Optimization of polling systems with batch service

Jan-Pieter Dorsman, Rob van der Mei, Erik Winands

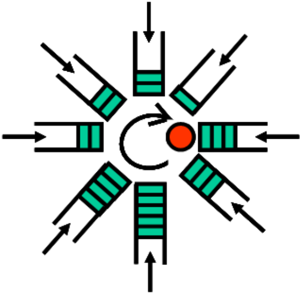
Technische Universiteit Eindhoven University of Technology

CWI Centrum Wiskunde & Informatica

11/27/2010

The polling model

A queueing system consisting of a number of queues and a single server, where there is a switch-over time when the server moves between queues.

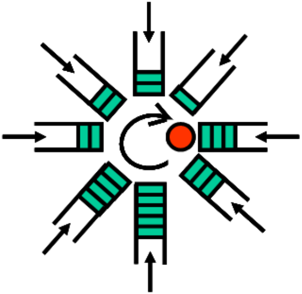


The analysis of this kind of system is important in many real-life applications:

- Manufacturing systems
- Computer communication systems
- Traffic systems

The polling model

A queueing system consisting of a number of queues and a single server, where there is a switch-over time when the server moves between queues.



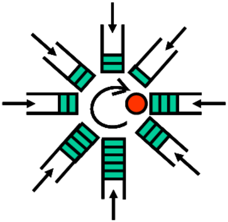
The analysis of this kind of system is important in many real-life applications:

- Manufacturing systems
- Computer communication systems
- Traffic systems

The polling model

We must decide

1. The order in which to serve the queues;
2. The number of customers to be served during a service period;
3. The order in which customers within each queue are served.



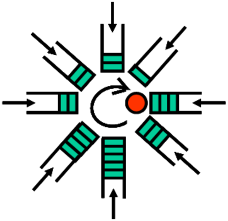
We will limit the analysis to the most common configuration:

1. Cyclic service
2. Exhaustive (server will switch iff queue is empty)
3. First come first served

The polling model

We must decide

1. The order in which to serve the queues;
2. The number of customers to be served during a service period;
3. The order in which customers within each queue are served.



We will limit
the analysis to the most common configuration:

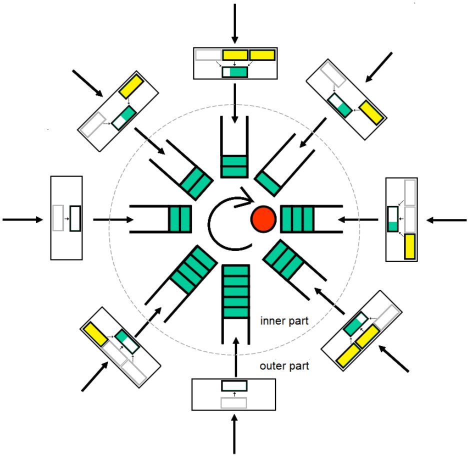
1. Cyclic service
2. Exhaustive (server will switch iff queue is empty)
3. First come first served

Extended model

We consider an extended version of this model.

Assumption

Processing time of a type-i batch is independent of its size!



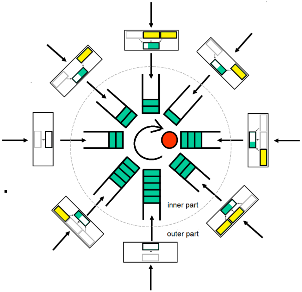
Examples:

- Oven with a fixed number of baking slots,
- Paint bath which paints several items at once.

A trade-off arises...

The waiting time of a customer can be decomposed in two parts:

- V_i, the time a type-i customer waits until its type-i batch is fully accumulated,
- W_i, the time a type-i customers waits in the inner system as part of a type-i batch.



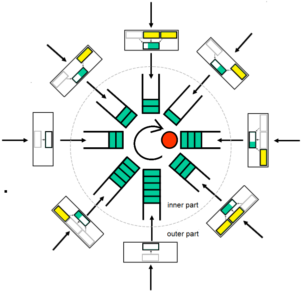
We are confronted with a trade-off when choosing batch sizes. D_i is the size of type-i batches:

- When taking D_i small, V_i will be low, but W_i will be high!
- When taking D_i large, W_i will be low, but V_i will be high!

A trade-off arises...

The waiting time of a customer can be decomposed in two parts:

- V_i, the time a type-i customer waits until its type-i batch is fully accumulated,
- W_i, the time a type-i customers waits in the inner system as part of a type-i batch.



We are confronted with a trade-off when choosing batch sizes. D_i is the size of type-i batches:

- When taking D_i small, V_i will be low, but W_i will be high!
- When taking D_i large, W_i will be low, but V_i will be high!

Problem description

The problem at hand

How to take batch sizes $\vec{D} = (D_1, \ldots, D_N)$ such that the weighted sum of the mean *total* waiting times is minimized?

We need to define a cost function:

$$C(\vec{D}) = \sum_{i=1}^{N} c_i(\mathbb{E}[V_i] + \mathbb{E}[W_i]).$$

Let E_{ij} be the event that an arriving type-i customer is the j^{th} taking place in the currently accumulating type-i batch.

