Energy-Efficient Scheduling Algorithm of Task Dependent Graph on DVS-Unable Cluster System

Yan Ma, Bin Gong, Lida Zou

Shandong University

山东省高性能计算中心
Shandong High Performance Computing Center

Outline

- Preliminary Knowledge
- Related Work
- Our System Model
- Our Scheduling Algorithm
- Case Study and Experimental Result
- Conclusion and Future Work

Preliminary Knowledge-1

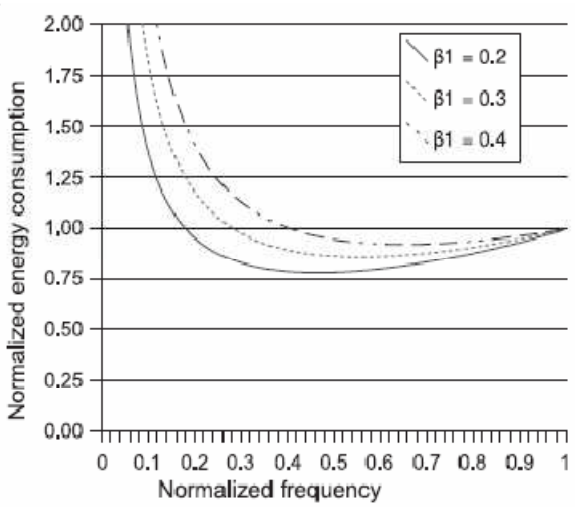
- Power Management in HPC need researching?
  - High operation cost: Shanghai Supercomputing Center
  - Low reliability: Pflops system; temperature effect
  - Environmental effect: Carbon dioxide; chip miniature

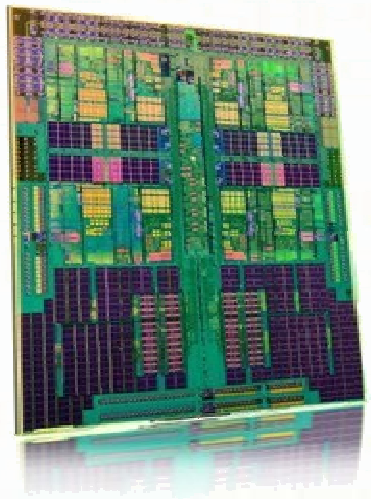
Preliminary Knowledge-2

- Current power-saving measures
  - choose products whose performance match their power
  - install low-power cooling devices and heat sinks
  - enhance equipment maintenance
- Drawback? →power-aware software based method
- Power-aware task scheduling algorithm
  - Wireless sensor network, embedded system
  - Scheduling objective:
    - save energy with no (little) performance loss

Preliminary Knowledge-3

- System-level power saving techniques
  - Dynamic Voltage Scaling (DVS)
    - Reduce supply voltage of PE
    - power will generate square-reduction when supply voltage decreases in CMOS circuit.
  - Dynamic Power Management (DPM): selectively turn off system components when they are idle.





Outline

- Preliminary Knowledge
- Related Work
- Our System Model
- Our Scheduling Algorithm
- Case Study and Experimental Result
- Conclusion and Future Work

Related Work-1

Authors and papers	Objective	Application	Environment	Methods
Kim et al. [8]	Energy optimization with specified deadline	Bag of tasks	Cluster	DVS for processing elements
Freeh et al. [9,10,11]	Energy optimization	MPI program	HPC	DVS for non-critical node
Qin et al. [12,13,14]	Energy efficient without performance loss	Parallel application	Cluster	DVS or Task duplication
Cameron et al. [15,16,17,18]	Phase based energy efficient	Scientific application	Cluster	DVS

Related Work-2

- Analysis of previous work
  - Unavailability of DVS technique
  - Inadaptability of data-intensive application
  - Neglect of static power in PE
- Our objective
  - Application: data-intensive, DAG
  - System: DVS-Unable cluster system
  - Objective: minimize the execution time
  - Constraint: maintain low energy consumption

Outline

- Preliminary Knowledge
- Related Work
- Our System Model
- Our Scheduling Algorithm
- Case Study and Experimental Result
- Conclusion and Future Work

