Low Overhead Dynamic QoS Optimization Under Variable Task Execution Times

Sergiu Rafiliu   Petru Eles   Zebo Peng

Department of Computer and Information Science,
Linköping University, Sweden

November 3, 2010

Motivation

Large, unpredictable load variations

- Large
  - Complex set of applications
  - Not all functionality runs at all times
- Unpredictable
  - Complex hardware platform
  - Software may change during the lifetime of the system

Efficiency

- Not waste resources (power, memory, CPU time, cost . . .)
- Optimize some Quality-of-Service metric

Motivation

Large, unpredictable load variations

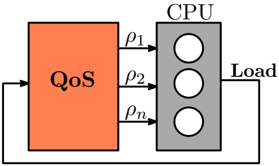
- Large
  - Complex set of applications
  - Not all functionality runs at all times
- Unpredictable
  - Complex hardware platform
  - Software may change during the lifetime of the system

Efficiency

- Not waste resources (power, memory, CPU time, cost ...)
- Optimize some Quality-of-Service metric

Motivation

Low Overhead Dynamic QoS Optimization Under Variable Task Execution Times

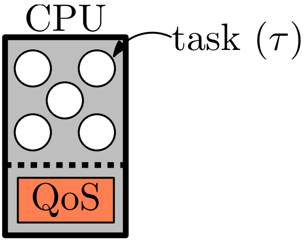


On-line approaches:

- Keep a certain reference Load
- Optimize Quality-of-Service
- Low Overhead

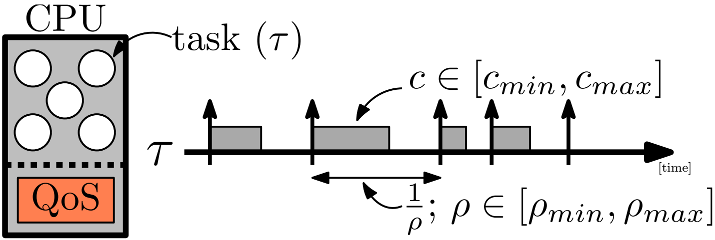
Outline

System Model



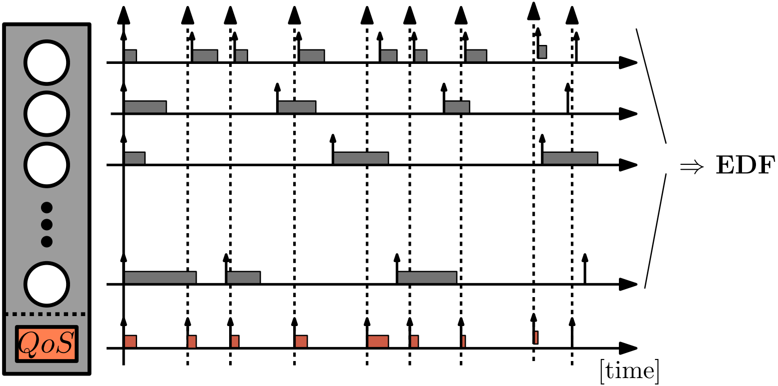
- Execution times vary in unknown ways
- Task rates are decided by the **QoS Controller**
- Jobs are scheduled through **EDF**

System Model



- Execution times vary in unknown ways
- Task rates are decided by the **QoS Controller**
- Jobs are scheduled through **EDF**

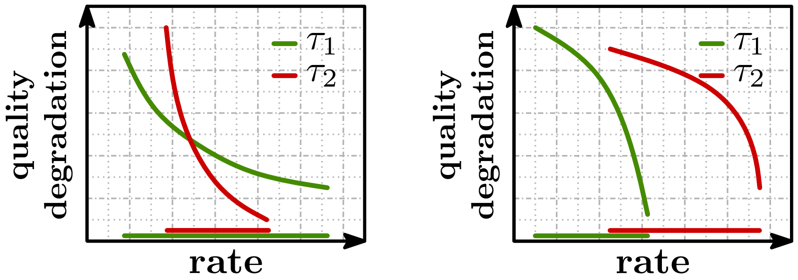
System Model



Quality-of-Service

Each task posseses an abstract **quality degradation** curve:

