Dynamic Scheduling of Imprecise-Computation Tasks in Maximizing QoS under Energy Constraints

Heng Yu, Bharadwaj Veeravalli, Yajun Ha

Department of Electrical & Computer Engineering,
National University of Singapore

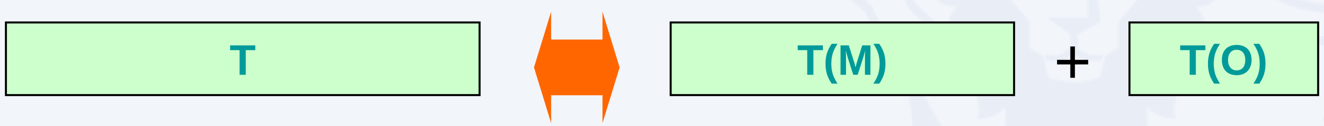
Outline

- Introduction
- The GCS scheduling algorithm
- Experimental results
- Conclusions

The Imprecise-Computation (IC) model

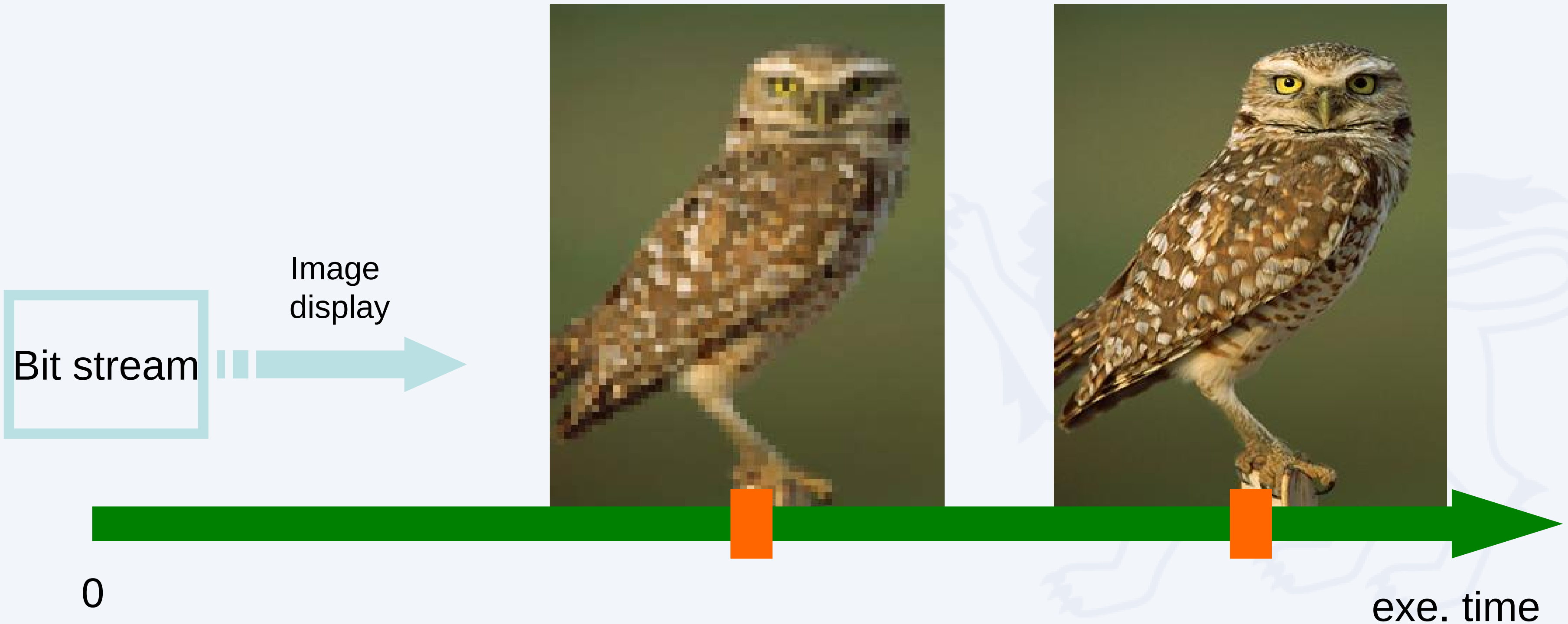
- **Certain real-time applications have variable code sizes, i.e., flexible task execution.**

- **In Imprecise-Computation model, a task is decomposed into: <u>mandatory</u> part and <u>optional</u> part.**



The IC model – Applications

- **Applications:**
  multimedia image processing, track tracing, etc.