Dependent Task Model

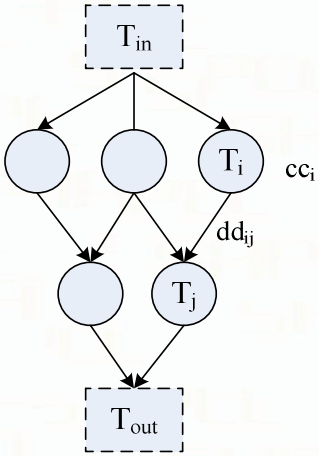


Fig.1. Dependent task model

DVS-Unable Cluster System

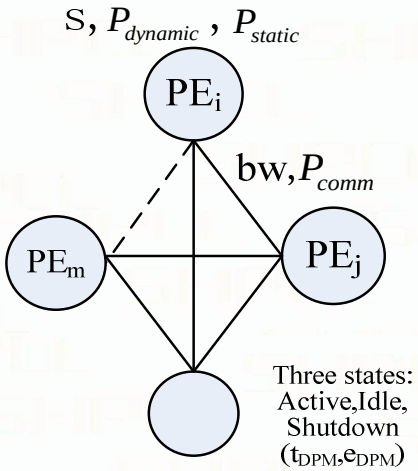


Fig.2. DVS-Unable Cluster System

Power Model

- Computation energy, executing task T_i :

  $E^i_{comp} = (P_{static} + P_{dynamic}) \cdot t_i = (P_{static} + P_{dynamic}) \cdot cc_i / s$

- Communication energy, PE_i transfers data to PE_j

  $E_{comm} = P_{comm} * dd_{ij}$

- Note:
  - Pure communication time: t_{comm}
  - Shutdown cost: t_{DPM} & e_{DPM}
    - Algorithm must check if the time and energy saving after DPM can compensate the cost of executing DPM.

Outline

- Preliminary Knowledge
- Related Work
- Our System Model
- Our Scheduling Algorithm
- Case Study and Experimental Result
- Conclusion and Future Work

Energy-efficient Scheduling Framework

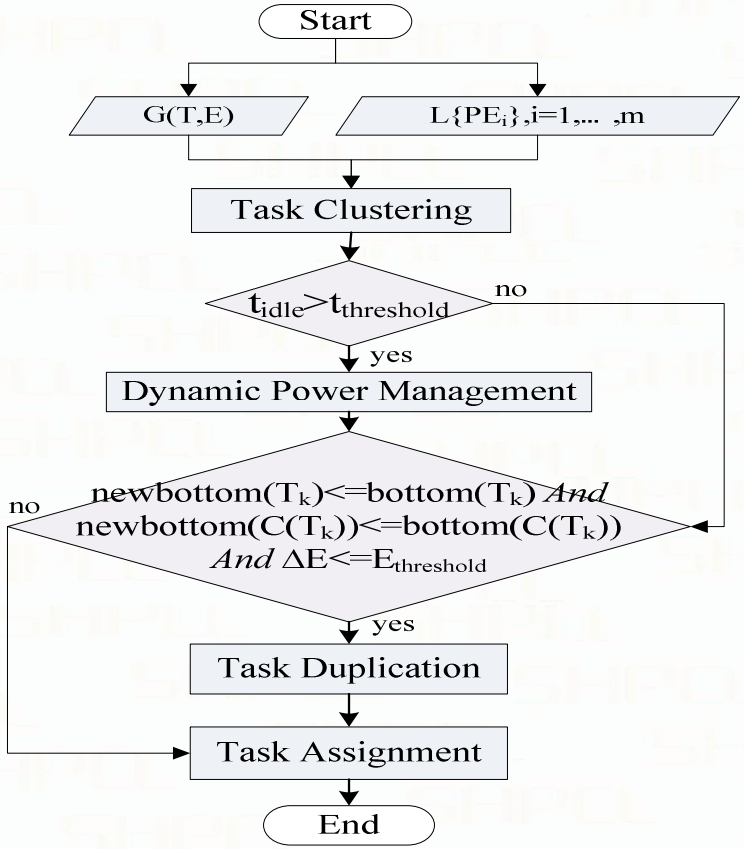


Fig.3. The energy-efficient scheduling framework of task dependency graph on DVS-Unable cluster system

