Size-aware Dispatching Problems in MDP Framework

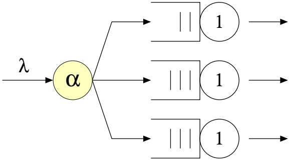
Esa Hyytiä, Aleksi Penttinen, Samuli Aalto, Jorma Virtamo

Department of Communications and Networking
Aalto University, School of Electrical Engineering, Finland

6.7.2011, INFORMS APS

Aalto University
School of Electrical
Engineering

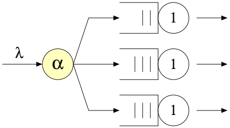
Dispatching Problem to Parallel Queues



- Upon arrival a job is routed to one of the *m* servers
- Each server processes jobs according to a certain scheduling discipline (e.g., FCFS)
- Objective: minimize the mean delay (mean sojourn time)
- Examples: manufacturing sites, job assignment in supercomputing, traffic routing, web-server farms, and other distributed computing systems

Aalto University
School of Electrical Engineering

Size- and State-aware Dispatching Problem



- Poisson arrival process, rate λ
- *m* parallel heterogeneous servers
- General job size distribution
- Service requirements become known upon arrival (possibly server specific)
- Queue states (job sizes and their service order) are known
- Scheduling discipline known: FCFS, LCFS, SPT, SRPT
- Dispatching policy α chooses the queue upon arrival
- Objective: minimize the mean delay

Aalto University
School of Electrical Engineering

State-independent Policies

1. **Bernoulli splitting (RND)**:
   Choose queue in random using probabilities p_i
2. **Size-Interval-Task-Assignment (SITA)**:
   “short jobs to one queue and rest to another”
   - Proposed in Crovella et. al (Sigmetrics’98) and Harchol-Balter et. al (J. of PDC, vol. 59, 1999).
   - SITA-E uses such intervals that balance the load.
   - Optimal size-aware state-free for FCFS (Feng et. al, 2005).

Aalto University
School of Electrical Engineering

State-dependent Policies

1. **Join-the-Shortest-Queue (JSQ)**:
   Optimal when Poisson arrivals, Exp-distributed job sizes, identical servers, and only the queue occupancy is known (Winston, 1977).
2. **Round-robin (RR)**:
   Optimal with identical servers that were initially in a same state (Ephremides et. al, 1980).
3. **Least-Work-Left (LWL)**:
   Pick the queue with the shortest backlog (Sharifnia, 1997).

Aalto University
School of Electrical
Engineering

Approach: MDP and FPI

- Size- and state-aware setting; future arrivals not known
- Idea: start with a reasonable basic dispatching policy, and carry out the first policy iteration (FPI) step
- Policy iteration finds the optimal policy, and the FPI step typically yields the highest improvement.
- Requires the relative values of states $v_{\mathbf{z}}$
- However, our state-space is quite complex (job sizes etc.)

Aalto University
School of Electrical Engineering

Delay Costs and Relative Value

- Delay costs are accrued at rate

$$N_{\mathbf{z}}(t) \triangleq \text{"the number of jobs in the system"},$$

where $\mathbf{z}$ denotes the initial state at time $t = 0$.
- The delay costs accrued during $(0, t)$ are

$$V_{\mathbf{z}}(t) \triangleq \int_0^t N_{\mathbf{z}}(s)\, ds.$$

- The relative value is the expected difference in the cumulative costs between a system initially in state $\mathbf{z}$ and a system initially in equilibrium,

$$\begin{aligned} v_{\mathbf{z}} &\triangleq \lim_{t\to\infty} \mathrm{E}[V_{\mathbf{z}} - r\,t] \\ &= \lim_{t\to\infty} \left(\mathrm{E}\Big[\int_0^t N_{\mathbf{z}}(s)\, ds\Big] - \mathrm{E}[N]\, t \right). \end{aligned}$$

Aalto University
School of Electrical Engineering

First Policy Iteration (FPI)

- Assume: Relative values $v_{\mathbf{z}}$ available (for basic policy)
- Improved decision according to FPI at state $\mathbf{z}$:

$$\alpha(\mathbf{z}, x) \triangleq \underset{i}{\operatorname{argmin}} \, \left(v_{\mathbf{z}'(i)} - v_{\mathbf{z}} \right),$$

  where $\mathbf{z}'(i)$ is the new state if job x is added to queue i.

