SCHEDULING IN POLLING SYSTEMS

Adam Wierman
adamw@caltech.edu
California Institute of Technology, USA

Erik Winands  Onno Boxma
Technische Universiteit Eindhoven, the Netherlands

This work came out of an extended visit to the EURANDOM Institute in the Netherlands
WHAT IS A POLLING SYSTEM?
A system consisting of a number of queues and a single server, where there is a switchover time when the server moves between queues.
WHAT IS A POLLING SYSTEM?
A system consisting of a **number of queues** and a **single server**, where there is a **switchover time** when the server moves between queues.
WHAT IS A POLLING SYSTEM?

A system consisting of a **number of queues** and a **single server**, where there is a **switchover time** when the server moves between queues.

**Applications in:**
- Wireless protocols
- Routers
- Web servers
- Telecommunications
- Manufacturing
- Maintenance
- …

Enormous body of literature
3 MAIN DESIGN DECISIONS

When to switch?
3 MAIN DESIGN DECISIONS

When to switch?

Lots of analysis

• exhaustive
• gated
• k-limited
• time-limited
• …
3 MAIN DESIGN DECISIONS

Where to switch to?

When to switch?

Lots of analysis
- exhaustive
- gated
- k-limited
- time-limited
- …
3 MAIN DESIGN DECISIONS

Where to switch to?
Lots of analysis
static  dynamic

When to switch?
Lots of analysis
• exhaustive
• gated
• k-limited
• time-limited
• …
3 MAIN DESIGN DECISIONS

Where to switch to?
Lots of analysis
- static
- dynamic

When to switch?
Lots of analysis
- exhaustive
- gated
- k-limited
- time-limited
- ...

What order to serve jobs within a queue?
3 MAIN DESIGN DECISIONS

Where to switch to?
Lots of analysis
static  dynamic

When to switch?
Lots of analysis
- exhaustive
- gated
- k-limited
- time-limited
- ...

What order to serve jobs within a queue?
ALMOST NO ANALYSIS
FCFS is almost always assumed
WHY IS THERE NO ANALYSIS?!?

What order to serve jobs within a queue?

ALMOST NO ANALYSIS

FCFS is almost always assumed
WHY IS THERE NO ANALYSIS?!!?

1. It seems hard
   Even the analysis of the delay under FCFS requires solving a set of N linear equations.

2. It seems that scheduling won’t help much
   Scheduling only affects local performance and doesn’t change time spent waiting for the server

ALMOST NO ANALYSIS

FCFS is almost always assumed
OUR PAPER:
Mean value analysis framework that applies to a wide range of scheduling policies
OUR PAPER:
Mean value analysis framework that applies to a wide range of scheduling policies

- Analysis is tractable for many (but not all) policies
- Scheduling can provide significant reduction of delay
THE SPECIFICS OF THE MODEL

N queues, 1 server

Each queue has:
- Poisson arrivals
- General switchover times
- General job sizes
THE SPECIFICS OF THE MODEL

N queues, 1 server

Each queue has:
• Poisson arrivals
• General switchover times
• General job sizes

GOAL: Mean delay, $E[D_i]$
THE SPECIFICS OF THE MODEL

N queues, 1 server

Each queue has:
- Poisson arrivals
- General switchover times
- General job sizes

Polling order: Fixed, cyclic

Polling policies:
1. gated
   only customers present when server arrives are served
2. exhaustive
   server must empty queue before moving on

GOAL: Mean delay, $E[D_i]$
WHERE DOES DELAY COME FROM?
WHERE DOES DELAY COME FROM?

The time waiting for the server to arrive
WHERE DOES DELAY COME FROM?

The time waiting for the server to arrive

Jobs that arrived since the beginning of the cycle
WHERE DOES DELAY COME FROM?

Jobs that will arrive in the remaining part of the cycle

Jobs that arrived since the beginning of the cycle

The time waiting for the server to arrive
Theorem:

\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

**RESIDUAL** of the cycle length

i.i.d. contribution of work from each job arriving **EARLIER** in the cycle

i.i.d. contribution of work from each job arriving **LATER** in the cycle
Example: First Come First Served (FCFS)
Example: First Come First Served (FCFS)

\[ c_1(x) = X_i \]
\[ c_2(x) = 0 \]
Theorem:
\[ E[D_i(x)] = E[R_{C_i}] \left( 1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)] \right) \]

