Scheduling in the Dark

Jeff Edmonds
York University

Non-clairvoyant Multiprocessor

Scheduling of Jobs

with Arbitrary Arrival Times

and Changing Execution Characteristics
The Scheduling Problem

- Allocate $p$ processors to a stream of $n$ jobs

\[ \text{Average Response Time:} \]
\[ \text{AvgResp}(S(J)) = \frac{1}{n \sum_{i \in [1..n]} c_i - r_i} \]

- Competitive Ratio:
\[ \text{Min}_{S \in S} \text{Max}_{J \in J} \frac{\text{AvgResp}(S(J))}{\text{AvgResp}(OPT(J))} \]

Time

J1 J2 J3

J4 J5

p Processors
Different Classes $J$ of Job Sets $J$

- Arrival Times (Arbitrary or Batch)
- Some Class of Speedup Functions
  
  $\Gamma(\beta)$ is the rate (work/time) when allocated $\beta$ processors.

Sequential | Fully Par. | NonDecreasing SubLinear

Super-Linear | Gradual | …

- # of Phases in a Job (Single or Arbitrary)
SubLinear-NonDecreasing Speedup Functions

- A set of jobs \( J = \{J_1, J_2, \ldots, J_n\} \)
- Each job has phases \( J_i = \langle J_i^1, J_i^2, \ldots, J_i^{q_i} \rangle \)
- Each job phase \( J_i^q = \langle W_i^q, \Gamma_i^q \rangle \) is defined by
  - \( W_i^q \) is the amount of *work*
  - \( \Gamma_i^q(\beta) \) is the rate (work/time) with \( \beta \) processors
- Speedup functions must be:
  - NonDecreasing: \( \beta_1 \leq \beta_2 \Rightarrow \Gamma(\beta_1) \leq \Gamma(\beta_2) \).
  - SubLinear: \( \beta_1 \leq \beta_2 \Rightarrow \Gamma(\beta_1)/\beta_1 \geq \Gamma(\beta_2)/\beta_2 \).

### Examples

- Sequential
- Fully Par.
- Fully to \( \beta \)
Examples of Schedulers (algorithms)

- **Shortest Remaining Work First (SJF)**
- **Balance (BAL)**
  - Shortest Run First
- **Round Robin (RR)**
- **Equal-Partition (EQUI)**
Different Classes $S$ of Schedulers $S$

- Clairvoyance
  - No, partial, or complete knowledge
- Computation Time
  - Unbounded, Poly Time, or Reasonable
- # of Preemptions (re-allocation of processors)

The Optimal Scheduler

- Unbounded
  - Clairvoyance
  - Computation Time
  - Preemptions
Devil and one player
Lower Bounds

• Equal-Partition

\[
\text{Flow}(\text{OPT}) = O(n) \\
\text{Flow}(\text{EQUI}) = O(n) \\
\text{AvgResp}(\text{EQUI}(J)) \geq \Omega \left( \frac{n}{\log n} \right) \cdot \text{AvgResp}(\text{OPT}(J))
\]

• Balance

\[
\text{Flow}(\text{OPT}) = O(n) \\
\text{Flow}(\text{BAL}) = O(n^2) \\
\text{AvgResp}(\text{BAL}(J)) \geq \Omega \left( n \right) \cdot \text{AvgResp}(\text{OPT}(J))
\]

• General Non-Clairvoyant Schedulers \( S \)

\[
\text{AvgResp}(S(J)) \geq \Omega \left( \sqrt{n} \right) \cdot \text{AvgResp}(\text{OPT}(J))
\]
Devil $2 + \epsilon$. 
Main Result

For any set of jobs $J$ with

- arbitrary arrival times
- arbitrary number of phases
- sublinear-nondecreasing speedup functions

$$\frac{\text{AvgResp}(\text{EQUI}_{2+\epsilon}(J))}{\text{AvgResp}(\text{OPT}(J))} \leq \mathcal{O}\left(1 + \frac{1}{\epsilon}\right)$$
Wasting Resources on Sequential Jobs

