Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS

Author: Lizhe Wang, Gregor von Laszewski, Jai dayal

Speaker: Jong Youl Cho

Community Grids Lab, Indian University

Outlook

- Background
- Problem definition
- Proposed algorithm
- Evaluation
- Conclusion

Background

- Parallel task scheduling
  - Static scheduling
  - Dynamic scheduling
- Dynamic voltage and frequency scaling (DVFS)
- Power aware task scheduling with DVFS

DVFS model

$$V = \bigcup_{1 \leq m \leq M} \{v_m\} \tag{1}$$

$$F = \bigcup_{1 \leq m \leq M} \{f_m\} \tag{2}$$

where,
v_m is the m-th processor operating voltage;
f_m is the m-th processor operating frequency;
$v_{min} = v_1 \leq v_2 \leq ... \leq v_M = v_{max}$;
$f_{min} = f_1 \leq f_2 \leq ... \leq f_M = f_{max}$;
$1 \leq m \leq M$, M is the total number of processor operating points.

Energy model

The energy consumption

$$\xi = \sum_{\triangle t} (\delta \cdot v^2 \cdot f \cdot \triangle t) \tag{8}$$

Where,
δ is a constant determined by the PE.
v is the processor operating voltage during $\triangle t$;

f is the processor operating frequency during $\triangle t$;
$\triangle t$ is a time period.

Cluster model

- $pe_k.v^{op} \in V$ is the processor operating voltage
- $pe_k.f^{op} \in F$ is the processor operating frequency

$1 \leq k \leq K$, K is the total number of PEs.

A cluster C is defined by its set of processing elements

$$C = \bigcup_{1 \leq k \leq K} \{pe_k\} \tag{9}$$

Job model

DAG model: T= (J, E)

$$J = \bigcup_{1 \le n \le N} \{job_n\} \tag{10}$$

A job, job_n, has 3 properties:

- $weight$ is the instruction number of job_n.
- t^{st} is the starting time of job_n.
- t is the execution time of job_n. if job_n is executed on pe_k, the job execution time is calculated as follows:

$$job_n.t = \frac{job_n.weight \cdot CPI}{pe_k.f^{op}} \tag{11}$$

Job model

- E: a set of precedence constraints (edges in a DAG)
  E defines partial orders (operational precedence constraints) on J. e_{ij} is an edge between job_i and job_j, it means that job_i must be completed before job_j can begin, $1 \leq i, j \leq N$, $job_i, job_j \in J$. e_{ij} sometime can also be represented $job_i < job_j$.
  e has one property:
  $e_{ij}.cost \geq 0$, is the amount of data required to be transferred from job_i to job_j, $1 \leq i, j \leq N$, $job_i, job_j \in J$. Data are transferred from the PE where job_i is executed to the PE where job_j is executed.

Problem definition (1)

- Problem 1: Best-effort scheduling
  - Schedule parallel tasks to a cluster
  - Minimize the makespan
  - Reduce energy consumption without increasing the makespan

Problem definition (2)

- energy-performance tradeoff scheduling
  - Users can adopt some performance loss, for example, increase the makespan
  - Schedule tasks to a cluster, minimize the energy consumption

Best Effort Scheduling Algorithm (1)

- schedule tasks via the ETF scheduling algorithm
- scale down PE’s voltages for all non-critical jobs

