Power-Aware Temporal Isolation with Variable-Bandwidth Servers

Silviu S. Craciun, Christoph M. Kirsch, Ana Sokolova
Department of Computer Sciences
University of Salzburg
Process Model

process \rightarrow time
Process Model

process

arrival

action $\alpha_1$

termination

time
Process Model

• action is a piece of code
- action is a piece of code
• action is a piece of code
Process Model

- action is a piece of code
- process is a sequence of actions
• action is a piece of code
• process is a sequence of actions
• throughput vs latency of process execution
Scheduling Problem

process 1

process 2

process n

uniprocessor

schedule the processes so that each of their actions maintains its response time
Scheduling Problem

schedule the processes so that each of their actions maintains its response time

Solvable with variable-bandwidth servers (VBS)
Scheduling Problem

Solveable with variable-bandwidth servers (VBS)

Results [SIES09]:
• constant-time scheduling algorithm
• constant time admission test

schedule the processes so that each of their actions maintains its response time
## Resources and VBS

| virtual periodic resources | period $\pi$ | limit $\lambda$ | utilization $\frac{\lambda}{\pi}$ |
Resources and VBS

virtual periodic resources

<table>
<thead>
<tr>
<th>period</th>
<th>limit</th>
<th>utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_1$</td>
<td>$\lambda_1$</td>
<td>$\frac{\lambda}{\pi}$</td>
</tr>
<tr>
<td>$\pi_2$</td>
<td>$\lambda_2$</td>
<td>$\frac{\lambda}{\pi}$</td>
</tr>
</tbody>
</table>

$\lambda_1$ and $\lambda_2$ represent the utilization within their respective periods $\pi_1$ and $\pi_2$. The utilization is calculated as $\frac{\lambda}{\pi}$. The diagram illustrates the periodic nature of the resources and their utilization.
Resources and VBS

- VBS is determined by a bandwidth cap ($u$)
- VBS processes dynamically adjust speed (change resources)
  \[
  \frac{\lambda_1}{\pi_1} \leq u \quad \frac{\lambda_2}{\pi_2} \leq u
  \]
- Generalization of constant bandwidth servers (CBS)

[Abeni and Buttazzo 2004]
One Process on a VBS

- Process running on a VBS
- Action $\alpha_1$
- Load
- Response time
One Process on a VBS

process running on a VBS

action $\alpha_1$

load

response time

process

time
One Process on a VBS

- **Process** running on a VBS
- **Time**
- **Action** $\alpha_1$
- **Load**
- **Response Time**
- **Coordinate**: $\pi_1$
One Process on a VBS

- Process running on a VBS
- Load
- Response time
- Action $\alpha_1$
One Process on a VBS

- Process running on a VBS
- Action $\alpha_1$
- Action $\alpha_2$
- Response time
- Load

- Process
- Time
One Process on a VBS

- Action $\alpha_1$
- Action $\alpha_2$
- Load
- Response time
- Process running on a VBS

$\lambda_1$, $\pi_1$, $\lambda_2$, $\pi_2$
VBS

process running on a VBS
VBS

process running on a VBS

arrival


time
process running on a VBS

arrival

release

VBS

time
VBS

process running on a VBS

arrival

limit

release

time
VBS

process running on a VBS

arrival

limit

release

release

time
VBS

process running on a VBS

arrival

limit

preemption

release

release

time
process running on a VBS
process running on a VBS

arrival

limit

preemption

limit

release

release

release
VBS

process running on a VBS

arrival

limit

preemption

limit

completion

release

release

release

time
VBS

process running on a VBS

arrival

preemption

limit

release

limit

release

limit

release

termination
VBS

arrival

preemption

limit

limit

completion

release

release

release

termination

response time under VBS

process running on a VBS
VBS

process 1
(2,4)

process 2
(2,8)

process 3
(1,6)

multiple processes are EDF-scheduled
VBS

multiple processes are EDF-scheduled
multiple processes are EDF-scheduled
VBS

multiple processes are EDF-scheduled
VBS

process 1
(2,4)

process 2
(2,8)

process 3
(1,6)

multiple processes are EDF-scheduled
multiple processes are EDF-scheduled
multiple processes are EDF-scheduled
Processes $P_1, P_2, \ldots, P_n$ on VBSs $u_1, u_2, \ldots, u_n$ are schedulable if $\sum_{i=1}^{n} u_i \leq 1$
Scheduling Result and Bounds

Processes $P_1, P_2, \ldots, P_n$ on VBSs $u_1, u_2, \ldots, u_n$ are schedulable if $\sum_{i=1}^{n} u_i \leq 1$

For any action $\alpha$ on a resource $(\lambda, \pi)$ we have:

- upper response-time bound $\left\lceil \frac{\text{load}}{\lambda} \right\rceil \pi + \pi - 1$
- lower response-time bound $\left\lfloor \frac{\text{load}}{\lambda} \right\rfloor \pi$
- jitter $\pi - 1$
Processes $P_1, P_2, \ldots, P_n$ on VBSs $u_1, u_2, \ldots, u_n$ are schedulable if $\sum_{i=1}^{n} u_i \leq 1$

For any action $\alpha$ on a resource $(\lambda, \pi)$ we have:

- upper response-time bound $\left\lceil \frac{\text{load}}{\lambda} \right\rceil \pi + \pi - 1$
- lower response-time bound $\left\lfloor \frac{\text{load}}{\lambda} \right\rfloor \pi$
- jitter $\pi - 1$
Scheduler Overhead
[SIES09]
Power-Aware VBS

Dynamic Voltage and Frequency Scaling
Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)
Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

process 1
(1,4)

process 2
(1,4)
Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

process 1
(2,4)

process 2
(2,4)
Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

- **Process 1**
  - (2,4)
  - Timeline: 0, 4, 8, 12

- **Process 2**
  - (2,4)
  - Timeline: 0, 4, 8, 12

Possible whenever there is slack in the system
An EDF-schedulable set of tasks is still schedulable if the processor frequency in between any two release times is set to at least $U_c \cdot f_{max}$.