Task Clustering (TC)

- Task clustering is to assign all the tasks in a cluster to the same resource, and aims to reduce data transfer cost between tasks.
- CASS-II algorithm [23]
  - Firstly, it computes parameter top_i for each task according to formula (1)
  - Secondly, it computes another parameter $bottom_j$ for each task from bottom to up according to formula (2)
  - Lastly, in the bottom computation process, it selects the task whose *priority=top+bottom* value is the largest to try to merge with its dominant successor into a cluster:
    - If the *bottom* values of current task and all the tasks in the cluster do not increase, the actual merger can be performed
    - else, the current task becomes a separate cluster.

$$top_i = \begin{cases} 0 \text{.....} T_i \text{is.entry.node} \\ \max\{top_j + t_j + t_{ji}, (j,i) \in E\} \text{.....otherwise} \end{cases} \quad ...(1)$$

$$bottom_j = \begin{cases} t_j \text{.....} T_j \text{is.exit.task} \\ \max\{bottom_i + t_{ji} + t_j, (j,i) \in E\} \text{.....otherwise} \end{cases} \quad ...(2)$$

Dynamic Power Management (DPM)

- Steps of DPM
  - only when idle period exceeds a certain threshold can performing DPM method reduce time and energy.

    $t_{threshold} = \max\{t_{DPM}, e_{DPM} / P_{static}\}$
  - If $t_{idle} > t_{threshold}$, the processor can be turned off in the idle period.
  - Else, the processor does not shut down.

Task Duplication (TD)-1

- Task duplication is to use the idle time of resources to duplicate task running in another resource
- 1) Select duplicated candidate tasks:
  - find dominant predecessor for each task
  - dominant predecessor not in its cluster, become candidate task.
- 2) Make an attempt to duplicate candidate tasks
  - Duplication conditions
    - *bottom values* of duplicated task and all the tasks in the cluster not increase
    - *increased energy* not exceed a given threshold, $\Delta E = E_2 - E_1 \leq E_{threshold}$
  - In the process of computing *E2*, if the new idle time exceeds the time threshold, it need call DPM function.

Task Duplication (TD)-2

- Computation of ΔE
  - If it cannot execute DPM in the idle period caused by duplication, then the energy variability caused by duplication is $\Delta E^1 = P_{dynamic} \cdot t_j - P_{comm} \cdot dd_{ji}$
  - If it can execute DPM in the idle period caused by duplication, then the energy variability caused by duplication is $\Delta E^2 = P_{dynamic} \cdot t_j - P_{comm} \cdot dd_{ji} - \Delta E_{DPM}$

Task Assignment (TA)

- Assign each cluster to a processor
- Note:
  - It applies for corresponding number of processors to cluster system according to the result of task clustering.
  - Execution is non-preemptive and predictive
  - The application has the right of executing duplication and DPM technique for applied processors

Outline

- Preliminary Knowledge
- Related Work
- System Model
- Scheduling Algorithm
- Case Study and Experimental Result
- Conclusion and Future Work