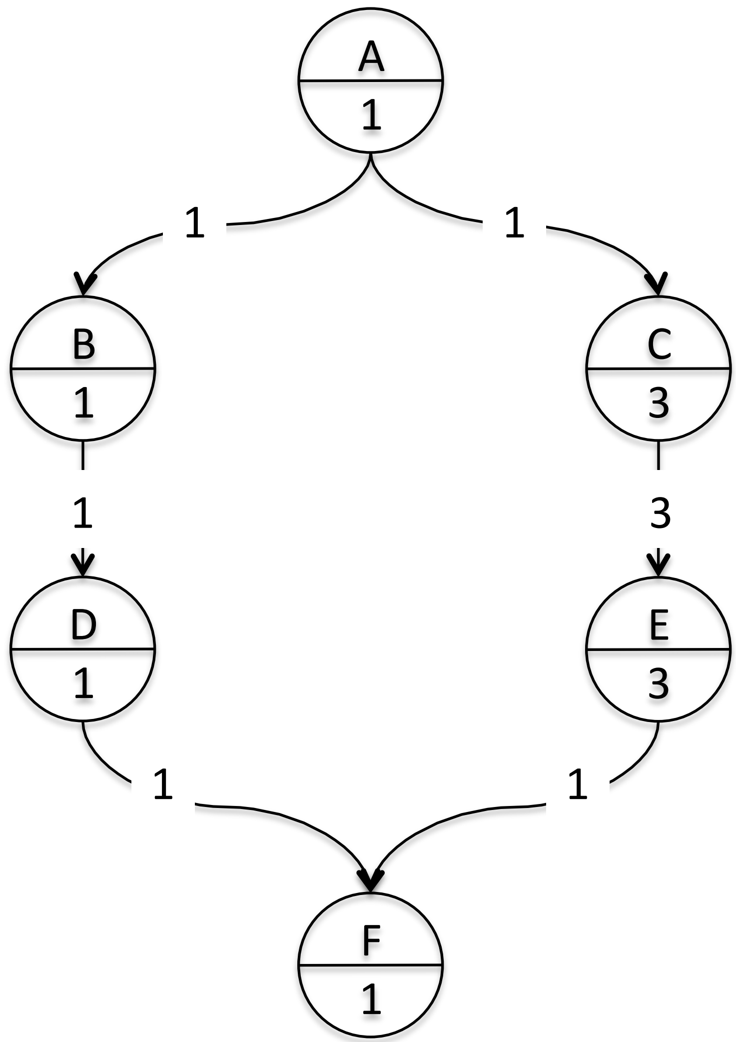
ETF scheduling algorithm

- ETF: Early task first algorithm
- Compute priorities for all tasks
  - Currently we use b_level, which is the long length from a task to the exist node
- Sort all tasks
- Put tasks that ready to execute in the ready queue with task priority
- Select the first task from ready queue
- Select a resource for this task, so as to give the earliest task finish time
- Loop this scheduling till all tasks are scheduled