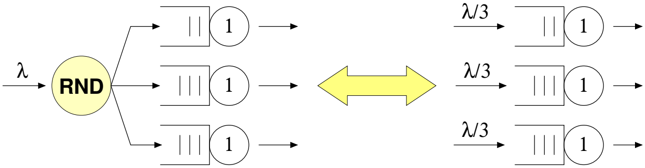
"choose the action with the smallest expected future cost"

- Recall: in addition to $\mathbf{z}$, relative value $v_{\mathbf{z}}$ depends also on
  1. basic dispatching policy
  2. scheduling discipline
  3. arrival rate λ, and
  4. job size distribution.

Aalto University
School of Electrical Engineering

Decomposition to Independent M/G/1 Queues

- Deriving relative values is generally difficult task.
- However, any state-independent policy feeds each server jobs according to a Poisson process (cf. Bernoulli split)



- Analyze single M/G/1 queues instead?

Aalto University
School of Electrical Engineering

Relative Values for Single M/G/1 Queue

Plan:

- Assume a state-independent basic policy.
- Derive relative values for the "isolated queues" first.
- Relative value of the whole system for any state-independent policy is the sum of the queue specific relative values:

$$v_{\mathbf{z}} = \sum_i v_{\mathbf{z}_i}.$$

- Carry out the FPI step $\Rightarrow$ new efficient policy.
  (In practice, it is sufficient to know, e.g., $v_{\mathbf{z}} - v_0$.)

Next step:

$\boxed{\text{Derive } v_{\mathbf{z}} - v_0 \text{ for an M/G/1 queue with LCFS, FCFS, SPT and SRPT.}}$

Aalto University
School of Electrical Engineering

M/G/1-LCFS (preemptive)

Notation:

- λ is the Poisson arrival rate.
- $\rho = \lambda \mathrm{E}[X]$ and $\mathrm{E}[X]$ denotes the mean job size.
- $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denotes the state, where Δ_i is the known (remaining) service time of job i, $i = 1, .., n$.
- The nth job is the latest arrival currently being processed.

Proposition: The size-aware relative value of state $\mathbf{z}$ with respect to delay in an M/G/1-LCFS queue is

$$\boxed{v_{(\Delta_1;..;\Delta_n)} - v_0 = \frac{1}{1-\rho} \sum_{i=1}^{n} i \cdot \Delta_i .}$$

Insensitivity: $v_{(\Delta_1;..;\Delta_n)} - v_0$ depends only on ρ.

Aalto University
School of Electrical Engineering

Proof

- Consider two systems under same arrivals:
  1. S1 initially in state $\mathbf{z} = (\Delta_1, .., \Delta_n)$,
  2. S2 initially empty.
- Let D_i denote the (remaining) delay of job i in S1.
- With LCFS, the current state has no effect on the future arrivals' sojourn times.
- The difference between the relative value of S1 and S2 is equal to the mean remaining delay of the n present jobs,

$$v_{(\Delta_1;..;\Delta_n)} - v_0 = \sum_{i=1}^{n} \mathrm{E}[D_i].$$

Aalto University
School of Electrical Engineering

- Remaining delay D_n of job n is given by a random sum,

$$D_n = \Delta_n + \left(B_1 + \ldots + B_{A(\Delta_n)}\right)$$

where $A(\Delta_n)$ denotes the number of (mini) busy periods during time Δ_n, and B_i the corresponding durations,

$$\mathrm{E}[B_i] = \mathrm{E}[X]/(1-\rho).$$

- Taking the expectation on both sides gives

$$\mathrm{E}[D_n] = \Delta_n + \mathrm{E}[A(\Delta_n)] \cdot \mathrm{E}[B] = \frac{\Delta_n}{1-\rho}.$$

- Similarly, $\mathrm{E}[D_i] = (1-\rho)^{-1} \sum_{j=i}^{n} \Delta_j.$

$$\Rightarrow \quad v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n} \mathrm{E}[D_i] = \boxed{\frac{1}{1-\rho} \sum_{i=1}^{n} i \cdot \Delta_i.}$$

Aalto University
School of Electrical Engineering

M/G/1-FCFS

Notation:

- λ is the Poisson arrival rate.
- $\rho = \lambda\,\mathrm{E}[X]$ and $\mathrm{E}[X]$ denotes the mean job size.
- Let $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denote the state, where Δ_i is the known (remaining) service time of job i, $i = 1, .., n$.
- The nth job is served first.

