Real-Time SMP Scheduling

Ted Baker
Department of Computer Science
Florida State University
Tallahassee, FL 32312
http://www.cs.fsu.edu/~baker
1. A taste of real-time scheduling theory
2. A research process
   - Where an idea comes from
   - Why doing research in "backwaters" may lower stress
   - What to do when somebody else "scoops" you
3. A recent research result of mine
   - What to expect from referees
   - How to publish
4. What I hope to do with the idea next

Overview


Background: Periodic Task Model

- Each task has a period $T_i$.
- Each task has a worst case compute time $c_i$.
- Each task has a relative deadline $d_i$.
- Set of tasks $\tau_1, \ldots, \tau_n$.

\[
\frac{1}{\sum_{i=1}^{n} c_i/T_i} = \text{total utilization}
\]

\[
\frac{1}{c_i} = \text{processor utilization of task } \tau_i
\]
Gantt Chart of a Periodic Task’s Execution
Period is Just a Lower Bound
Theorem

A set of $n$ independent periodic tasks is schedulable by preemptive EDF scheduling on one processor if

$$1 \geq \frac{p_i}{c_i} \sum_{u}$$

Q: How does this generalize for $m$ processors?
Bad Example for MP EDF Scheduling
Papers are written on how to partition tasks between processors, a bin packing problem. Everybody assumes tasks must be bound to processors in a static (or nearly static) way, and single-processor scheduling applied to each processor.

Everybody says EDF scheduling is no good for multiprocessors. This example, which shows worst-case achievable processor utilization can be as bad as \( 1 \) (compared to ideal value of \( m \)).
1990: My Observation about a "Problem Window"
If we have an upper bound on individual task utilizations, we have a lower bound on the worst-case achievable utilization. If we use an upper bound on \( \lambda(1 - \gamma) \), where \( \lambda = \max_u \sum_i^L \frac{1}{\Lambda_i} \) where \( \lambda_{\max} \) is the maximum rate at which tasks can be processed, looking at the example, the worst-case achievable utilization with EDF seems to be close to \( m(1 - \gamma) \).
Years Go By

Italk to Lui Sha (then CMU/SEI and now UIUC) about the idea. He doesn’t seem to understand what I am talking about enough pick up on it. I suggest to three different Ph.D. students that they work on the problem. They get nowhere.

I am still convinced it shouldn’t be too hard to prove something here.

I talk to Lui Sha (then CMU/SEI and now UIUC) about the idea. He doesn’t seem to understand what I am talking about enough pick up on it.

By
No longer department chair, with no current Ph.D. students, I decide to work out the result myself.
The Key Lemma

Lemma (upper bound on EDF load) For any busy window $[t, t + \Delta)$ with respect to $\tau_k$, the EDF load $W_i$ due to $\tau_i$ is at most $|p|$, where

\[
\begin{align*}
\frac{|t|}{\tau_k} & > \gamma \quad \text{if} \quad \frac{\gamma p}{\frac{\gamma p}{|p|} + (\frac{\gamma p}{|p|} + 1)\frac{|t|}{\tau_k}} = |p| \\
\frac{|t|}{\tau_k} & < \gamma \quad \text{if} \quad \frac{\gamma p}{\frac{\gamma p}{|p|} + (\frac{\gamma p}{|p|} + 1)\frac{|t|}{\tau_k}} = |p|
\end{align*}
\]
Theorem (EDF schedulability test) A set of periodic tasks $T_1,T_2,...,T_n$ is schedulable on $m$ processors using preemptive EDF scheduling if, for every task $T_k$, where $p_i$ is as defined in the lemma above.

\[
\frac{\gamma_p}{\gamma_c} + \left(\frac{\gamma_p}{\gamma_c} - 1\right) w \geq \min_{i=1}^{n} \left\{ p_i \right\}
\]

The Final Result
The Nice Corollary

A set of periodic tasks $T_1, \ldots, T_n$ all with deadline equal to period is guaranteed to be schedulable on $m$ processors using preemptive EDF scheduling if

$$\chi + (\chi - 1)m \geq \frac{\sum_{i=1}^{n} \frac{c_i}{T_i}}{m}$$

The Nice Corollary
How I was "Scooped"

A periodic task set \( \{ \tau_1, \tau_2, \ldots, \tau_n \} \) is light on \( m \) processors if:

1. \[ \sum_{i=1}^{n} \frac{1}{c_i T_i} > \frac{1}{C} \]
2. \[ \frac{1 - \frac{w}{m} \tau_i}{c_i} > \frac{1}{C} \]

Theorem (Srinivasan, Baruah [4]) Any periodic task system that is light on \( m \) processors is scheduled to meet all deadlines on \( m \) processors by EDF.
What I Thought I was able to Salvage

- new proof technique
- pre-period deadlines
- more general utilization bound test: $\frac{m^2}{2m-1}$ is just a special case of $m(1-\lambda) + \lambda$
- proof that the utilization bound is tight
Theorem (Goossens, Funk, Baruah[3]) A set of periodic tasks \( \tau_1, \ldots, \tau_n \) with deadline equal to period, is guaranteed to be schedulable on \( m \) processors using preemptive EDF scheduling if

\[
\gamma + (\gamma - 1)\mu \geq \frac{1}{\gamma} \sum_{i=1}^{n} \frac{1}{c_i T_i}
\]

where \( \lambda \) max = \( \gamma \) and

\[
\{u \cdot \cdots \cdot i = 1 \mid i T/c_i \} \}
\]
What I was able to salvage

- decided to merge fixed-priority results into same paper
- new proof technique
- pre-period deadlines
What Referee 1 Said

"...Although the paper has some contributions to be presented, the topic and motivation is not that exciting. ..."