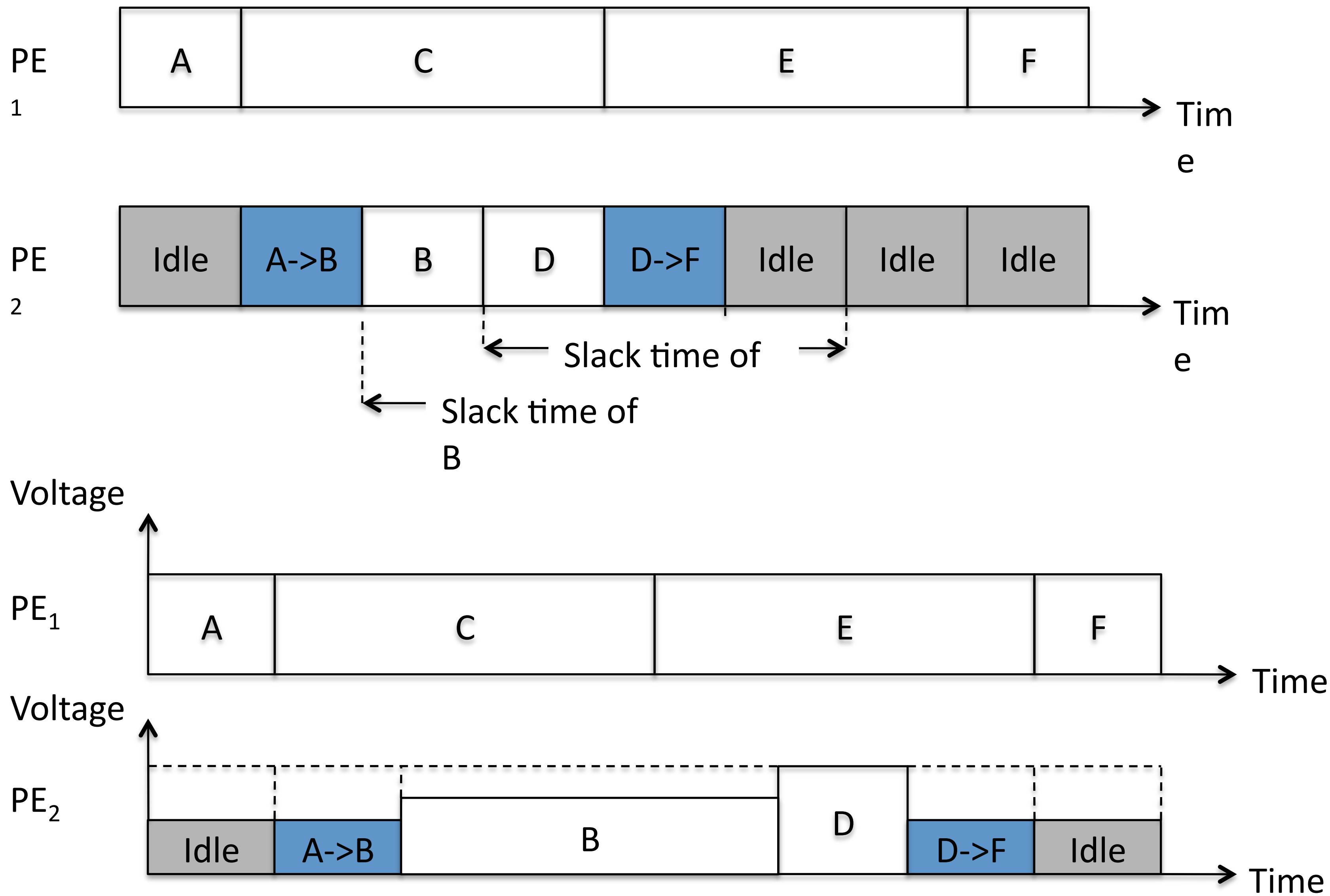
Scale down non-critical tasks

- for all PEs
  - for all time slots in this PE
    - If this time slot executes a communication or this time slot is idle
    - Then scale down the voltage of this PE in this time slot
    - If this time slot execute a non-critical task
    - Then scale down the voltage of this PE in this time slot

Example DAG



Gantt Chart

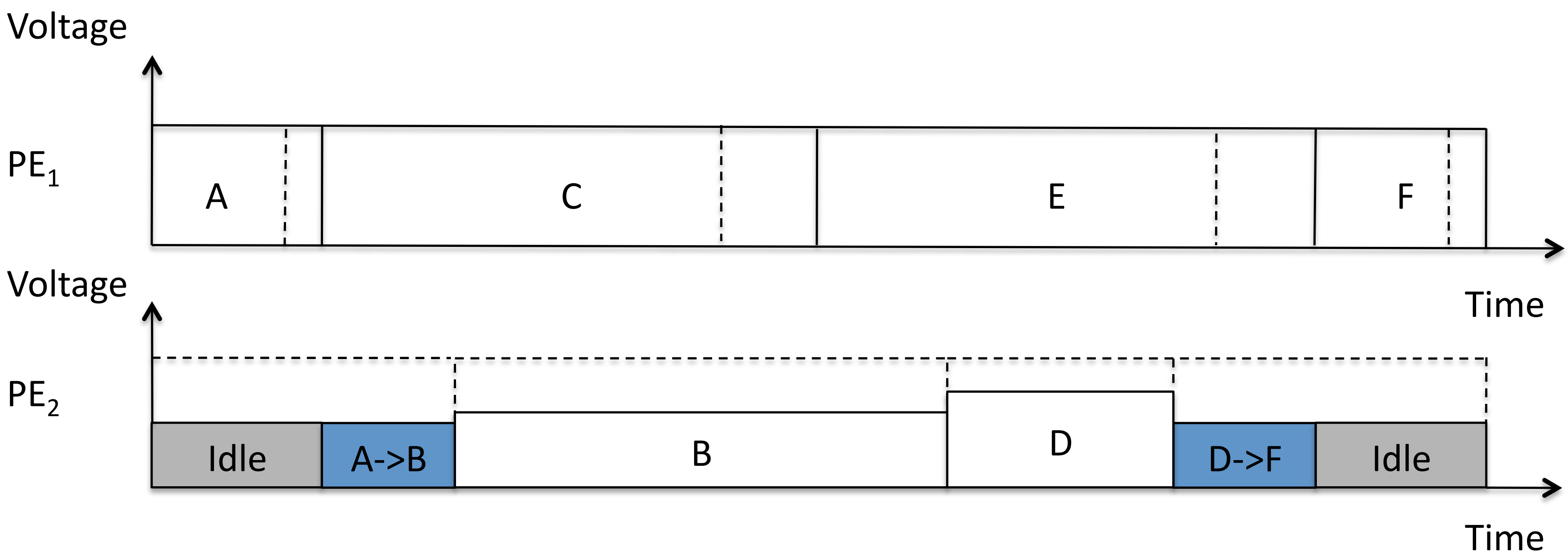


PE
1
A
C
E
F
Tim
e
PE
2
Idle
A->B
B
D
D->F
Idle
Idle
Idle
Tim
e
Slack time of
Slack time of
B
Voltage
PE$_1$
A
C
E
F
Time
Voltage
PE$_2$
Idle
A->B
B
D
D->F
Idle
Time