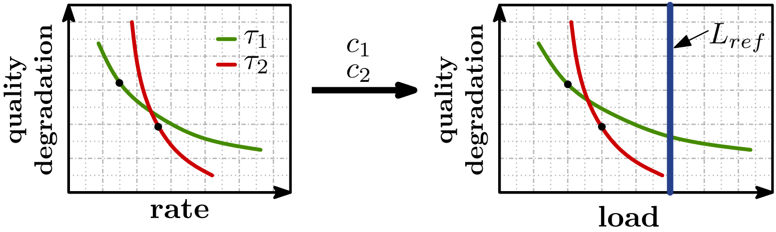
- functions of rate
- decreasing, strictly monotonic curves
- higher rates ⇒ better quality ⇒ lower quality degradation.



QoS Controller

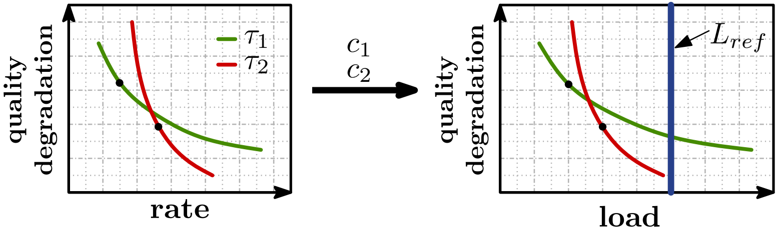
1. Determines task rates:
   - Maximize overall Quality-of-Service
   - Keep a reference CPU load given by the designer
     - determine execution times
     - determine number of jobs to be executed
2. Determines its next activation time.

QoS Controller

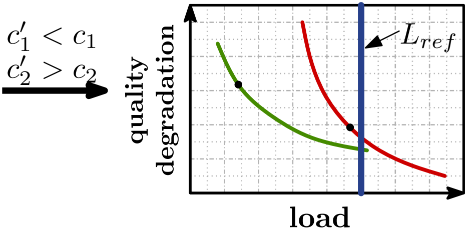


$$L = \sum_i c_i \cdot \rho_i$$

QoS Controller



$$L = \sum_i c_i \cdot \rho_i$$

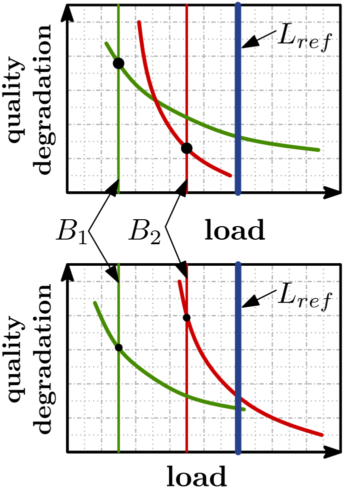


Approaches

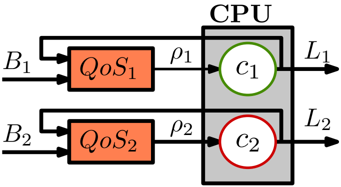
On-line approaches (QoS controllers):

