PARALLEL PROGRAMMING LABORATORY

Energy-efficient computing for HPC workloads on Heterogeneous Chips

Akhil Langer, Ehsan Totoni, Udatta Palekar*, Laxmikant (Sanjay) V. Kale
Parallel Programming Laboratory, Department of Computer Science
*Department of Business Administration
University of Illinois at Urbana-Champaign
http://charm.cs.uiuc.edu/research/energy

PMAM 2015
6th International Workshop on Programming Models and Applications for Multicores and Manycores
February 7-8, 2015

Outline

- ❑ Introduction
- ❑ Background
- ❑ Problem Statement
- ❑ Approach
- ❑ Results

Introduction

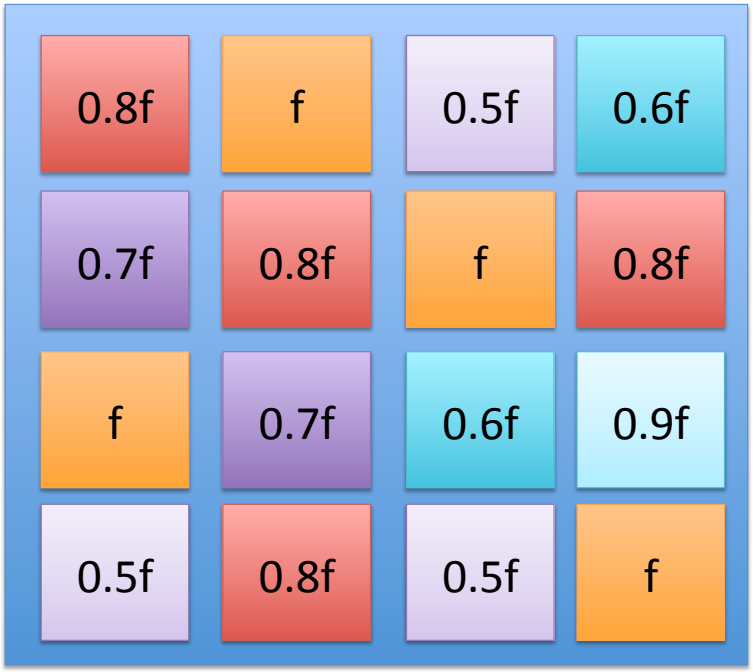
- Motivation
  - Huge energy consumption of data centers
  - 20MW power @ $0.15 per KWh, costs $2.2 M per month
  - Energy efficiency identified as a major exascale challenge by DoE
  - Consider charging users in energy units (KWh) instead (or in addition) of SUs