Energy-performance tradeoff scheduling algorithm

- Execute Early task first algorithm (ETF)
- Scale down PE's voltages for critical tasks with the predefined acceptable performance loss rate.
- Scale down PE's voltages for non-critical jobs

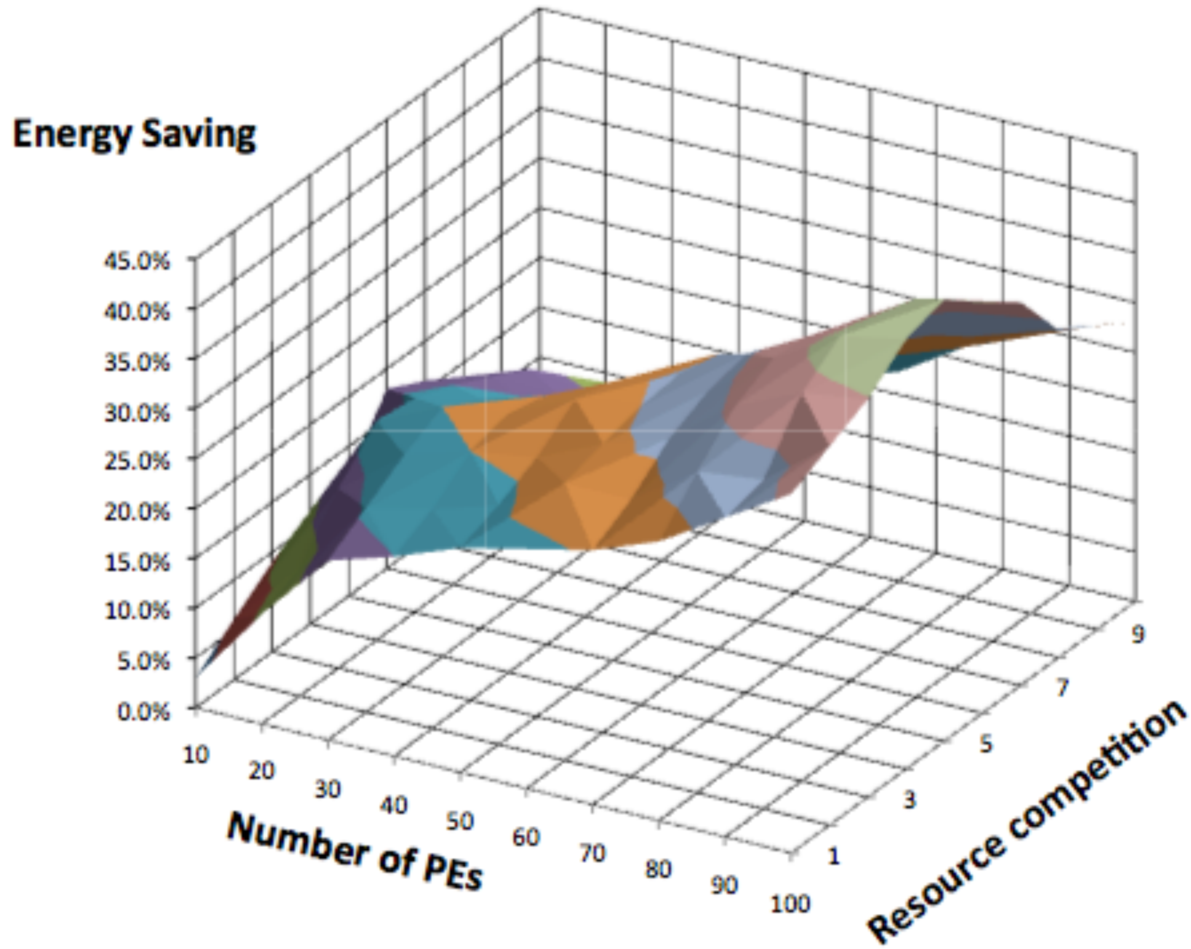
Example



Evaluation (1)

