Aalto University
School of Electrical Engineering

Size-aware MDP approach to dispatching problems

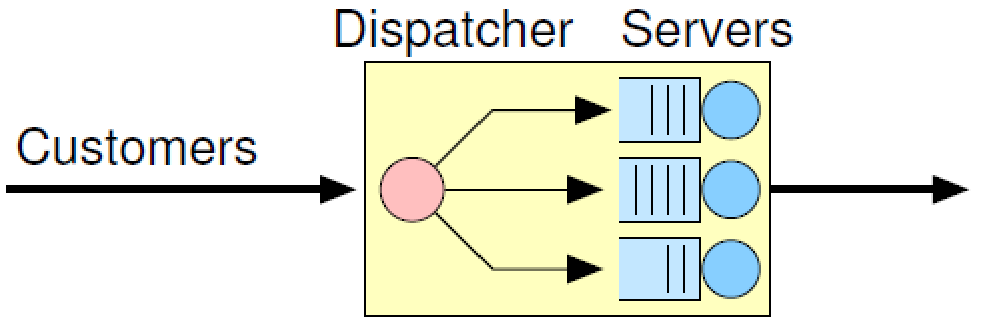
Esa Hyytiä & Samuli Aalto
Aalto University, Finland

XXVI EURO-INFORMS
1–4 July 2013
Rome, Italy

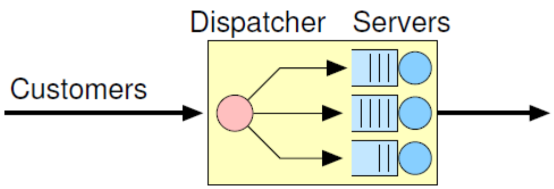
Part I
Dispatching

Dispatching problem

- Dispatching = Task assignment = Routing
  - random customer arrivals with random service requirements
  - dispatching decision made upon the arrival
    - no jockeying amonst the queues allowed
  - minimize e.g. the mean delay (i.e. latency, sojourn time)
  - ICT applications: web server farms, supercomputer grids, etc.

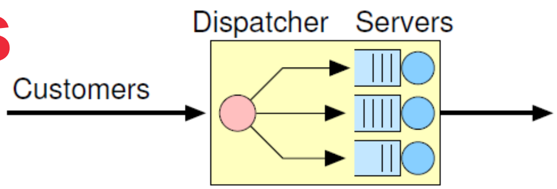


Static dispatching policies



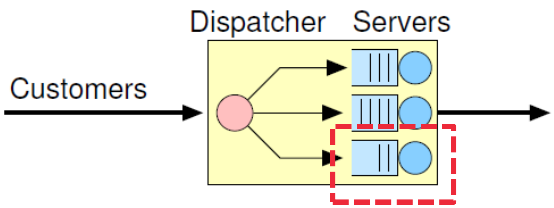
- RND = Bernoulli splitting
  - choose the queue randomly (according to the given distribution)
  - no state or size information needed
- SITA = Size Interval Task Assignment
  - choose the queue with similar customers (according to the given service time thresholds)
  - based on the service time of the arriving customer, no state information needed
  - SITA-E uses thresholds that balance the load in each server
  - Harchol-Balter et al. (1999), Feng et al. (2005)

Dynamic dispatching policies



- JSQ = Join the Shortest Queue
  - choose the queue with the smallest number of customers
  - state information needed
  - Haight (1958), Winston (1977)
- LWL = Least Work Left
  - choose the queue with the smallest workload
  - more detailed state information needed
  - Harchol-Balter et al. (1999)

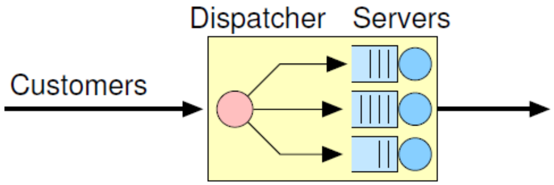
Scheduling policies



- Scheduling policy = service policy = queueing discipline
  - applied in each queue separately

- FCFS = First Come First Served
  - serve the customer who arrived first (”ordinary queue”)
  - vulnerable to very long service times
  - e.g. supercomputing settings with non-preemptible jobs
- PS = Processor Sharing
  - serve customers parallelly with equal shares (”fair queue”)
  - insensitive to the service time disribution
  - e.g. web server farms with time-sharing servers
- SRPT = Shortest Remaining Processing Time
  - serve the customer who has the shortest remaining service time
  - minimizes the queue length at any time in each sample path