Consequences of being scooped.
Quantitative justification of the proposed analysis is required. The proposed analysis is more general in the sense that it can handle preperiod deadlines. Obviously, this simple modification of the original execution time C to C+(P-D) to assure P-D (D is the preperiod deadline) can simplify the sense that it can handle preperiod deadlines. We can simplify the change of the execution time by C to C+(P-D). More general in the sense that it can handle preperiod deadlines is required.

There is an improvement, but to show it is a good idea for more research. One way to do this is via simulation on a large randomly chosen collection of task sets.

What Referee 1 Said
"...for some important theorems, only sketch of proof is given referring their two technical reports. This makes readers hard to follow the theorems ...

You can't win on this; given the 20-page limit for papers. Putting in more proofs means less results, and maybe an even less exciting paper.

What Referee I Said
What Referee 2 Said

"Given the originality of this work, I strongly recommend that this paper be accepted."
"The paper is well written, and the results are of theoretical interest. ..."
What Referee 3 Said

“... practical usage ... is limited ... unrealistic system model ... scalability and processor cache considerations ... modern operating systems use a priority queue per processor ... schedule the task on the processor where its previous instance executed ... not ... the processor that is executing the lowest priority task ... ... introduces a form of priority inversion when tasks are dynamically dispatched ... challenging to dynamically schedule tasks in a multiprocessor in consistent priority order ... many other factors make the assumption of perfect preemption invalid.”

A valid question. This is something we need to look into further. Clearly, there are trade-offs involved.
What Referee 3 said...

Reference to the people who "scooped" me, the reviewer missed it. We actually answered in the paper, but since it was just a few sentences and a reference to the people who "scooped" me, the reviewer missed it. Some statement on tightness of the bounds is needed. Since it was just a few sentences and a reference to the people who "scooped" me, the reviewer missed it.

...Are the bounds tight, in the sense that Liu and Layland bound is while many subsequent schedulability are not? Some statement on tightness of the bounds is needed.
What Referee 3 Said

"...Lemma 9 is obvious. The proof obscures the result..."

You can't please everybody on this kind of issue. Referee 1 wanted more details on proofs.
What I Hope to Do Next

- Revise periodic server scheduling algorithms in the MP context
- Extend analysis to include blocking for mutexes
  1. Simulate processors
  2. Implement and test to determine real switching overheads
  3. Distribute implementation
- Try to resolve Referee 3’s issue about fixed vs. dynamic binding of tasks to
  the tighter preperiod deadline schedulability test
- Try to resolve Referee 1’s issue about how much is gained, and how often, by
The reasoning

The demand of a time interval is the total amount of computation that would need to be completed within the window for all the deadlines within the interval.

If we can find a lower bound on the load of a problem window that is necessary to possibly generate so much load in the problem window, that would be sufficient to serve as a schedulability condition.

Definition

The load of an interval \([i, i+\Delta]\) is \(W/\Delta\), where \(W\) is the demand of the interval.

If we can show that a given set of tasks could not possibly miss its deadline, and we can show that a given set of tasks could not possibly generate so much load in the problem window that is necessary to serve as a schedulability condition.
Since the problem job misses its deadline, the sum of the lengths of all intervals in which the problem job does not execute must exceed its slack time, $d_k - c_k$. Therefore, the lower bound on load is released.
Lemma

If $W_{d_k}$ is the load of the interval $t_{d_k}$, where $t_{d_k}$ is a missed deadline of $\tau_k$, then

$$\gamma p + \left( \frac{\gamma p}{\gamma c} - 1 \right) w < \frac{\gamma p}{M}$$

where $\gamma p + i$ is the load of the interval $[\gamma p + i, i]$ if $\gamma p / M$ is the load on load.
Analysis of Maximum Load
Carried-in Load

**Definition** The *carry-in* of $\tau_i$ at time $t$ is the residual compute time of the last job of task $\tau_i$ released before $t$, if any, and is denoted by the symbol $\varepsilon$. 
Upper Bound on EDF Demand

Lemma

For any busy window $t + \Delta$ of task $\tau_k$ (i.e., the maximal $\lambda$-busy downward extension of a problem window) and any task $\tau_i$, the EDF demand $W_i$ of $\tau_i$ in the busy window is no greater than

$$\max\{0, c_i^{\Delta} - \phi T_i + p - (\phi - c_i^{\Delta}) \}$$

where

$$g = 0$$

and $u$ otherwise.

$$\phi + \lambda L / (p - \lambda) = u, \quad \phi - p + \lambda L = \phi$$

For any busy window $t + \Delta$ of task $\tau_k$ (i.e., the maximal $\lambda$-busy downward extension of a problem window) and any task $\tau_i$, the EDF demand $W_i$ of $\tau_i$ in the busy window is no greater than

$$\max\{0, c_i^{\Delta} - \phi T_i + p - (\phi - c_i^{\Delta}) \}$$

where

$$g = 0$$

and $u$ otherwise.

$$\phi + \lambda L / (p - \lambda) = u, \quad \phi - p + \lambda L = \phi$$
References