Proposition: The size-aware relative value of state $\mathbf{z}$ with respect to the delay in an M/G/1-FCFS queue is given by

$$\boxed{v_{(\Delta_1;..;\Delta_n)} - v_0 = \frac{\lambda\, u_{\mathbf{z}}^2}{2(1-\rho)} + \sum_{i=1}^{n} i\,\Delta_i,}$$

where $u_{\mathbf{z}} = \sum_i \Delta_i$ denotes the backlog in the queue.

Insensitivity: $v_{(\Delta_1;..;\Delta_n)} - v_0$ depends only on λ and $\mathrm{E}[X]$.

Aalto University
School of Electrical Engineering

Proof

Consider two systems under the same arrivals:

- S1 initially in state $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ and
- S2 initially empty.

Both systems behave identically once S1 becomes empty. The difference in the relative values is equal to the additional time jobs spent in S1,

$$v_{\mathbf{z}} - v_0 = V_1 + V_2,$$

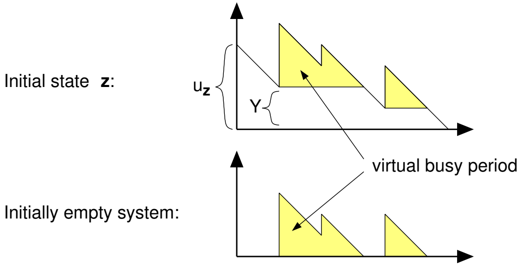
where V_1 denotes the (remaining) delay of present jobs, and V_2 the additional mean delay the later arrivals experience in S1.

The total delay of the *n* present jobs in S1 is already fixed,

$$V_1 = \sum_{i=1}^{n} i\,\Delta_i.$$

Aalto University
School of Electrical Engineering

- A later arriving task starts a busy period in S2, which corresponds to a mini busy period in S1.

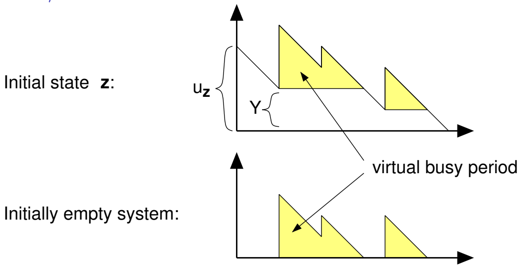


- During busy periods, arriving jobs increase the cumulative delay by an amount equal to the post arrival workload.
- These jobs experience an additional delay Y in S1.
- Otherwise the delay contributions are equal!

Aalto University
School of Electrical Engineering

Summing up:

- Mean number of busy periods before S1 empty: $\lambda\, u_{\mathbf{z}}$.
- Mean number of jobs served during a busy period: $1/(1-\rho)$.
- The mean offset $\mathrm{E}[Y] = u_{\mathbf{z}}/2$.



Therefore,

$$V_2 = \lambda\, u_{\mathbf{z}} \cdot \frac{1}{1-\rho} \cdot \frac{u_{\mathbf{z}}}{2} = \frac{\lambda\, u_{\mathbf{z}}^2}{2(1-\rho)},$$

and $V_1 + V_2 = v_{\mathbf{z}} - v_0$, which completes the proof.

Aalto University
School of Electrical Engineering

M/G/1-SPT (Non-preemptive)

Notation

- $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denotes the state of a queue; job n is currently receiving service, jobs $1, \ldots, (n-1)$ wait in the queue, so that $\Delta_1 > \Delta_2 > \ldots > \Delta_{n-1}$. (SPT order)
- Let $f(x)$ denote the job size pdf.
- $\rho(x) = \lambda \int_0^x x\, f(x)\, dx$, i.e., load due to jobs shorter than x.
- Define $\tilde{\Delta}_0 = \infty$, $\tilde{\Delta}_n = 0$ and $\tilde{\Delta}_i = \Delta_i$ for $i = 1, \ldots, (n-1)$.

$$\boxed{\begin{aligned} v_{(\Delta_1;..;\Delta_n)} - v_0 = \sum_{i=1}^{n} \left(\Delta_i + \frac{\sum_{j=i+1}^{n} \Delta_j}{1-\rho(\Delta_i)} \right) + \\ \frac{\lambda}{2} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{i-1} \Delta_j^2 + (\sum_{j=i}^{n} \Delta_j)^2 \right) \int_{\tilde{\Delta}_i}^{\tilde{\Delta}_{i-1}} \frac{f(x)}{(1-\rho(x))^2}\, dx \right). \end{aligned}}$$