$\mathbb{E}[V_i]$ is then easily computed by using

$$\mathbb{P}[E_{ij}] = \frac{1}{D_i} \text{ and } \mathbb{E}[V_i|E_{ij}] = \frac{D_i - j}{\lambda_i},$$

We end up with

$$C(\vec{D}) = \sum_{i=1}^{N} c_i \left(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_i] \right).$$

Problem description

The problem at hand

$$\arg\min_{\vec{D}} \sum_{i=1}^{N} c_i \left(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_i] \right)$$

No exact expression or even numerical method available to compute $\mathbb{E}[W_i]$. Two approaches possible:

- Approach 1: Use approximation by Boon et al., and approximate solution numerically,
- Approach 2: Use an even simpler approximation, $\mathbb{E}[W_{i,app}]$, and gain a closed-form approximation.

Boon et al.

Boon, Winands, Adan and Van Wijk (2009) provide an approximation of the mean waiting time for general load:

$$\mathbb{E}[W_{i,Boon}] = \frac{K_0 + K_{1,i}\rho + K_{2,i}\rho^2}{1-\rho},$$

where K_0, $K_{1,i}$ and $K_{2,i}$ are closed-form functions dependent on the polling system parameters and determined by three requirements:

Light traffic requirements:

1. $\mathbb{E}[W_{i,Boon}]|_{\rho=0} = \mathbb{E}[W_i]|_{\rho=0}$
2. $\frac{d}{d\rho}\mathbb{E}[W_{i,Boon}]|_{\rho=0} = \frac{d}{d\rho}\mathbb{E}[W_i]|_{\rho=0}$

Heavy traffic requirement:

3. $(1-\rho)\mathbb{E}[W_{i,Boon}]|_{\rho=1} = (1-\rho)\mathbb{E}[W_i]|_{\rho=1}$

Numerical Approach

Solve

$$\arg\min_{\vec{D}} C_{Boon}(\vec{D}) = \arg\min_{\vec{D}} \sum_{i=1}^{N} c_i \left(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,Boon}] \right)$$

numerically and round the obtained values.

Pro's:

- Very good performance.

Con's:

- Computation time may grow infeasibly long when the number of queues/dimensions increases.
- The numerical approach only gives limited insight in how the optimal batch sizes behave in the system's parameters.
- Implementation may be quite cumbersome.

Closed-form approach

$\mathbb{E}[W_{i,Boon}]$ is too complex to gain a nice closed-form approximation of optimal batch sizes.
Therefore, we use a simpler approximation of $\mathbb{E}[W_i]$ with the form:

$$\mathbb{E}[W_{i,app}] = \frac{a + b_i \rho}{1 - \rho}$$

Coefficients a and b_i are determined by two requirements:

- LT: $\mathbb{E}[W_{i,app}]|_{\rho=0} = \mathbb{E}[W_i]|_{\rho=0}$
- HT: $(1-\rho)\mathbb{E}[W_{i,app}]|_{\rho=1} = (1-\rho)\mathbb{E}[W_{i,simplification}]|_{\rho=1}$

We end up with the following approximative cost function

$$C_{app}(\vec{D}) = \sum_{i=1}^{N} c_i \left(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,app}] \right)$$

Closed-form approach

Solve

$$\arg\min_{\vec{D}} C_{app}(\vec{D}) = \arg\min_{\vec{D}} \sum_{i=1}^{N} c_i \left(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,app}] \right)$$

with ordinary calculus.

- The one-dimensional case is easily solved.
- For the multi-dimensional case, the system of equations $\frac{d}{dD_i} C_{app}(\vec{D}) = 0 \ \forall i$ does not result in nice expressions.

Work-around: reduce multi-dimensional problem to a one-dimensional problem by a priori assuming ratios between optimal batch sizes.

Closed-form approach

Solve

$$\arg\min_{\vec{D}} C_{app}(\vec{D}) = \arg\min_{\vec{D}} \sum_{i=1}^{N} c_i \left(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,app}] \right)$$

with ordinary calculus.

- The one-dimensional case is easily solved.
- For the multi-dimensional case, the system of equations $\frac{d}{dD_i} C_{app}(\vec{D}) = 0 \; \forall i$ does not result in nice expressions.

Work-around: reduce multi-dimensional problem to a one-dimensional problem by a priori assuming ratios between optimal batch sizes.

Closed-form approach

The problem at hand

$$\arg\min_{\vec{D}} C_{app}(\vec{D}) = \arg\min_{\vec{D}} \sum_{i=1}^{N} c_i(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,app}]).$$

As batch sizes increase, the load of the inner system drops rapidly, hence $\mathbb{E}[W_{i,app}]$ will be near-insensitive to the batch sizes.