At most $\frac{1}{2+\epsilon}$ of our resources are wasted on sequential jobs.
Designing an Operating System

- Predict the future.
- How much work in job?
- Fully par. or seq.?
- Design & code better algs.
- Spend more cpu time.
- Buy $2 + \epsilon$ times as many processors.
- Run $EQUI$. 
<table>
<thead>
<tr>
<th></th>
<th>Sched</th>
<th>Jobs</th>
<th>s</th>
<th>comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MPT]</td>
<td>EQUI</td>
<td>Batch</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>[ECBD]</td>
<td></td>
<td>, , or</td>
<td>1</td>
<td>[2.71, 3.74]</td>
</tr>
<tr>
<td>[KP]</td>
<td>BAL</td>
<td>Arb. Arr.</td>
<td>1</td>
<td>Ω((n))</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 + (\epsilon)</td>
<td>1 + (\frac{1}{\epsilon})</td>
</tr>
<tr>
<td>[BC]</td>
<td></td>
<td>, , or</td>
<td>(s \geq 2)</td>
<td>(\frac{2}{s})</td>
</tr>
<tr>
<td>new</td>
<td></td>
<td></td>
<td>(s)</td>
<td>Ω((n))</td>
</tr>
<tr>
<td>[MPT]</td>
<td>EQUI</td>
<td></td>
<td>1</td>
<td>Ω((\frac{n}{\log n}))</td>
</tr>
<tr>
<td>[KP]</td>
<td></td>
<td></td>
<td>1 + (\epsilon)</td>
<td>Ω((n^{1-\epsilon}))</td>
</tr>
<tr>
<td>new</td>
<td></td>
<td></td>
<td>2 + (\epsilon)</td>
<td>([1 + \frac{1}{\epsilon}, 2 + \frac{4}{\epsilon}])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>, , or</td>
<td>(s \geq 1)</td>
<td>Ω((\frac{1}{s}))</td>
</tr>
<tr>
<td>new</td>
<td>EQUI'</td>
<td>few preempt</td>
<td>4 + (\epsilon)</td>
<td>Θ(1)</td>
</tr>
<tr>
<td>new</td>
<td>HEQUI</td>
<td>, or</td>
<td>4 + (\epsilon)</td>
<td>Θ(1)</td>
</tr>
<tr>
<td>new</td>
<td>HEQUI'</td>
<td>, or</td>
<td>Θ((\log p))</td>
<td>Θ(1)</td>
</tr>
</tbody>
</table>
Main Result

For any set of jobs $J$ with

- arbitrary arrival times
- arbitrary number of phases
- sublinear-nondecreasing speedup functions

\[
\frac{AvgResp(EQUI_{2+\epsilon}(J))}{AvgResp(OPT(J))} \leq O\left(1 + \frac{1}{\epsilon}\right)
\]
Worst Case $J$

In the worst case set of jobs $J$
each phase is either fully parallelizable or sequential.

$$\forall J \exists J' \frac{AvgResp(EQUI_{2+\epsilon}(J))}{AvgResp(OPT(J))} \leq \frac{AvgResp(EQUI_{2+\epsilon}(J'))}{AvgResp(OPT(J'))}$$

![Graph showing the comparison between EQUI and OPT for J and J']
Integrating Through Time

\[
\frac{\text{AvgResp}(\text{EQUI}_{2+\epsilon}(J'))}{\text{AvgResp}(\text{OPT}(J'))}
\]

\[
= \int_{0}^{\infty} \frac{\text{(# par. EQUI)}_t + \text{(# seq. EQUI)}_t \delta t}{1 + \text{(# seq. OPT)}_t \delta t}
\]

\[
= \int_{0}^{\infty} \frac{\text{(# par. EQUI)}_t + \text{(# seq. EQUI)}_t \delta t}{1 + \text{(# seq. EQUI)}_t \delta t}
\]
Extra Resources $s = 2 + \epsilon$