Aalto University
School of Electrical Engineering

M/G/1-SRPT

The size-aware relative value of state **z** with respect to delay in an M/G/1-SRPT queue is

$$\boxed{\begin{aligned} v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n} \left(\Delta_i + \frac{u_{\mathbf{z}}(\Delta_i)}{1-\rho(\Delta_i)} + \int_0^{\Delta_i} \frac{\rho(t)}{1-\rho(t)}\, dt \right) \\ + \int_0^{\infty} \frac{\lambda\, f(x)\, \left(u_{\mathbf{z}}(x)^2 + n_{\mathbf{z}}(x)\, x^2\right)}{2(1-\rho(x))^2}\, dx, \end{aligned}}$$

where

- $f(x)$ = job size pdf,
- $\rho(x)$ = offered load due to jobs shorter than x,
- $u_{\mathbf{z}}(x)$ = backlog due to jobs shorter than x in state **z**,
- $n_{\mathbf{z}}(x)$ = number of jobs longer than x in state **z**.

Aalto University
School of Electrical Engineering

SITA with Switch

First application of the relative values:

- Consider a SITA policy with identical servers
- The role of any two servers can be exchanged, e.g., after a state change (arrival)
- Relative values tell us when this is beneficial

Example:

Two identical FCFS queues and SITA-E (equal load per queue)

Switch: $\boxed{\text{short jobs to queue with smaller backlog.}}$

$\Rightarrow$ New policy: SITA-E with Switch (SITA-Es)

(Generalizes to $n > 2$ queues)

Aalto University
School of Electrical Engineering

Numerical Examples

Example 1

- Two identical queues with FCFS
- Job size distribution:
  1. Uniform U(0,2)
  2. Exponential Exp(1)
  3. Pareto(β) with $\beta = 3$: $\quad \mathrm{P}\{X > t\} = (1 + t)^{-\beta}$
- Performance metrics:
  1. Absolute mean delay (sojourn time)
  2. Relative delay when compared to SITA-E policy

Example 2

- Two identical queues with SRPT
- Exponential job size distribution, Exp(1)
- Relative delay when compared to a single shared SRPT queue processed by two identical servers

Aalto University
School of Electrical Engineering

FCFS and Uniformly distributed jobs

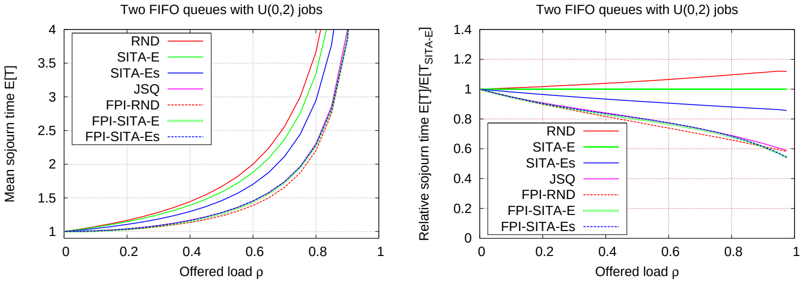
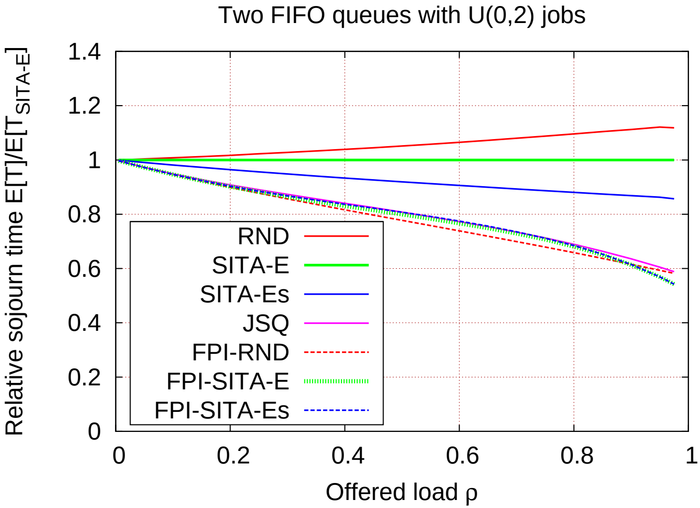


Figure: Mean delay under FCFS with uniformly distributed job sizes.