The *ratios* of the optimal variables of the following problem *can* be determined:

$$\arg\inf_{\vec{D}} C_{app}(\vec{D}) = \arg\inf_{\vec{D}} \sum_{i=1}^{N} c_i \frac{D_i - 1}{2\lambda_i}.$$

I.e., the numbers $d_2 = \frac{D_2^{opt}}{D_1^{opt}}, \ldots, d_N = \frac{D_N^{opt}}{D_1^{opt}}$ are known here:

$$d_i = \frac{\lambda_i}{\lambda_1}\sqrt{\frac{c_1\mathbb{E}[B_i]}{c_i\mathbb{E}[B_1]}}$$

Closed-form approach

The problem at hand

$$\arg\min_{\vec{D}} C_{app}(\vec{D}) = \arg\min_{\vec{D}} \sum_{i=1}^{N} c_i \Big(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,app}] \Big).$$

As batch sizes increase, the load of the inner system drops rapidly, hence $\mathbb{E}[W_{i,app}]$ will be near-insensitive to the batch sizes.

The *ratios* of the optimal variables of the following problem *can* be determined:

$$\arg\inf_{\vec{D}} C_{app}(\vec{D}) = \arg\inf_{\vec{D}} \sum_{i=1}^{N} c_i \frac{D_i - 1}{2\lambda_i}.$$

I.e., the numbers $d_2 = \frac{D_2^{opt}}{D_1^{opt}}, \ \ldots \ , d_N = \frac{D_N^{opt}}{D_1^{opt}}$ are known here:

$$d_i = \frac{\lambda_i}{\lambda_1} \sqrt{\frac{c_1 \mathbb{E}[B_i]}{c_i \mathbb{E}[B_1]}}$$

Closed-form approach

The problem at hand

$$\arg\min_{\vec{D}} C_{app}(\vec{D}) = \arg\min_{\vec{D}} \sum_{i=1}^{N} c_i\Big(\frac{D_i - 1}{2\lambda_i} + \mathbb{E}[W_{i,app}]\Big).$$

As batch sizes increase, the load of the inner system drops rapidly, hence $\mathbb{E}[W_{i,app}]$ will be near-insensitive to the batch sizes.

The *ratios* of the optimal variables of the following problem *can* be determined:

$$\arg\inf_{\vec{D}} C_{app}(\vec{D}) = \arg\inf_{\vec{D}} \sum_{i=1}^{N} c_i \frac{D_i - 1}{2\lambda_i}.$$

I.e., the numbers $d_2 = \frac{D_2^{opt}}{D_1^{opt}}, \ \ldots \ , d_N = \frac{D_N^{opt}}{D_1^{opt}}$ are known here:

$$d_i = \frac{\lambda_i}{\lambda_1}\sqrt{\frac{c_1 \mathbb{E}[B_i]}{c_i \mathbb{E}[B_1]}}$$

Closed-form approach

Back to our problem, we write $D_1^{opt} = E, D_2^{opt} = d_2 E, \ldots, D_N^{opt} = d_N E$ with the d_i as found in the simple problem. Only optimization in E is needed:

$$E_{app}^{opt} = \sum_{i=1}^{N} \frac{\lambda_i \mathbb{E}[B_i]}{d_i} + \sqrt{2 \left(\sum_{i=1}^{N} \frac{c_i d_i}{\lambda_i} \right)^{-1} \left(\sum_{i=1}^{N} c_i \omega_{i,app} \right) \left(\sum_{i=1}^{N} \frac{\lambda_i \mathbb{E}[B_i]}{d_i} \right)}.$$

Going back to optimal batch sizes by multiplying with the d_i, we have:

$$\begin{aligned}\vec{D}_{app}^{opt} =& (D_{1,app}^{opt}, D_{2,app}^{opt}, \ldots, D_{N,app}^{opt}) \\ =& (E_{app}^{opt}, \frac{\lambda_2}{\lambda_1} \sqrt{\frac{c_1 \mathbb{E}[B_2]}{c_2 \mathbb{E}[B_1]}} E_{app}^{opt}, \ \ldots \ , \frac{\lambda_N}{\lambda_1} \sqrt{\frac{c_1 \mathbb{E}[B_N]}{c_N \mathbb{E}[B_1]}} E_{app}^{opt}).\end{aligned}$$

Proper rounding of these values gives a nicely defined and well-performing closed-form approximation.

Closed-form approach

The obtained closed-form approximation...

Pro's:

- Performs increasingly well when the number of queues or switch-over times get larger,
- Is easily implemented and requires virtually no computation time,
- Gives insights in the dependency of the optimal batch sizes on the systems parameters.

Con's:

- Performs not as well as the numerical approach in case of small numbers of queues or small switch-over times. Complementary effect!

Validation

Validation by means of simulation, based on testbeds of symmetric and asymmetric systems, with a total of 1260 systems.

Differences expressed relatively in terms of costs in %.

Overall:

Test bed	Numerical	Closed-form
Symmetric	0.303	1.624
Asymmetric	0.645	4.848

Categorized in the number of queues:

Test bed	N=2		N=5	
	Numerical	Closed-form	Numerical	Closed-form
Symmetric	0.428	2.776	0.178	0.471
Asymmetric	0.870	7.715	0.419	1.981

Further Research

The current analysis may be extended in multiple directions:

- Optimization based on higher moments and tail probabilities,
- Model variations: non-cyclic routing, other service disciplines, size-dependent service requirements, introduction of the possibility to idle, etc.
- Rounding strategy of the fractional D_i-values.