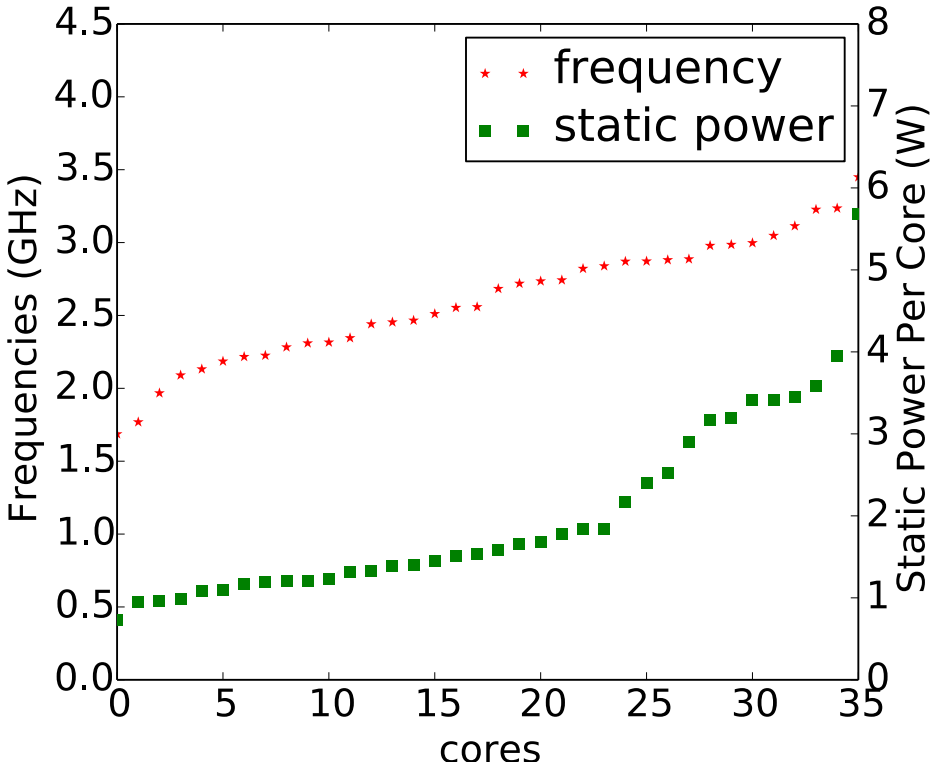
Introduction

- Low voltage operation
  - For high energy efficiency
  - For example, 10x increase in energy efficiency near threshold voltage
- But
  - Variation in CMOS manufacturing process
  - Low voltage operation introduces variability on chip
  - Cores have different frequencies and power consumption

Process Variation

- Low voltage operation





Programming Systems*

- *Problem*
  - HPC applications are highly synchronized
  - Speed determined by speed of slowest processor
- *Solution*
  - Do overdecomposition of work (e.g. Charm++)
  - Load Balance according to core speeds
- *Result*
  - Overdecomposition ratio of 16 => 2-6% load imbalance
  - No changes required in application code

*Under Review

Problem Statement

- Not optimal to use all cores on chip for execution
  - Shared resources cause contention
  - High energy consumption
- A configuration is defined as the cores on which the application is run

Determine optimal configuration that minimizes energy consumption (with optional timing constraints) of the chip for a given application

Performance Modeling*

- Exhaustive evaluation of configurations infeasible
- *Model 1*
  - Sum of individual core performance
  - Memory contention not modeled

$$S = \sum_{i \in c} s_i$$

- *Model 2*
  - Add memory access time
  - # of active cores not accounted

$$T = \frac{T_{cpu}}{\sum_{i \in c} f_i} + T_{mem}$$

*Under Review

Performance Modeling*

- Model 3
  - One model each for configurations with same number of cores
  - Performance is linear function of frequency
  - Total #cores (n) models
    - k is number of cores in configuration c
    - a_k, b_k are line constants
    - f_i is frequency of core i

$$S = a_k(\sum_{i \in c} f_i) + b_k$$

  - Average prediction error less than 1.6%
  - Dynamic power consumption can be modeled in same way

*Under Review

Energy Optimization Approach

$$min \quad \sum_{k=1}^{N} (n_k * (a_k^p \sum_i x_i f_i + b_k^p + \sum_i s_i x_i) * (a_k^t \sum_i x_i f_i + b_k^t))$$

Power

Time

Static Power

Dynamic Power

Select One Value of k

$$\sum_{k=1}^{n} n_k = 1$$

Total Number of Cores Equals k

$$\sum_{i=0}^{n-1} x_i = \sum_{k=1}^{n} n_k k$$

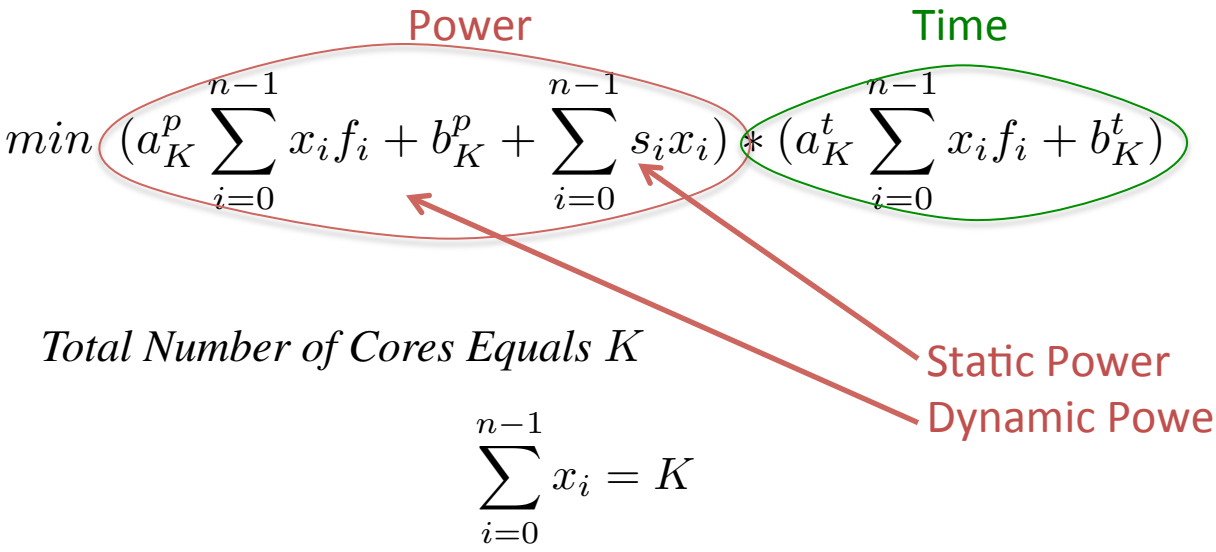
Cubic Objective Function!