\textbf{Example:} First Come First Served (FCFS)

\[ c_1(x) = X_i \]
\[ c_2(x) = 0 \]

\[ E[D_i(x)] = E[R_{C_i}] \left( 1 + \rho_i \right) \]
\[ E[D_i] = E[R_{C_i}] \left( 1 + \rho_i \right) \]
**Theorem:**

\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

**Example:** Last Come First Served (LCFS)
Theorem:
\[ E[D_i(x)] = E[R_{C_i,}] (1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

**Example:** Last Come First Served (LCFS)

\[ c_1(x) = 0 \]
\[ c_2(x) = X_i \]
Theorem:

\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_iE[c_1(x)] + \lambda_iE[c_2(x)]) \]

**RESIDUAL** of the cycle length

\[ \rho = \lambda + \mu \]

i.i.d. contribution of work from each job arriving **EARLIER** in the cycle

i.i.d. contribution of work from each job arriving **LATER** in the cycle

**Example:** Last Come First Served (LCFS)

\[ c_1(x) = 0 \]
\[ c_2(x) = X_i \]

\[ E[D_i(x)] = E[R_{C_i}](1 + \rho_i) \]
\[ E[D_i] = E[R_{C_i}](1 + \rho_i) \]
**Theorem:**
\[ E[D_i(x)] = E[R_{C_i}] (1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

- **RESIDUAL** of the cycle length
- i.i.d. contribution of work from each job arriving **EARLIER** in the cycle
- i.i.d. contribution of work from each job arriving **LATER** in the cycle

**Example:** Processor Sharing (PS)
### Mean Value Analysis Framework

**Theorem:**
\[
E[D_i(x)] = E[R_{C_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)])
\]

- **RESIDUAL** of the cycle length
- i.i.d. contribution of work from each job arriving **EARLIER** in the cycle
- i.i.d. contribution of work from each job arriving **LATER** in the cycle

**Example:** Processor Sharing (PS)

\[
c_1(x) = \min(X_i, x)
\]

\[
c_2(x) = \min(X_i, x)
\]
Theorem:
\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

Example: Processor Sharing (PS)
\[ c_1(x) = \min(X_i, x) \]
\[ c_2(x) = \min(X_i, x) \]
\[ E[D_i] = E[R_{C_i}](1 + 2\lambda_i E[\min(X_{i,1}, X_{i,2})]) \]
**Mean Value Analysis Framework**

**Theorem:**

\[ E[D_i(x)] = E[R_{c_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

**Example:** Optimal = Shortest Job First (SJF)
**Theorem:**
\[
E[D_i(x)] = E[R_{c,i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)])
\]

- **RESIDUAL** of the cycle length
- i.i.d. contribution of work from each job arriving **EARLIER** in the cycle
- i.i.d. contribution of work from each job arriving **LATER** in the cycle

**Example:** Optimal = Shortest Job First (SJF)

\[
c_1(x) = X_i 1_{x_i < x}
\]
\[
c_2(x) = X_i 1_{x_i < x}
\]
Theorem:
\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_t E[c_1(x)] + \lambda_t E[c_2(x)]) \]

**Example:** Optimal = Shortest Job First (SJF)

\[
c_1(x) = X_i \mathbb{1}_{x_i < x} \]

\[
c_2(x) = X_i \mathbb{1}_{x_i < x} \]

\[ E[D_i] = E[R_{C_i}](1 + \lambda_t E[\min(X_{i,1}, X_{i,2})]) \]
Theorem:
\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

(policy dependent)
Theorem:

\[ E[D_i(x)] = E[R_{C_i}](1 + \lambda_i E[c_1(x)] + \lambda_i E[c_2(x)]) \]

still unknown

policy dependent
CALCULATING THE RESIDUAL CYCLE LENGTH

Notice that $E[R_{ci}]$
1. differ across $i$
2. are highly correlated across $i$
3. do not depend on the scheduling policy
CALCULATING THE RESIDUAL CYCLE LENGTH

Notice that $E[R_{C_i}]$
1. differ across $i$
2. are highly correlated across $i$
3. do not depend on the scheduling policy

Our approach:
1. Choose an easy policy: **FCFS**
2. Relate $E[R_{C_i}]$ to $E[L_{i,j}]$
3. Use recent mean value analysis of FCFS by Winands, Adan, & van Houtum

→ **Solution of $N(N+1)$ linear equations for the same # of unknowns yields** $E[R_{C_i}]$
What is the impact of scheduling?
What is the impact of scheduling?

Significant gains under heavy load, but smaller than in the M/GI/1.
Significant gains from scheduling, but not as large as in M/GI/1

Surprisingly simple results
Significant gains from scheduling, but not as large as in M/GI/1

Surprisingly simple results
WHERE DOES DELAY COME FROM?