Optimality results



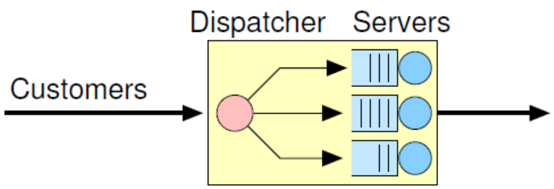
- JSQ optimal for any arrival process, homogeneous servers, and exponential (or IFR) service times
  - Winston (1977), Weber (1978), Ephremides et al. (1980)
- LWL optimal for Poisson arrivals, homogeneous FCFS [PS] servers, and deterministic service times (= JSQ!)
  - Hyytiä et al. (2011a)
- SITA optimal for Poisson arrivals and homogeneous FCFS servers
  - Feng et al. (2005)
- RND optimal for Poisson arrivals and homogeneous PS servers
  - Altman et al. (2011)

References until now

- Haight (1958)
  Two Queues in Parallel, *Biometrika*
- Winston (1977)
  Optimality of the shortest line discipline, *JAP*
- Weber (1978)
  On the optimal assignment of customers to parallel servers, *JAP*
- Ephremides, Varaiya & Walrand (1980)
  A simple dynamic routing problem, *IEEE TAC*
- Harchol-Balter, Crovella & Murta (1999)
  On choosing a task assignment policy for a distributed server system, *JPDC*
- Feng, Misra & Rubenstein (2005)
  Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems, *PEVA*
- Altman, Ayesta & Prabhu (2011)
  Load balancing in processor sharing systems, *TS*

Part II
MDP approach

MDP approach



- Assume Poisson arrivals
- Any static policy (RND, SITA) results in parallel M/G/1 queues
  - due to the splitting property of the Poisson process
- Fix the static policy and determine the relative values for all these parallel M/G/1 queues starting in any initial state
- Evaluate the decision to dispatch an arriving customer to a queue in a given state by utilizing these relative values
- Dispatch an arriving customer to the queue that minimizes the mean additional costs
- As the result, you get a better dynamic policy
- This is called First Policy Iteration (FPI) in the MDP theory

Value function

- Fix the policy and cost structure
- Assume a stable system
- $Z(t)$ = state of the system at time t
- $C(t)$ = cost rate at time t
- $V(t)$ = cumulative cost at time t

$$V(t) = \int_0^t C(u)\,du$$

Value function

- Definition: For a fixed policy resulting in a stable system, the value function $v_{\mathbf{z}}$ gives the expected difference in the infinite horizon cumulative costs between
  - the system initially in state $\mathbf{z}$, and
  - the system initially in equilibrium,

$$v_{\mathbf{z}} = \lim_{t\to\infty} E[V(t) - rt \mid Z(0) = \mathbf{z}]$$

- Here r = average cost rate (in the long run)

Relative value

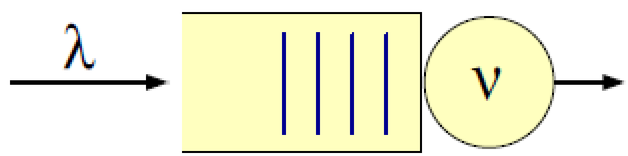
- Definition: For a fixed policy resulting in a stable system, the relative value $v_{\mathbf{z}} - v_{\mathbf{0}}$ gives the expected difference in the infinite horizon cumulative costs between
  - the system initially in state $\mathbf{z}$, and
  - the system initially in state $\mathbf{0}$,

$$v_{\mathbf{z}} - v_{\mathbf{0}} = \lim_{t\to\infty}(E[V(t) \mid Z(0)=\mathbf{z}] - E[V(t) \mid Z(0)=\mathbf{0}])$$

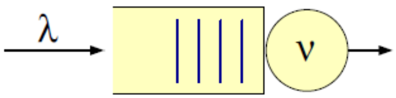
Part III
Size-awareness

M/G/1 queue

- Poisson arrivals with rate λ
- General IID service times with mean $E[S] = 1/\nu$
- Single server with load $\rho = \lambda E[S]$



Size-aware M/G/1-FCFS queue

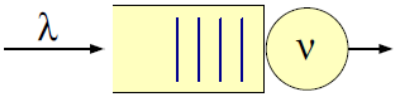


- Poisson arrivals with rate λ
- General IID service times with mean $E[S] = 1/\nu$
- Single server with load $\rho = \lambda E[S]$
- FCFS scheduling discipline
- State description:

$$\mathbf{z} = (\Delta_1, \ldots, \Delta_n)$$

  – Δ_i = (remaining) service time of customer i
  – n = customer in service (i.e., the ”oldest” one)