Scheduling IC modeled tasks

- **Static/offline scheduling**
  - Under time constraints, achieves max. QoS with an optimized formulation [Aydin’01]
  - Energy constrained optimization for embedded systems [Rusu’03]

- **Dynamic/online scheduling**
  - Quasi-static approach [Cortes’06]

A better dynamic scheduling approach

- **Opportunities for improvement**
  - inter-task DVS vs. intra-task DVS
    - Mandatory and optional tasks not necessarily have the same frequency
  - Quality-of-Service is usually a convex function of execution time

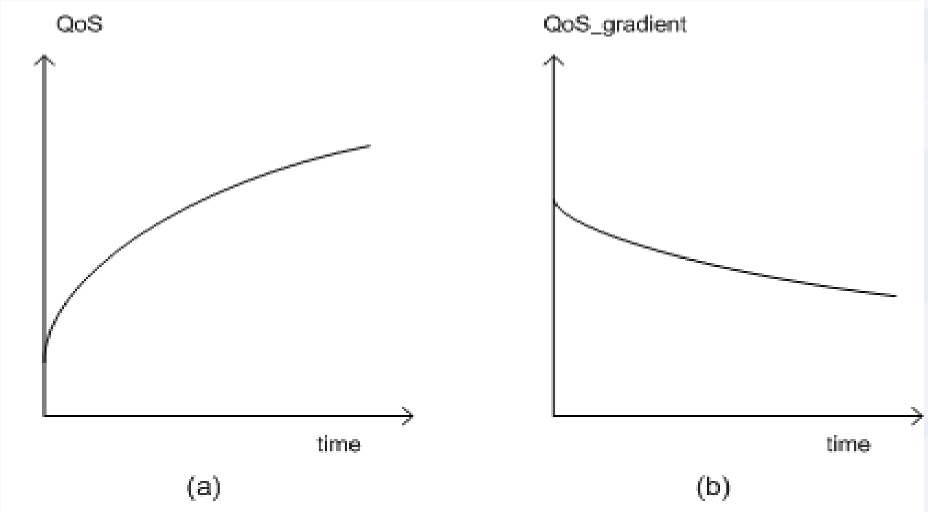


Fig. 1 (a) A convex QoS function and (b) its first derivative

A better dynamic scheduling approach (cont.)

- **Convex Quality-of-Service function**
  - Run-time slack reclamation: the slack time is distributed to tasks with larger gradients of QoS functions
  - Slack time allotting = gradient curve shifting

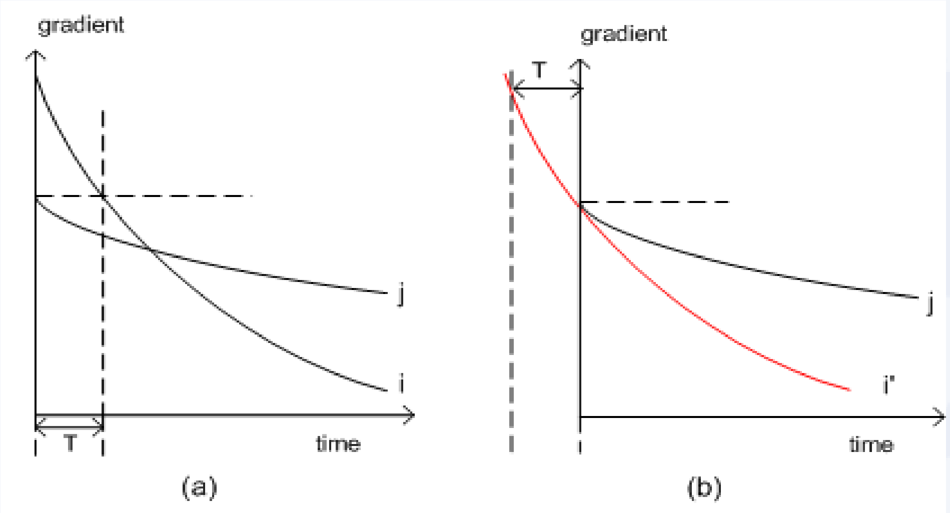


Fig. 2 (a) First derivatives of QoS functions *i* and *j* (b) shifting *i* to *i'* by *T*

The Gradient Curve Shifting Algorithm

- **The Algorithm**

```
FUNCTION: GCS
input  => slack cycles to be allotted
output <= remaining slack

if curve i is highest
      shift i until intercepting second highest curve on y-axis;
      return the remaining slack;
else, if i is as high as some other curves
      shift curves together until intercepting second highest on y-axis;
      return the remaining slack;
end if
END of GCS
```

- **Besides execution time allocation, voltage selection is essential**
- **In general,** $V(O_{slack}) \neq V(O_{prev\_allotted})$
  - Concept of intra-task DVS