Variables Range

$$\forall i \in [0, n), \quad x_i \in \{0, 1\}$$
$$\forall k \in (0, n], \quad n_k \in \{0, 1\}$$

Energy Optimization Approach

- Convert cubic program to n quadratic programs
- Each corresponding to all configurations with fixed number of cores
- Select best configuration across n quadratic programs



Quadratic Objective Function!

Energy Optimization Approach

- Quadratic programs hard to solve using non-linear methods
- Replace quadratic terms of form x_1x_2 with binary variables y_{12} and add following constraints

$$y_{12} \leq x_1$$

$$y_{12} \leq x_2$$

$$y_{12} \geq x_1+x_2-1$$

- Add timing constraint

$$a_K^t F + b_K^t \leq Pt_{min},$$

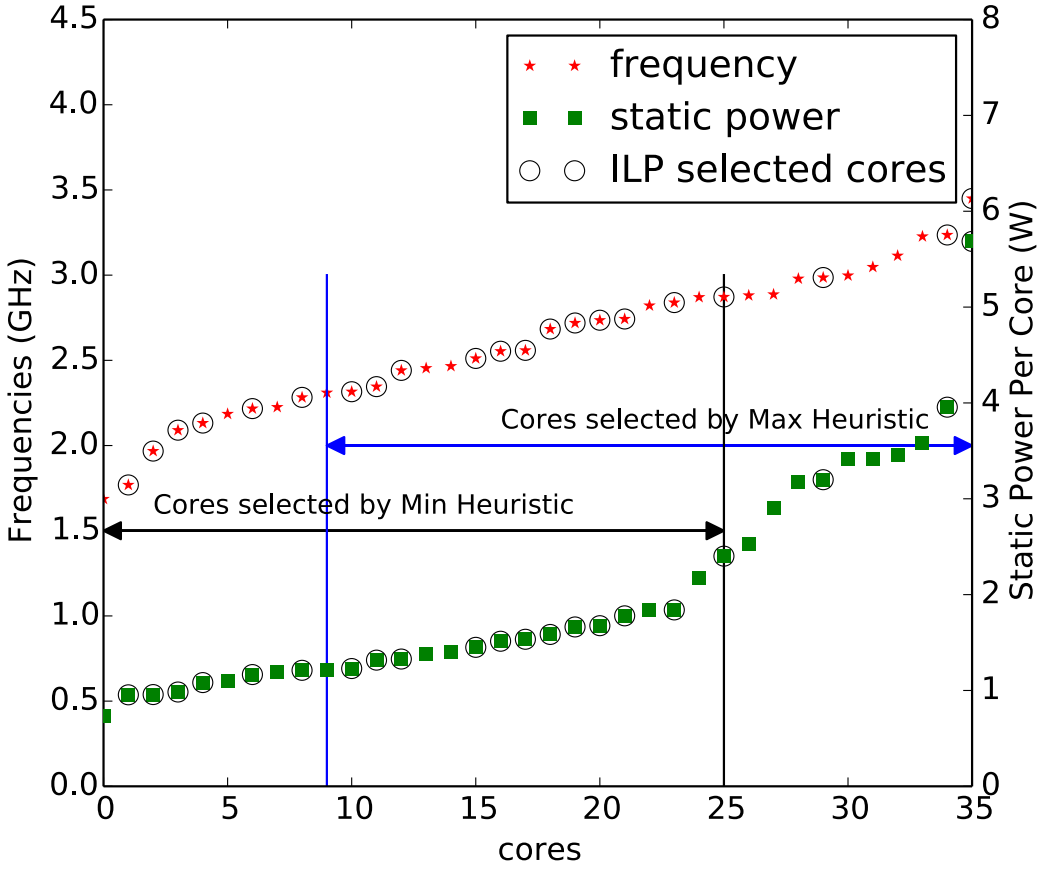
where F is sum of frequencies,

and P is allowed time penalty

Setup

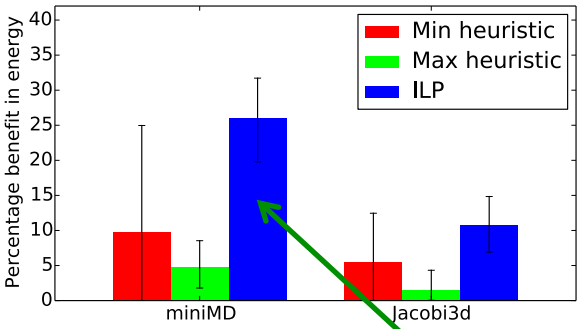
- Sniper Simulator
  - $V_{dd} = 0.765V$
  - 36 cores on chip
  - Results across 25 chips
- Applications
  - miniMD
    - Molecular dynamics mini application
    - Computationally intensive
  - Jacobi
    - 3D stencil code
    - Memory intensive
- Heuristics
  - Min heuristic
  - Max heuristic
- Integer Linear Program (ILP) Solver
  - Gurobi
  - Uses variant of branch-and-bound method

Results

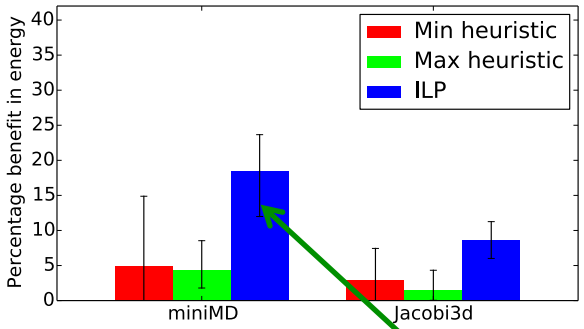

frequency
static power
ILP selected cores
Cores selected by Max Heuristic
Cores selected by Min Heuristic
Frequencies (GHz)
Static Power Per Core (W)
cores