Size-aware value function for M/G/1-FCFS



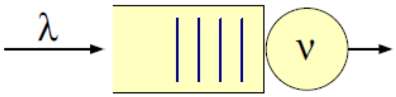
- When the objective is to minimize the mean delay, the cost rate $C(t)$ equals the queue length $N(t)$ so that

$$v_{\mathbf{z}} = \lim_{t\to\infty} \int_0^t \left(E[N(u) \mid Z(0) = \mathbf{z}] - E[N] \right) du$$

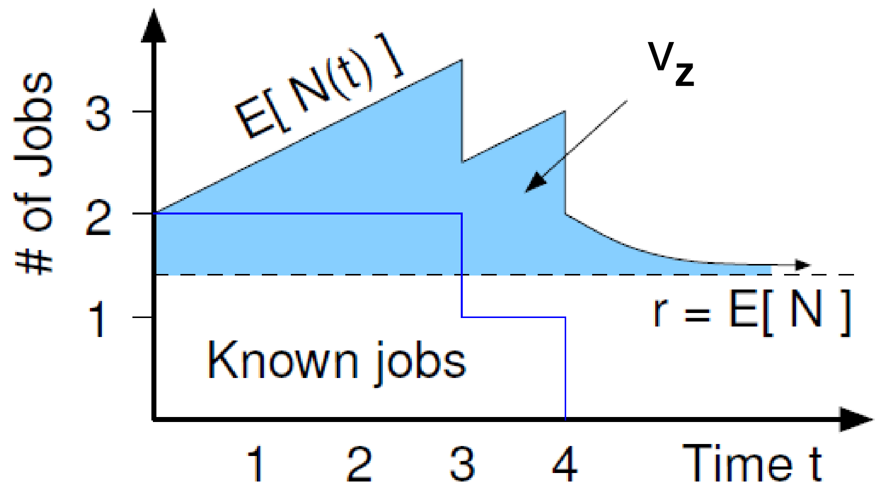
- Here by the Pollaczek-Khintchin formula,

$$E[N] = \lambda \left(E[S] + \frac{\lambda E[S^2]}{2(1-\rho)} \right)$$

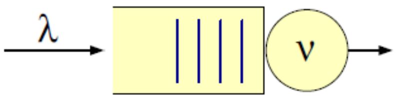
Size-aware value function for M/G/1-FCFS



- Example: Initial state $\mathbf{z} = (1,3)$
  - customer in service with remaining service time 3
  - one waiting customer with service time 1



Size-aware relative value for M/G/1-FCFS



- Proposition [Hyytiä et al. (2012a)]:

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

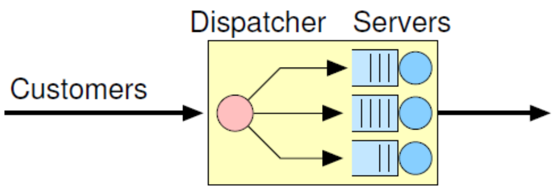
  – where

$$\mathbf{z} = (\Delta_1, \ldots, \Delta_n), \quad u_{\mathbf{z}} = \sum_{i=1}^{n} \Delta_i, \quad \rho = \lambda E[S] < 1$$