Aalto University
School of Electrical Engineering

FCFS and Uniformly distributed jobs



FCFS and Exponentially distributed jobs

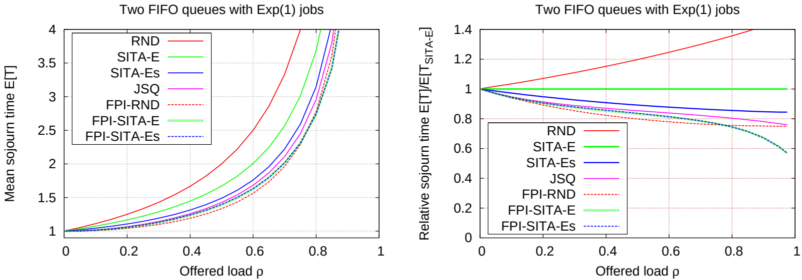


Figure: Mean delay under FCFS with Exp-distributed job sizes.

Aalto University
School of Electrical Engineering

FCFS and Exponentially distributed jobs

Aalto University
School of Electrical Engineering

FCFS and Pareto distributed jobs

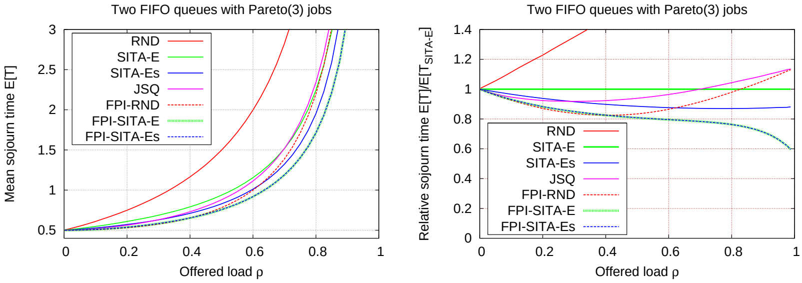
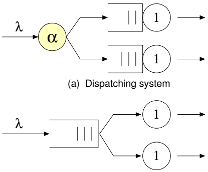


Figure: Mean delay under FCFS with Pareto distributed job sizes.

Aalto University
School of Electrical Engineering

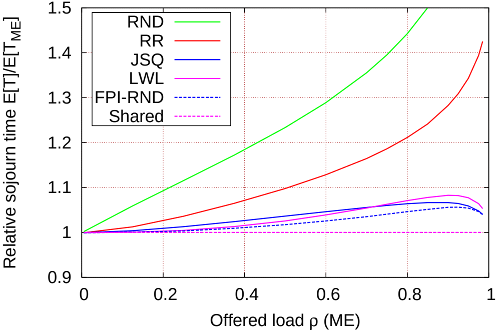
FCFS and Pareto distributed jobs

SRPT and Exponentially distributed jobs



(a) Dispatching system

(b) Multi–server system



- Dispatching system vs. a shared queue with SRPT.
- Appears that the disadvantage due to the dispatching can be insignificant (here order of 5% with FPI-RND).

Aalto University
School of Electrical Engineering

Conclusions

- Size- and state-aware dispatching problem can be approached in MDP framework
- Corresponding relative values required for the FPI step.
- For state-independent basic policies, sufficient to analyze M/G/1 queue in isolation
- Size-aware relative values for FCFS, LCFS, SPT and SRPT with respect to delay are available for M/G/1
- For FCFS and LCFS, the relative values are insensitive to job size distribution
- For SPT and SRPT, the relatives values in integral form.
- Robust, efficient and state-dependent dispatching policies taking into account the current and later arriving tasks

Thanks!

Aalto University
School of Electrical Engineering

References:

1. E. Hyytiä, A. Penttinen and S. Aalto, *Size- and State-Aware Dispatching Problem with Queue-Specific Job Sizes*, December, 2010, (submitted).
2. E. Hyytiä, J. Virtamo, S. Aalto and A. Penttinen, *M/M/1-PS Queue and Size-Aware Task Assignment*, in IFIP PERFORMANCE, October 2011, Amsterdam, Netherlands, (to appear).
3. E. Hyytiä, A. Penttinen, S. Aalto and J. Virtamo, *Dispatching problem with fixed size jobs and processor sharing discipline*, in 23rd International Teletraffic Congress (ITC'23), September 2011, San Fransisco, USA, (to appear).

Aalto University
School of Electrical
Engineering