Still Number of Jobs Alive

is Unbounded

Flow(OPT) = $O(1)$

Flow($\text{EQU}_{2+\epsilon}$) = $O(1)$
Steady State
Potential Function

- \( W_t = \) set of work completed by \( OPT \) but not by \( EQUI \).
- \( F(W_t) = \) a measure of the work
- \((\text{# par. } \text{EQUI})_t \geq \frac{1}{\epsilon}(\text{# seq. } \text{EQUI})_t\)
  \(\Rightarrow F(W_t)\) decreases with time

- \( F_T = F(W_T) + \int_0^T (\text{# par. } \text{EQUI})_t - \frac{1}{\epsilon}(\text{# seq. } \text{EQUI})_t \delta t \)
- \( F_0 = 0 \)
- \( \frac{\delta F_T}{\delta T} \leq 0 \)
- \( F_\infty \leq 0 \)
- \( \int_0^\infty (\text{# par. } \text{EQUI})_t - \frac{1}{\epsilon}(\text{# seq. } \text{EQUI})_t \delta t \leq 0 \)

- \( \frac{\text{Avg Resp}(EQUI_{2+\epsilon}(J))}{\text{Avg Resp}(OPT(J))} \leq \frac{\int_0^\infty (\text{# par. } \text{EQUI})_t + (\text{# seq. } \text{EQUI})_t \delta t}{1 + (\text{# seq. } \text{EQUI})_t \delta t} \leq O(1 + \frac{1}{\epsilon}) \)
All Jobs Fully Parallelizable or Sequential

Work Completed by \( OPT \) and not by \( EQUI \)

\[
F_T = \int_0^T (m_t - \frac{\ell_t}{\varepsilon}) dt + F(W_T)
\]

\[
= \int_0^T (m_t - \frac{\ell_t}{\varepsilon}) + \frac{2}{\varepsilon} \sum_{i=1}^{m_T} i \cdot w_i
\]

\[
\frac{\delta F_T}{\delta T} = (m_T - \frac{\ell_T}{\varepsilon}) + \frac{2}{\varepsilon} \left[ (m_T \cdot 1) - \sum_{i=1}^{m_T} i \cdot \left( \frac{2 + \varepsilon}{m_T + \ell_T} \right) \right]
\]

\[
= \frac{(m_T)^2}{2} \cdot \left( \frac{2 + \varepsilon}{m_T + \ell_T} \right)
\]

\[
\leq 0
\]
“logn” Preemptions

- Preempts only when # jobs increases or decreases by a factor of 2.
- Competitive with $s = 8 + \epsilon$. 
Super Linear Speedup Functions

- Time-Space Tradeoff and Highly Parallelizable
- Competitive with $s = 4 + \epsilon$.
  - *Round Robin* (super linear phases)
  - *EQUI* (sub-linear phases)

- Bounded preemptions $\Rightarrow \Omega(n)$
NonDecreasing or “Gradual” Speedup Functions

- Competitive with $s = \mathcal{O}(\log p)$:
  - Run each job
    * for a slice of time
    * with $2^k$ processors ($\forall k \in [1, \log p]$)
  - Guaranteed to run each job phase
    * with the “right” # of processors
Conjectures

- Are the $2 + \epsilon$ extra resources needed?

$$\forall \epsilon > 0, \exists \text{ a Non-Clairvoyant Scheduler } S$$

$$\forall J \frac{\text{AvgResp}(S_{1+\epsilon}(J))}{\text{AvgResp}(OPT(J))} \leq O\left(\frac{1}{\epsilon^2}\right)$$

- Jobs arrive in a Random order

$$\frac{\text{AvgResp}(EQUI_1(J))}{\text{AvgResp}(OPT(J))} \leq O\left(1\right)$$

- Lower bound for Non-clairvoyant Schedulers.