  – $u_{\mathbf{z}}$ = initial workload

- Note: Relative value insensitive to service time distribution

Size-aware MDP approach

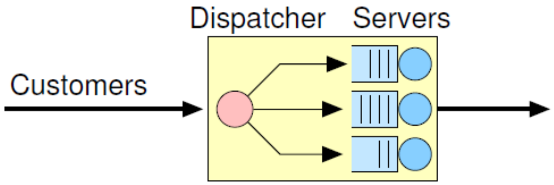


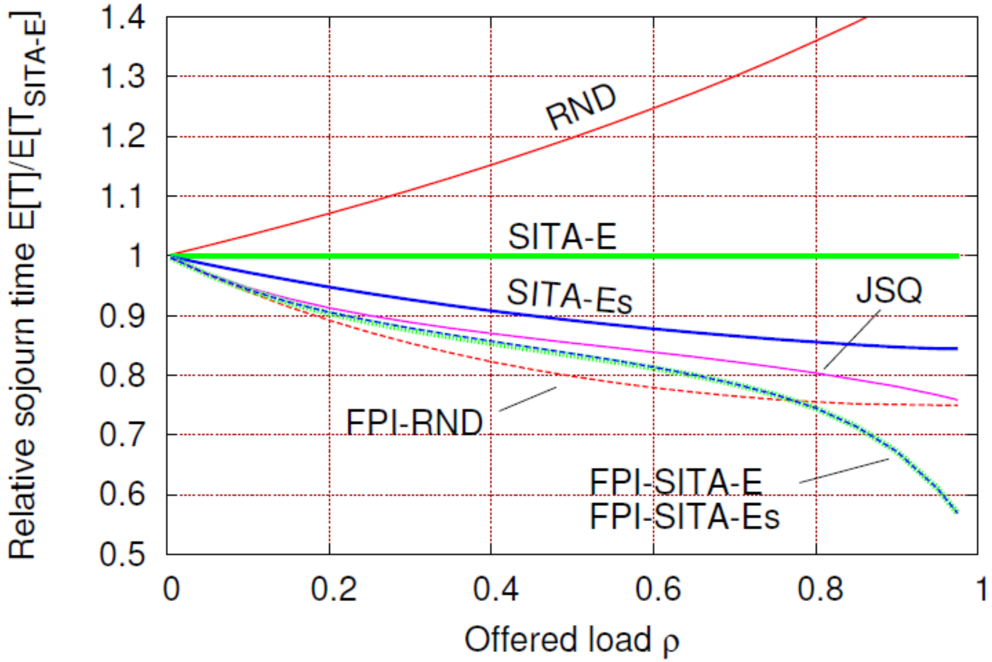
- Example: Dispatching problem with
  - Poisson arrivals (M),
  - general service times (G) and
  - homogeneous **FCFS** servers
- Mean additional cost to dispatch the arriving customer with service time Δ to queue k:

$$v_{\mathbf{z}_k \oplus \Delta} - v_{\mathbf{z}_k} = u_{\mathbf{z}_k} + \Delta + \frac{\lambda_k \Delta (2u_{\mathbf{z}_k} + \Delta)}{2(1-\rho_k)}$$

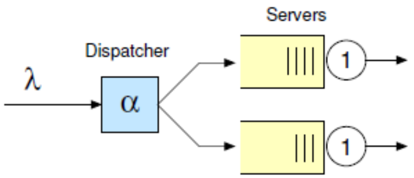
- Optimal RND policy dispatches with equal probabilities. Thus, $\lambda_k \equiv \lambda$, $\rho_k \equiv \rho$, and FPI-RND = LWL

Numerical results (Exp service times)

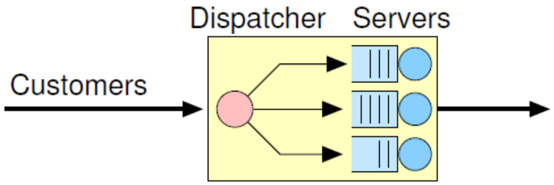


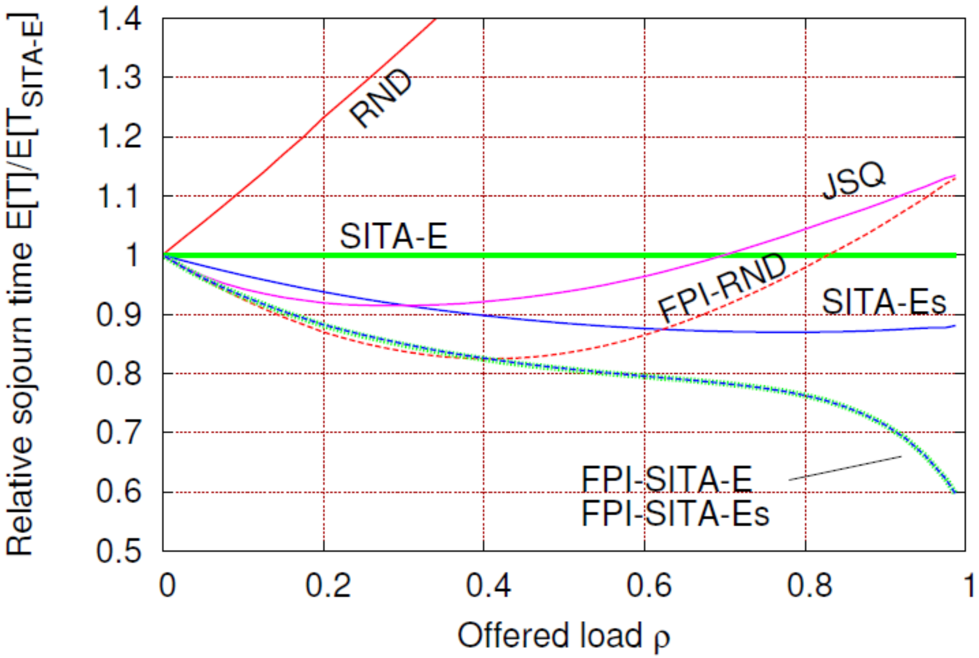


- Two identical FCFS servers
- $X \sim \text{Exp}(1)$

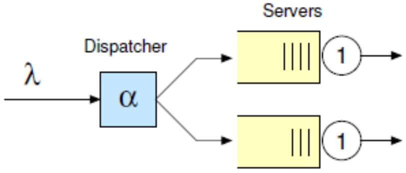


Numerical results (Pareto service times)