Simulation results

- **Comparison between GCS algorithm and quasi-static approach**

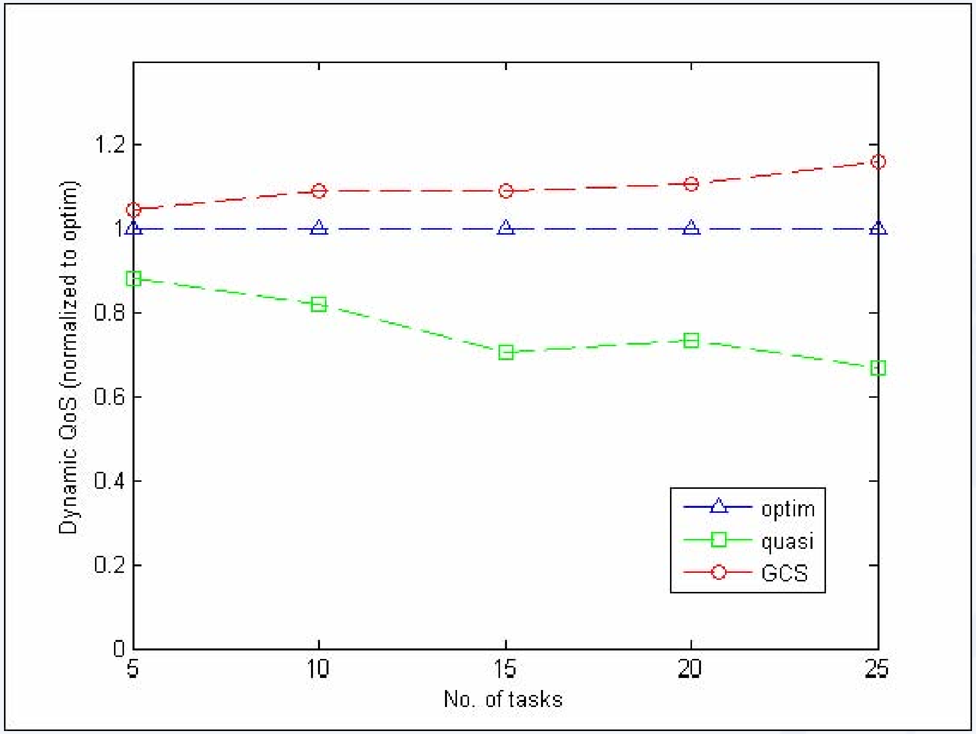


Fig. 3 Normalized dynamic QoS gain vs. no. of tasks

Simulation results (cont.)

- **Results after disabling DVS functionality**

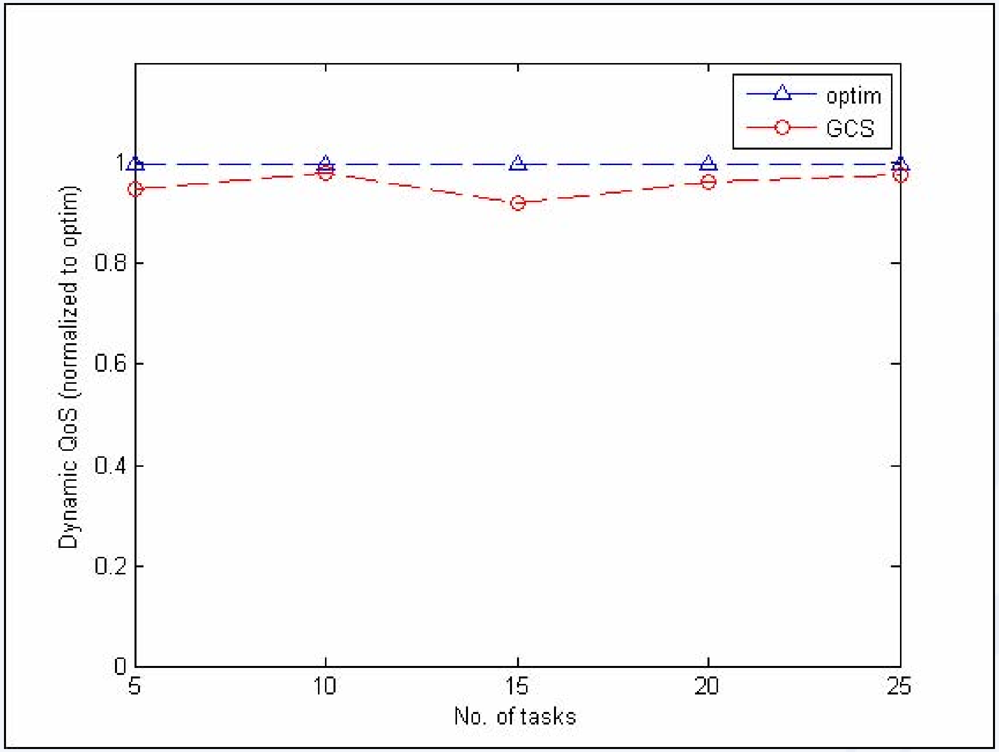


Fig. 4 Effects of no DVS applied to GCS and optimal solutions

Conclusions

- Introduction to Imprecise-Computation model and scheduling approaches
- The Gradient Curve Shifting algorithm
- Simulation results showed the superiority of our approach in achieving QoS under timing and energy constraints