- Constant Bandwidth
- Uniform QoS
- QoS Derivative
- Corner Case

Constant Bandwidth

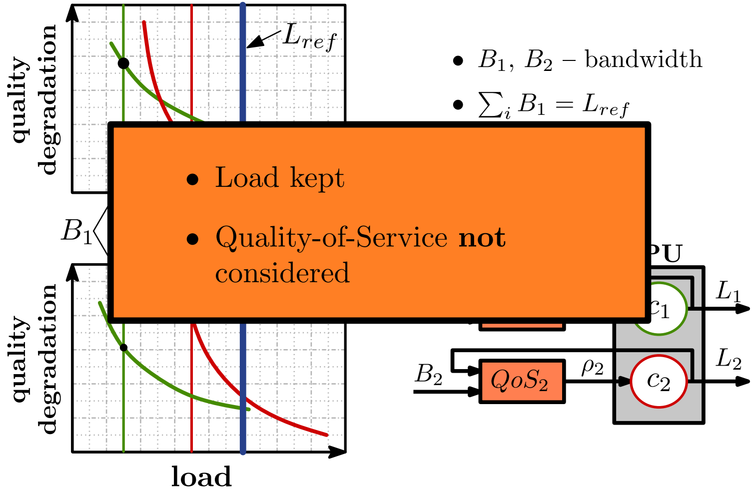


- B_1, B_2 – bandwidth
- $\sum_i B_1 = L_{ref}$



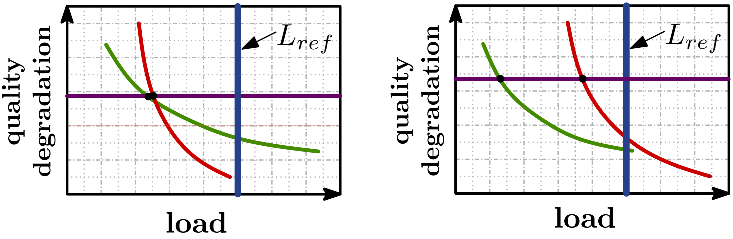
Constant Bandwitdh

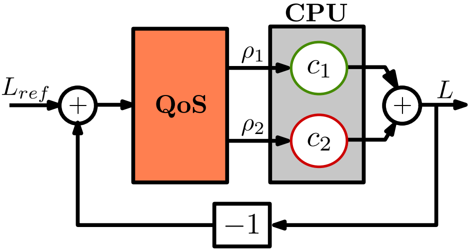
- B_1, B_2 – bandwidth
- $\sum_i B_1 = L_{ref}$



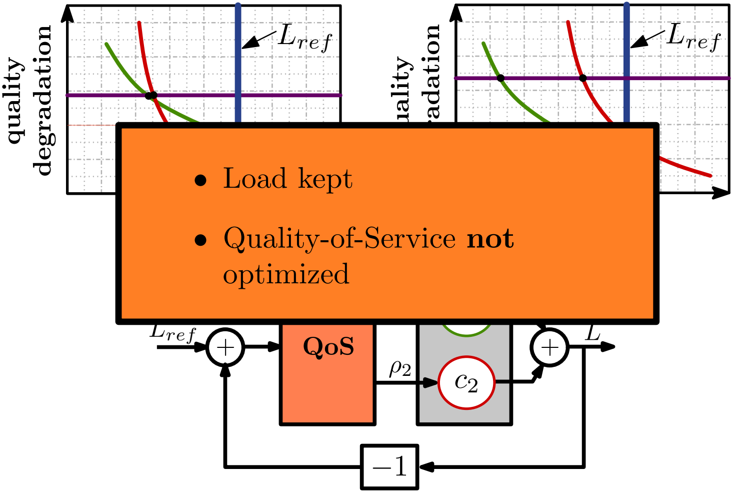
- Load kept
- Quality-of-Service **not** considered