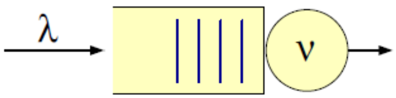




- Two identical FCFS servers
- $X \sim \text{Pareto}(1)$



Size-aware M/D/1-PS queue

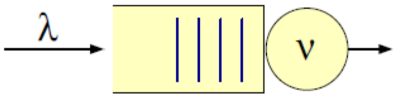


- Poisson arrivals with rate λ
- Deterministic service times s
- Single server with load $\rho = \lambda s$
- PS scheduling discipline
- State description:

$$\mathbf{z} = (\Delta_1, \ldots, \Delta_n)$$

  - Δ_i = (remaining) service time of customer i
  - n = next leaving customer (i.e., the ”oldest” one)

Size-aware relative value for M/D/1-PS



- Proposition [Hyytiä et al. (2011a)]:

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

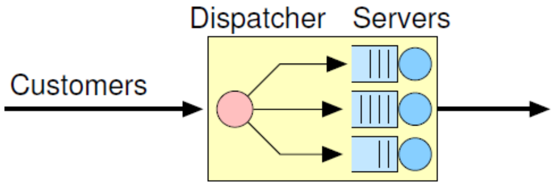
  – where

$$\mathbf{z} = (\Delta_1, \ldots, \Delta_n), \quad u_{\mathbf{z}} = \sum_{i=1}^{n} \Delta_i, \quad \rho = \lambda s < 1$$

  – $u_{\mathbf{z}}$ = initial workload

- Note: $[v_{\mathbf{z}} - v_0]_{\mathrm{PS}} = 2 * [v_{\mathbf{z}} - v_0]_{\mathrm{FCFS}} - u_{\mathbf{z}}$

Size-aware MDP approach



- Example: Dispatching problem with
  - Poisson arrivals (M),
  - deterministic service times (D) and
  - homogeneous **PS** servers
- Mean additional cost to dispatch the arriving customer with service time Δ to queue k:

$$v_{\mathbf{z}_k \oplus \Delta} - v_{\mathbf{z}_k} = \frac{2u_{\mathbf{z}_k} + \Delta}{1 - \rho_k}$$

- Optimal RND policy dispatches with equal probabilities. Thus, $\rho_k \equiv \rho$ and FPI-RND = LWL = OPT!

Generalizations

- Hyytiä et al. (2012a): Size-aware relative values for
  - M/G/1-FCFS, M/G/1-LCFS, M/G/1-SRPT, M/G/1-SPT
- Hyytiä et al. (2011a): Size-aware relative values for
  - M/D/1-PS
- Hyytiä et al. (2011b): Size-aware relative values for
  - M/M/1-PS
- Penttinen et al. (2011): Size and energy-aware relative values for
  - M/G/1-FCFS, M/G/1-LCFS
- Hyytiä et al. (2012b): Size-aware relative values for
  - M/G/1-SPTP (when minimizing the mean slowdown)
- Hyytiä et al. (2012c): Size-aware relative values for
  - M/G/1-FCFS, M/G/1-LCFS with general holding costs
- Hyytiä & Aalto (2013): Size-aware relative values for
  - M^X/G/1-FCFS with batch arrivals

Own references

- Hyytiä, Penttinen & Aalto (2012a)
  Size- and state-aware dispatching problem with queue-specific job sizes, *EJOR*
- Hyytiä, Penttinen, Aalto & Virtamo (2011a)
  Dispatching problem with fixed size jobs and processor sharing discipline, in *ITC*
- Hyytiä, Virtamo, Aalto & Penttinen (2011b)
  M/M/1-PS queue and size-aware task assignment, *PEVA*
- Penttinen, Hyytiä & S. Aalto (2011c)
  Energy-aware dispatching in parallel queues with on-off energy consumption, in *IEEE IPCCC*
- Hyytiä, Aalto & Penttinen (2012b)
  Minimizing slowdown in heterogeneous size-aware dispatching systems, in *ACM SIGMETRICS/PERFORMANCE*
- Hyytiä, Aalto, Penttinen & Virtamo (2012c)
  On the value function of the M/G/1 FCFS and LCFS queues, *JAP*
- Hyytiä & Aalto (2013)
  To split or not to split: Selecting the right server with batch arrivals, *ORL*

The End