strategies)

<u>HPC System</u>
Characterize how the computational (& communication) patterns affect the overall power draw

<u>HPC Application</u>
Characterize the computational (&communication) behavior of application

Design software- and hardware-aware green optimization techniques to reduce HPC's energy footprint

PMaC's Green Queue automated framework:

- Characterize systems power draw behavior when running various computational work using models
- Characterizes computational work of HPC application
- Creates customize fine-grained DVFS policies for application
  - <u>Intra-node</u>: exploits application phases where CPU is stalled waiting for resources
  - <u>Inter-node</u>: exploits load imbalances in HPC applications

Inter-node Technique

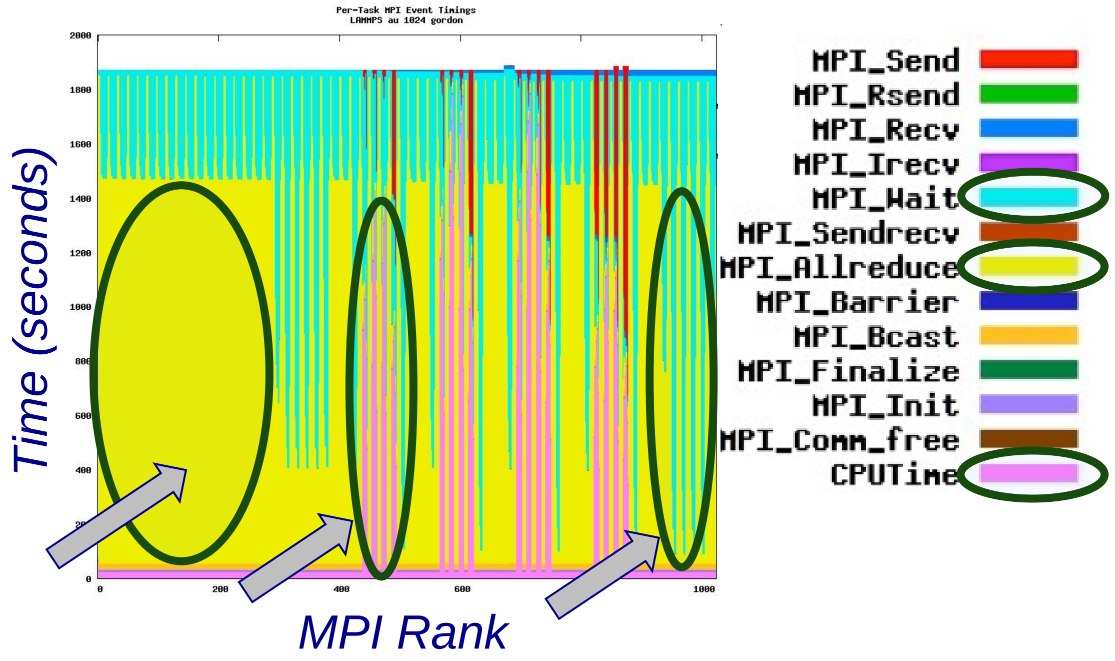
(Focusing on load imbalance in application due to work distribution)

- MPI load imbalance: a subset of MPI processes have less work to do and wait for others thereby wasting energy
  - Could arise due to inherent nature of the problem/dataset
- Large body of research on remedying load imbalance and on exploiting the same to save energy
- Green Queue’s approach is simple but we apply it at scale

Inter-node Technique

(Focusing on load imbalance in application due to work distribution)

- Green Queue captures and quantifies load imbalance by profiling all MPI communications and core-level computations



- Measure the “idleness” for each core by taking a simple ratio of its computation time to the computation time of the busiest core

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

PMaC
Performance Modeling and Characterization

Results – Experimental Setup

San Diego Supercomputer Center (SDSC) Gordon

Gordon, an Intel Sandybridge based supercomputer:

- Dual socket nodes. 8-core processor on each socket.