Jobs that will arrive in the remaining part of the cycle

Jobs that arrived since the beginning of the cycle

The time waiting for the server to arrive
WHERE DOES DELAY COME FROM?

Jobs that will arrive in the remaining part of the cycle
...including arrivals once the server is at the queue

Jobs that arrived since the beginning of the cycle

The time waiting for the server to arrive
WHERE DOES DELAY COME FROM?

Jobs that will arrive in the remaining part of the cycle …including arrivals once the server is at the queue

Jobs that arrived since the beginning of the cycle

The time waiting for the server to arrive

Analysis and results become more complicated
What is the impact of scheduling?

Huge gains, matching those in the M/GI/1
**GATED**

Significant gains from scheduling, but not as large as in M/GI/1

Surprisingly simple results

**EXHAUSTIVE**

Huge gains from scheduling, match those in M/GI/1

Analysis is more complex, some policies are intractable: PS
**GATED**

Significant gains from scheduling, but not as large as in M/GI/1

Surprisingly simple results

**EXHAUSTIVE**

Huge gains from scheduling, match those in M/GI/1

Analysis is more complex, some policies are intractable: PS
DESIGN IMPACT OF GATED VS. EXHAUSTIVE

Non-preemptive Priority Scheduling:

2 priority classes

- High Priority: job size \( \leq t \)
- Low Priority: job size > \( t \)
DESIGN IMPACT OF GATED VS. EXHAUSTIVE

Non-preemptive Priority Scheduling:

2 priority classes

{ High Priority: job size ≤ t
  Low Priority: job size > t

What is the best choice of t?
DESIGN IMPACT OF GATED VS. EXHAUSTIVE

Non-preemptive Priority Scheduling:

2 priority classes

\{ 
\text{High Priority: job size } \leq t \\
\text{Low Priority: job size } > t 
\}

What is the best choice of $t$?

- balance loads of classes?
- choose the mean job size?
- imbalance in favor of small jobs?
- imbalance in favor of large jobs?
DESIGN IMPACT OF GATED VS. EXHAUSTIVE

Non-preemptive Priority Scheduling:

2 priority classes

- High Priority: job size \( \leq t \)
- Low Priority: job size \( > t \)

What is the best choice of \( t \)?

Gated: choose \( t = E[X_i] \)

Exhaustive: choose \( t = \frac{E[X_i] - \rho F_i(t)}{1 - \rho F_i(t)} \)
WRAP UP
OUR PAPER
Mean value analysis of scheduling in polling systems
OUR PAPER
Mean value analysis of scheduling in polling systems

1) Analyzing non-FCFS policies is possible
   ➔ some policies are still open: PS

2) Scheduling has significant impact
   ➔ behavior strongly dependent on polling policy
SCHEDULING IN POLLING SYSTEMS

Adam Wierman  
adamw@caltech.edu  
California Institute of Technology, USA

Erik Winands  Onno Boxma  
Technische Universiteit Eindhoven, the Netherlands

This work came out of an extended visit to the EURANDOM Institute in the Netherlands
<table>
<thead>
<tr>
<th>Priority Scheme</th>
<th>Exhaustive Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>( E[R_{C_i}](1 - \rho_i) )</td>
</tr>
<tr>
<td>LCFS</td>
<td>( E[R_{C_i}](1 - \rho_i) )</td>
</tr>
<tr>
<td>Preemptive LCFS</td>
<td>( E[R_{C_i}](1 - \rho_i) - \frac{\rho_i}{1 - \rho_i} \left( E[R_{X_i}] - E[X_i] \right) )</td>
</tr>
<tr>
<td>SJF</td>
<td>( E[R_{C_i}] \int_0^\infty \left( \frac{1 - \rho_i}{1 - \rho_i(x)} \right)^2 dF_i(x) )</td>
</tr>
<tr>
<td>SRPT</td>
<td>( E[D_i]^{SJF} ) – complicated term (see paper)</td>
</tr>
</tbody>
</table>
Exhaustive

\[
E[R_c_i](1 - \rho_i)
\]

FCFS

\[
E[R_c_i](1 - \rho_i)
\]

LCFS

\[
E[R_c_i](1 - \rho_i) - \frac{\rho_i}{1 - \rho_i} \left( E[R_x_i] - E[X_i] \right)
\]

Preemptive LCFS

\[
E[R_c_i] \int_0^{\infty} \left( \frac{1 - \rho_i}{1 - \rho_i(x)} \right)^2 dF_i(x)
\]