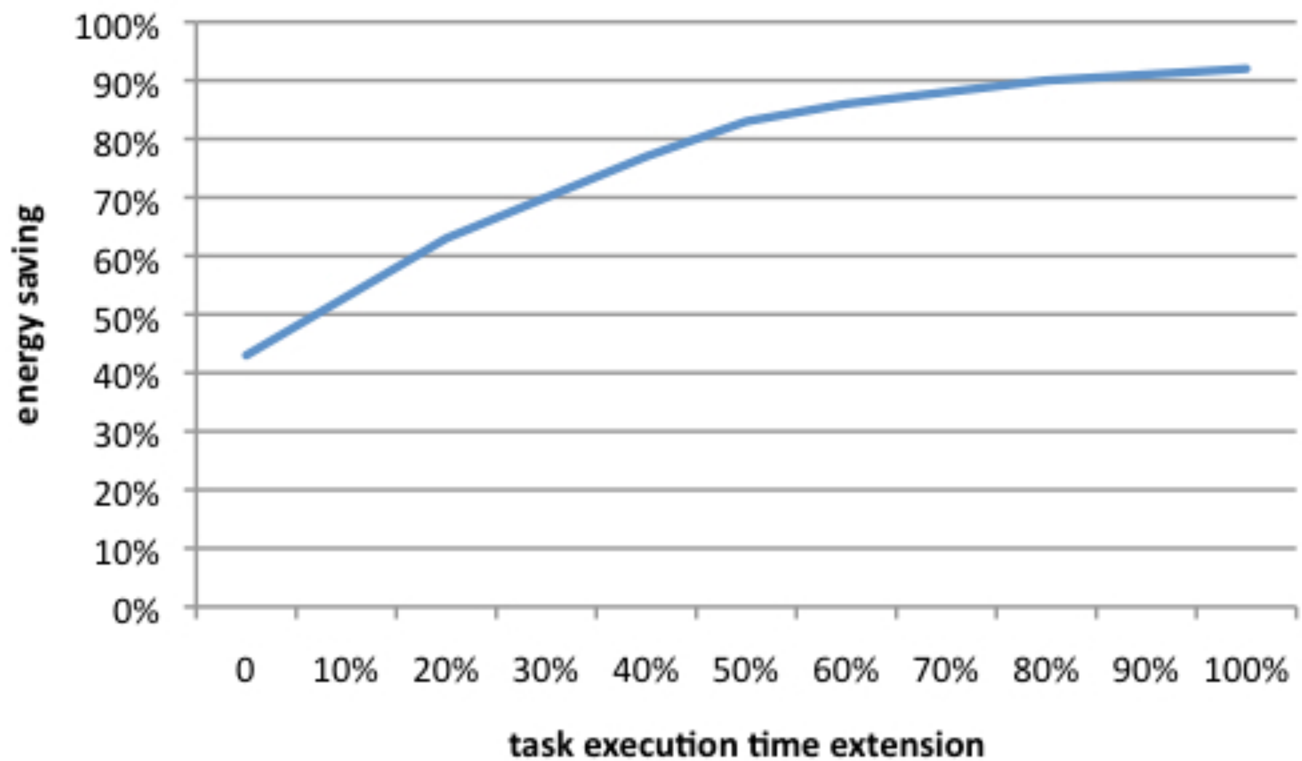
- Simulation study:
  - MT43 processor
  - Use synthetic DAG generation tool
- Results

Energy aware DAG scheduling algorithm	Maximum energy saving
EADUS & TEBUS [28]	16.8%
Energy Reduction Algorithm [31]	25%
LEneS [22]	28%
ECS [30]	38%
Our algorithm	44.3%



Energy Saving
45.0%
40.0%
35.0%
30.0%
25.0%
20.0%
15.0%
10.0%
5.0%
0.0%
10
20
30
40
50
60
70
80
90
100
Number of PEs
1
3
5
7
9
Resource competition

Result(2)



100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
energy saving
0
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
task execution time extension

Conclusion and future work

- We study energy aware cluster scheduling algorithms
- Two research issues are studied
  - Best-effort scheduling issue
  - Energy-performance tradeoff issue
- We proposed two algorithms
- Future work
    - Workload characterization
    - Runtime support and implementation