Uniform QoS


quality degradation
L_{ref}
load
quality degradation
L_{ref}
load

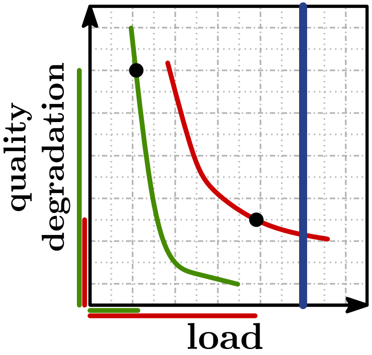

CPU
L_{ref}
QoS
ρ_1
ρ_2
c_1
c_2
L
-1

Uniform QoS

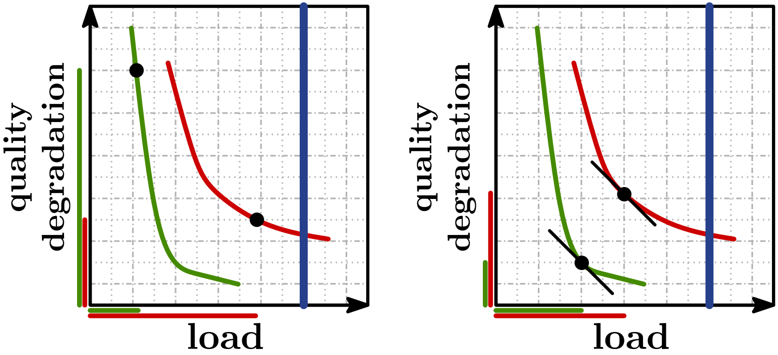


- Load kept
- Quality-of-Service **not** optimized

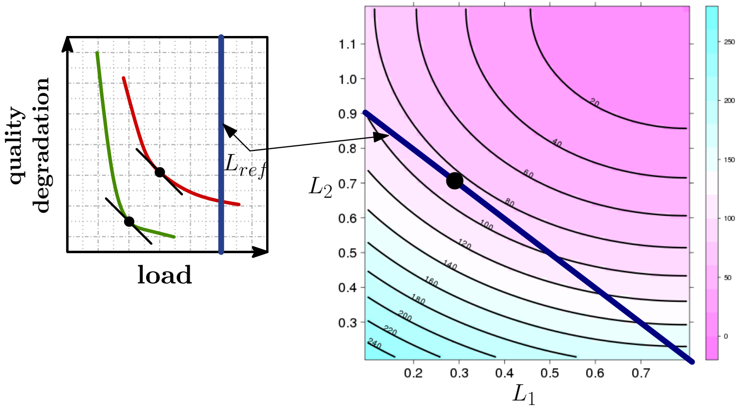
QoS Derivative



QoS Derivative

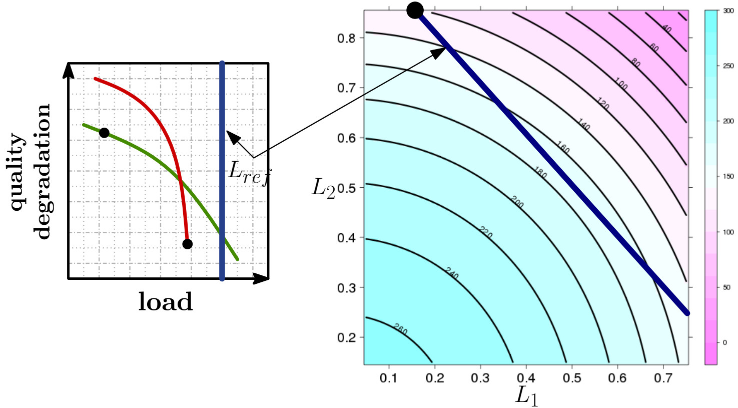


QoS Derivative



- Optimum approach for **convex** quality degradation curves
- Convex optimization problem

Corner Case

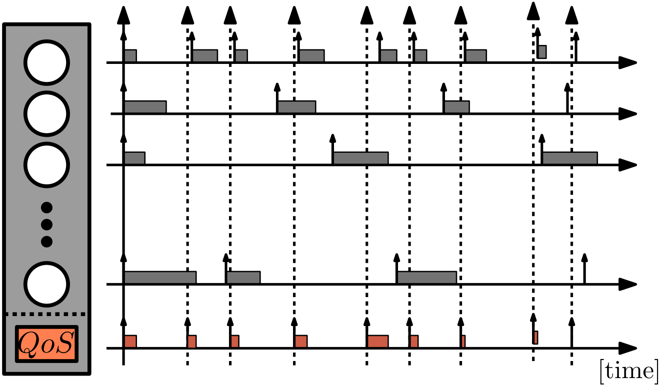


- For **concave** quality degradation curves, the optimum is one of the corner cases

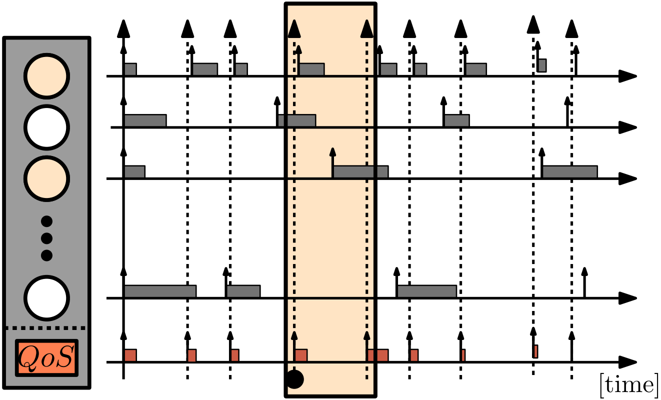
Overhead

- Active set of tasks (no. of tasks to be considered)
- Complexity

Active Set of Tasks



Active Set of Tasks



QoS Controller Complexity

QoS Controller	Complexity
Constant Bandwidth	$n \cdot O(1)$
Uniform QoS	$O(n \cdot O(q))$
QoS Derivative	$O(n \cdot O(q))$
Corner Case	$O(n \cdot log(n))$