Current total utilization of all released tasks in the considered interval of time between two releases.
Frequency-scaling VBS

Slack
Frequency-scaling VBS

- Slack
  - Static Slack
  - Dynamic Slack
- Action Slack
  - Termination Slack
Frequency-scaling VBS

Frequency is scaled to the sum of the bandwidth caps and not changed at runtime.

- Slack
  - Static Slack
  - Dynamic Slack
  - Action Slack
  - Termination Slack
Frequency-scaling VBS

- **Static Slack**: Frequency is scaled to the sum of the bandwidth caps and not changed at runtime.
- **Dynamic Slack**: Frequency is scaled at release time to the sum of the utilizations of the released actions.
- **Action Slack**:
- **Termination Slack**:
Frequency-scaling VBS

- **Static Slack**: Frequency is scaled at release time to the sum of the utilizations of the released actions.

- **Dynamic Slack**: Frequency is scaled to the sum of the bandwidth caps and not changed at runtime.

- **Action Slack**: New limits are computed for each action such that the upper response-time bound is maintained.

- **Termination Slack**: New limits are computed for each action such that the upper response-time bound is maintained.
Frequency-scaling VBS
Frequency-scaling VBS

Static slack

\[ f = \sum_{i=1}^{n} u_i \cdot f_{\text{max}} \]
Frequency-scaling VBS

Static slack

\[ f = \sum_{i=1}^{n} u_i \cdot f_{max} \]

Action slack

\[ f = \sum_{i=1}^{n} \frac{\lambda_{i,j}}{\pi_{i,j}} \cdot f_{max} \]
Frequency-scaling VBS

Static slack

\[ f = \sum_{i=1}^{n} u_i \cdot f_{max} \]

Action slack

\[ f = \sum_{i=1}^{n} \frac{\lambda_{i,j}}{\pi_{i,j}} \cdot f_{max} \]

Termination slack

\[ f = \sum_{i=1}^{n} \frac{\lambda_{i,j}^*}{\pi_{i,j}} \cdot f_{max} \]

\[ \lambda_{i,j}^* = \left[ \frac{l_{i,j}}{n_{i,j}} \right] \quad n_{i,j} = \left[ \frac{l_{i,j}}{\lambda_{i,j}} \right] \]
Frequency-scaling VBS

Static slack

\[ f = \sum_{i=1}^{n} u_i \cdot f_{max} \]

Action slack

\[ f = \sum_{i=1}^{n} \frac{\lambda_{i,j}}{\pi_{i,j}} \cdot f_{max} \]

Termination slack

\[ f = \sum_{i=1}^{n} \frac{\lambda_{i,j}^*}{\pi_{i,j}} \cdot f_{max} \]

\[ \lambda_{i,j}^* = \left[ \frac{l_{i,j}}{n_{i,j}} \right] \]

\[ n_{i,j} = \left[ \frac{l_{i,j}}{\lambda_{i,j}} \right] \]

Termination and action slack can be used separately or together
Power-Aware VBS

Assuming a simple power model \( (P \propto V^2) \)
Look-ahead FS-VBS
Look-ahead FS-VBS

With knowledge of future events:
redistribute computation time between periods
Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
- feasible online method
Look-ahead FS-VBS

With knowledge of future events:
- redistribute computation time between periods
  - optimal offline method
  - feasible online method

May help to handle:
more complex power models
Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
- feasible online method

May help to handle:

- more complex power models
- frequency switching cost (time and power)
Look-ahead FS-VBS

With knowledge of future events:
- redistribute computation time between periods
  - optimal offline method
  - feasible online method

May help to handle:
- more complex power models
- frequency switching cost *(time and power)*
  - time overhead included using overhead accounting [RTAS10]
Look-ahead FS-VBS

- **Process 1**: 30% utilization
- **Other Utilization**: 70% utilization
- **Total Utilization**: 100% utilization
Look-ahead FS-VBS

- Process 1: 30%
- Other utilization: 70% (30%)
- Total utilization: 100% (60%)

knowledge of future events
Look-ahead FS-VBS

- Process 1
  - Modified
    - 10% to 50%
  - Other utilization
    - 70% to 30%
  - Total utilization
    - 80% to 80%
Look-ahead FS-VBS

Actual improvement depends on the power model.
Look-ahead FS-VBS

Assuming a simple power model \( (P \propto V^2) \)
Look-ahead online FS-VBS
Look-ahead online FS-VBS

Assume a simple power model \((P \propto V^2)\)
Look-ahead online FS-VBS

Assume a simple power model \((P \propto V^2)\)

![Graph showing utilization over time with future events]

- \(u^\alpha\): Utilization over time
- \(u^P\): Power utilization over time
- \(u^S\): Average utilization over time

Knowledge of future events

Average utilization 65%
Look-ahead online FS-VBS

Assume a simple power model \((P \propto V^2)\)

Modify the limits in each period (whenever possible) s.t. the utilization approximates the average utilization
Look-ahead online FS-VBS

Assume a simple power model \((P \propto V^2)\)

Modify the limits in each period (whenever possible) s.t. the utilization approximates the average utilization.
Conclusions
Conclusions

- Server-based scheduling for temporal isolation
Conclusions

• Server-based scheduling for temporal isolation

• VBS for variable execution speed
Conclusions

- Server-based scheduling for temporal isolation

- VBS for variable execution speed

- Power-aware VBS