Case Study

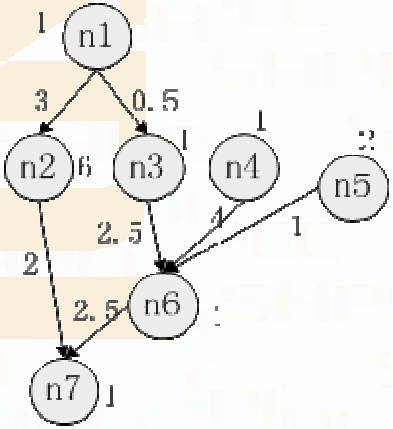


Fig.3. A instance of directed acyclic graph

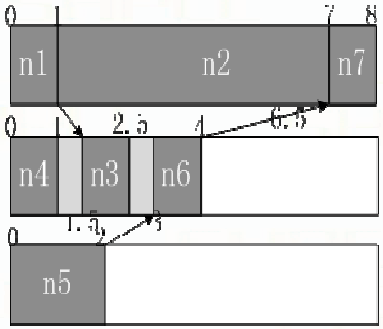


Fig.4. Task clustering result of instance in Figure 3

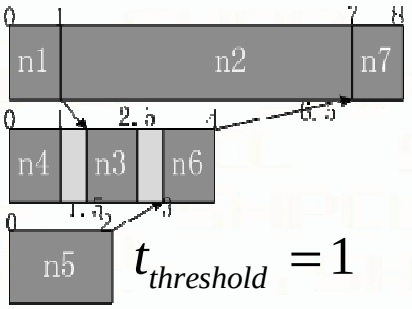


Fig.5. Result of executing DPM of instance in Figure 3

Task	DP
n1	-
n2	n1
n3	n1
n4	-
n5	-
n6	n3,n4
n7	n2

- n1 is candidate task of {n6, n3, n4}
- Make an attempt of duplication
- Check two conditions

1. First, the bottom value of n1 is *newbottom(n1)=6.5+1=7.5 < bottom(n1)=8* ;
2. Second, compute ΔE

no idle time, no call DPM

$$\Delta E = \Delta E^1 = 60*1 - 1.5*0.5*100 = -15 Joul <$$

$$E_{threshold} = 25 Joul$$

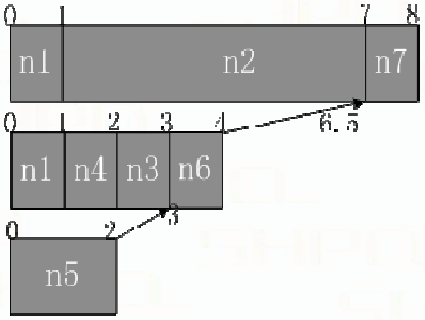
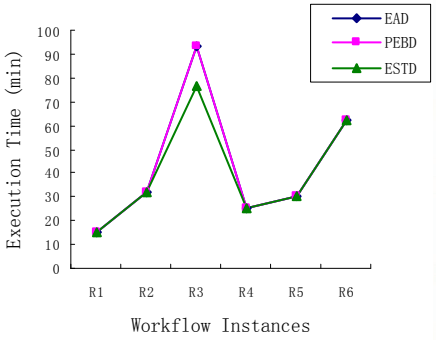
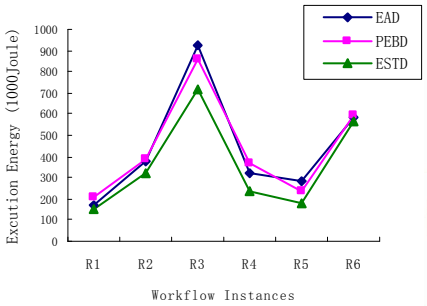


Fig.6. Task duplication result of instance in Figure 3

Experimental Result

- Comparison objects
  - EAD: last(i)-lact(j)<cj,i, threshold
  - PEBD: last(i)-lact(j)<cj,i, cost ratio
  - ESTD: bottom value, threshold, static
- Two kinds of applications
  - Synthetic workflow
  - Real application workflow
- Evaluation Parameters
  - Execution time
  - Energy consumption
- Analysis
  - last(i)-lact(j)<cj,i: ignore the pure communication time between tasks
  - Cost ratio: easier to duplicate
  - Ignore static power





Outline

- Preliminary Knowledge
- Related Work
- System Model
- Scheduling Algorithm
- Case Study and Experimental Result
- Conclusion and Future Work

Conclusion and Future Work

- Conclusion
  - Provide a novel energy-efficient scheduling framework for high-communication applications in DVS-Unable HPC environment.
  - Focus on the communication energy and static energy of PE.
  - Consider complex priority constraints, data transmission and the conflict of multiple scheduling indicators.
- Future Work
  - Extend to heterogeneous system
  - Research dynamic and adaptive algorithm

Thank you

Q & A