SJF

\[
E[D_i]^{SJF} - \text{complicated term (see paper)}
\]

SRPT

different than in gated systems
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>$E[R_{c_i}] (1 - \rho_i)$</td>
</tr>
<tr>
<td>LCFS</td>
<td>$E[R_{c_i}] (1 - \rho_i)$</td>
</tr>
<tr>
<td>Preemptive LCFS</td>
<td>$\frac{\rho_i (1 - \rho_i)}{1 - \rho_i} \left( E[R_{x_i}] - E[X_i] \right)$</td>
</tr>
<tr>
<td>SJF</td>
<td>$E[R_{c_i}] \int_0^\infty \left( \frac{1 - \rho_i}{1 - \rho_i(x)} \right)^2 dF_i(x)$</td>
</tr>
<tr>
<td>SRPT</td>
<td>$E[D_i]^{SJF}$ -- complicated term (see paper)</td>
</tr>
</tbody>
</table>

- \( E[D]^{PLCFS} \) - \( E[D]^{FCFS} \) different than in gated systems
- \( E[D]^{SJF} / E[D]^{FCFS} \) same as in M/GI/1
**EXAMPLES**

**FCFS**

\[ E[R_{c_i}](1 + \rho_i) \]

**Gated**

**Exhaustive**

\[ E[R_{c_i}](1 - \rho_i) \]
EXAMPLES

FCFS

Gated

\[ E[R_{c_i}](1 + \rho_i) \]

Exhaustive

\[ E[R_{c_i}](1 - \rho_i) \]

NOT EQUAL
**EXAMPLES**

Gated

\[ E[R_{c_i}](1 + \rho_i) \]

Exhaustive

\[ E[R_{c_i}](1 - \rho_i) \]

NOT EQUAL

---

FCFS

LCFS
## Examples

<table>
<thead>
<tr>
<th></th>
<th>Gated</th>
<th>Exhaustive</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>$E[R_{c,i}](1 + \rho_i)$</td>
<td>$E[R_{c,i}](1 - \rho_i)$</td>
</tr>
<tr>
<td>LCFS</td>
<td>$E[R_{c,i}](1 + \rho_i)$</td>
<td>$E[R_{c,i}](1 - \rho_i)$</td>
</tr>
<tr>
<td>Preemptive LCFS</td>
<td>same as LCFS</td>
<td>$E[R_{c,i}](1 - \rho_i) - \frac{\rho_i}{1 - \rho_i}(E[R_{x,i}] - E[X_{t,i}])$</td>
</tr>
</tbody>
</table>

---

NOT EQUAL
EXAMPLES

- FCFS: \( E[R_{C_i}] (1 + \rho_i) \)
- LCFS: \( E[R_{C_i}] (1 + \rho_i) \)
- Preemptive LCFS: \( E[R_{C_i}] (1 + \rho_i) \) (same as LCFS)
- SJF: \( E[R_{C_i}] (1 + \lambda_i E[\text{min}]) \)

Gated

Exhaustive

\( E[R_{C_i}] (1 - \rho_i) \)

\( E[R_{C_i}] (1 - \rho_i) \)

\( E[R_{C_i}] (1 - \rho_i) - \frac{\rho_i}{1 - \rho_i} \left( E[R_{X_i}] - E[X_i]\right) \)

\( E[R_{C_i}] \int_0^\infty \left( \frac{1 - \rho_i}{1 - \rho_i(x)} \right)^2 dF_i(x) \)

NOT EQUAL
EXAMPLES

- FCFS: $E[R_{c_i}](1 + \rho_i)$
- LCFS: $E[R_{c_i}](1 + \rho_i)$ (same as LCFS)
- Preemptive LCFS: $E[R_{c_i}](1 + \lambda_i E[\text{min}])$
- SJF: $E[R_{c_i}](1 + \lambda_i E[\text{min}])$
- SRPT: $E[R_{c_i}](1 + \lambda_i E[\text{min}])$ (same as SJF)

Gated:

- $E[R_{c_i}](1 + \rho_i)$

Exhaustive:

- $E[R_{c_i}](1 - \rho_i)$
- $\frac{E[R_{c_i}](1 - \rho_i) - \rho_i}{1 - \rho_i} (E[R_{X_i}] - E[X_{i}])$
- $E[R_{c_i}] \int_0^\infty \left(\frac{1 - \rho_i}{1 - \rho_i(x)}\right)^2 dF_i(x)$
- $E[D_i]^{SJF}$ – complicated term (see paper)