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
0
1
2
3
4
5
6
7
8
0
5
10
15
20
25
30
35

Results

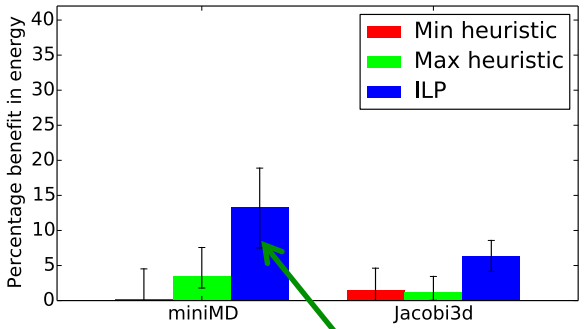
Energy Savings



(a) With no time constraint



(b) Maximum 15% time penalty



(c) Maximum 5% time penalty

26%

18.4%

13.4%

ILP Solution Time:

745 seconds *26 seconds* *9seconds*

vs

Exhaustive Evaluation: 74 hours

Conclusions

- Negligible overhead
  - O(n) samples required
  - Performance models developed with negligible overhead
- ILP solvers to optimize energy consumption with timing constraints
  - Significant energy savings as compared to sub-optimal heuristics
- No extra compute resources required
  - Solve ILPs on respective chips prior to job execution

Future Work

- Further improvement of performance models
- Evaluate approach with even larger number of cores
- Optimization methods to further improve solution time
- Apply to other HPC applications

PARALLEL PROGRAMMING LABORATORY

QUESTIONS!

http://charm.cs.uiuc.edu/research/energy

Energy-efficient computing for HPC workloads on Heterogeneous Chips

Akhil Langer, Ehsan Totoni, Udatta Palekar*, Laxmikant V. Kale
Parallel Programming Laboratory, Department of Computer Science
*Department of Business Administration
University of Illinois at Urbana-Champaign

PMAM 2015
6th International Workshop on Programming Models and Applications for Multicores and Manycores
February 7-8, 2015
San Francisco Bay Area, USA