  (15 available clock frequencies)

- Nodes configured as a 3D torus. QDR Infiniband network

- Experiments run using a single rack of Gordon (1024 cores)
  - Not a limitation of this work
- Rack-level power measurement obtained from PDUs

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

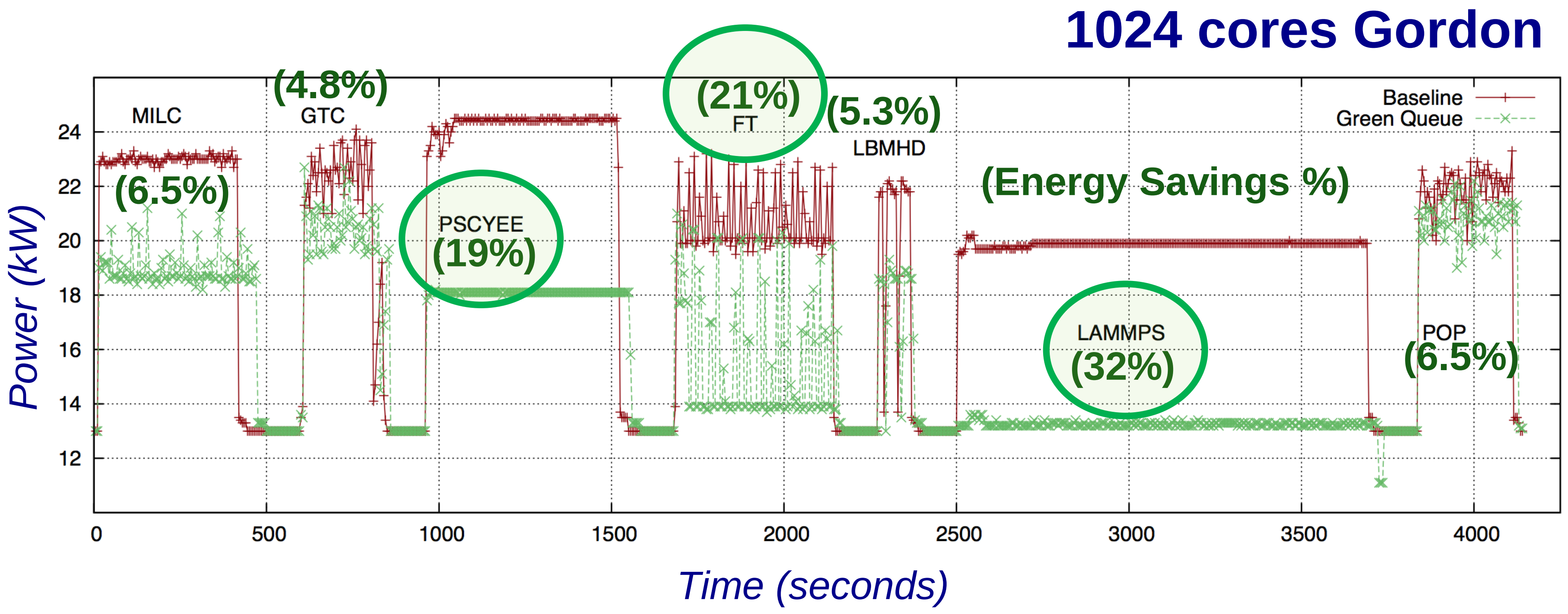
PMaC
Performance Modeling and Characterization

Results – Experimental Setup

Large scale applications and benchmarks:

Application	Description
Milc	Quantum Chromodynamics (QCD) application
GTC	Particle-in-cell application for magnetic fusion
PSCYEE	3-D Finite-difference time-domain for Maxwell equations
LBMHD	Simulation of turbulence in dissipative magnetohydrodynamics
POP	3D Ocean circulation model
LAMMPS	Molecular dynamics code
FT	Nas Parallel Benchmark kernel

Results – Overall & Discussion



- Ongoing work
  - Merge inter and intra node techniques

Contributions & Conclusions

- Phase detection based on the structure of the program
- Optimal frequency assignment for *all* phases in an application
- Framework deployed at scale on current generation supercomputer

Tiwari A, Laurenzano M, Peraza J, Carrington L, Snavely A: **Green Queue: Customized Large-scale Clock Frequency Scaling**. *CGC 2012* 2012.

Peraza J, Tiwari A, Laurenzano M, Carrington L, Snavely A: **PMaC's Green Queue: A Framework for Selecting Energy Optimal DVFS Configurations in Large Scale MPI Applications**. *Concurrency and Computation: Practice and Experience* 2012.

For details on PMaC Lab's recent energy efficiency work, please visit: http://www.sdsc.edu/pmac/

Or e-mail: lcarring@sdsc.edu

Questions ?