- **n** – number of tasks (in the active set)
- **O(q)** – complexity of calling the quality degradation function for a value

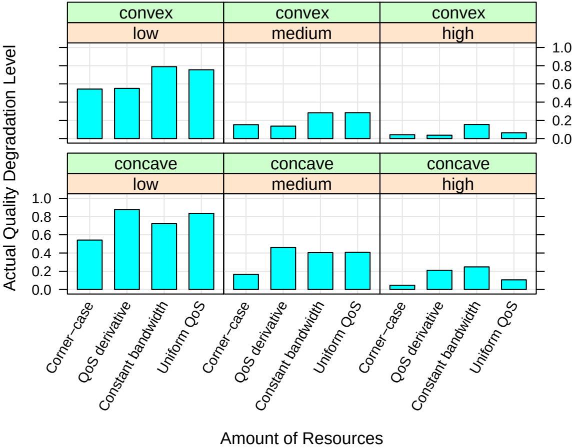
Other Problems

- Measure execution times
- Determine the number of jobs to execute (in the case of overload)
- Determine the QoS Controller's next activation point

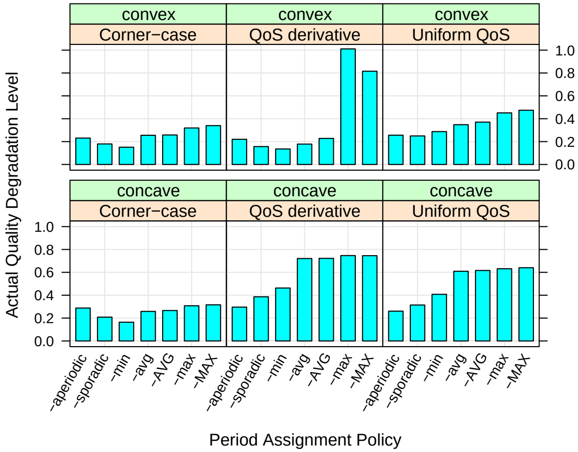
Setting

- 240 randomly generated test-cases with convex or concave quality degradation curves
- 5 to 100 tasks
- execution times variation $[c_{min}, c_{max}] = [c_{min}, 100 \cdot c_{min}]$
- 3 different processors (*low*, *medium*, *high*)
- 6 different overhead setting ($P1$ to $P6$)

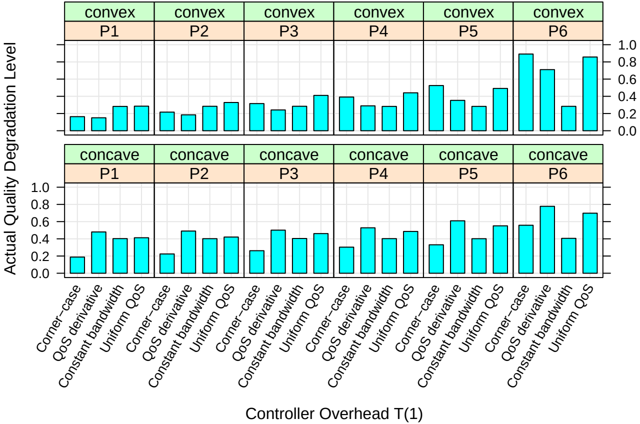
On-line Approaches



Period Assignment



Active Task Set and Overheads



Conclusions

- Solved the QoS problem for mono-processor systems with independent tasks
- Proposed approaches for **convex** and **concave** quality degradation curves
- Determined Controller complexity
- Considered overheads
- Considered other practical problems

Future Work

- More complex architectures
- Other types of resources
- More optimization criteria
- Stability

Questions???