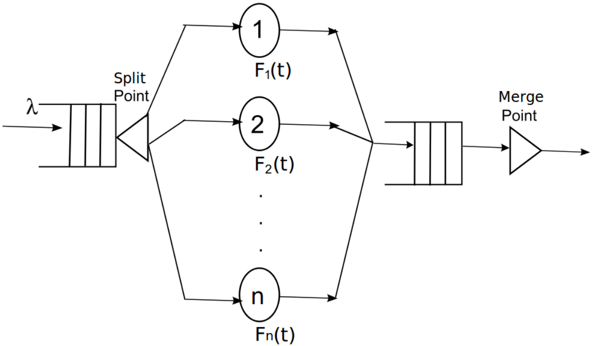
Reduction of Variability in Split–Merge Systems

Iryna Tsimashenka, William Knottenbelt

Imperial College London

September 29, 2011

Introduction



- Proposition is to reduce variability
- Minimise utilisation of physical buffer space

- Vector of Delays $\mathbf{d} = (d_1, d_2, ..., d_i, ..., d_{n-1}, d_n)$

- Vector of Delays $\mathbf{d} = (d_1, d_2, ..., d_i, ..., d_{n-1}, d_n)$
- Cost Function $C(\mathbf{d}) = E(X) - E(Y)$, where X and Y are random variables denoting max and min completion time across all items in an order

$$F_X(t) \sim \prod_{i=1}^{n} F_i(t - d_i) \tag{1}$$

$$F_X(t) \sim \prod_{i=1}^{n} F_i(t - d_i) \tag{1}$$

$$F_Y(t) \sim (1 - \prod_{i=1}^{n}(1 - F_i(t - d_i))) \tag{2}$$

$$F_X(t) \sim \prod_{i=1}^{n} F_i(t - d_i) \tag{1}$$

$$F_Y(t) \sim (1 - \prod_{i=1}^{n}(1 - F_i(t - d_i))) \tag{2}$$

$$E[X] = \int_0^\infty 1 - F_X(t)\,dt \tag{3}$$

$$E[X] = \int_0^\infty 1 - F_X(t)\, dt \tag{3}$$

$$E[Y] = \int_0^\infty 1 - F_Y(t)\, dt \tag{4}$$

$$E[X] = \int_0^\infty 1 - F_X(t)\,dt \tag{3}$$

$$E[Y] = \int_0^\infty 1 - F_Y(t)\,dt \tag{4}$$

$$\widetilde{\mathbf{d}} = (\widetilde{d}_1, \widetilde{d}_2, ..., \widetilde{d}_{i-1}, 0, \widetilde{d}_{i+1}, ..., \widetilde{d}_{n-1}, \widetilde{d}_n) \tag{5}$$

Numerical Results

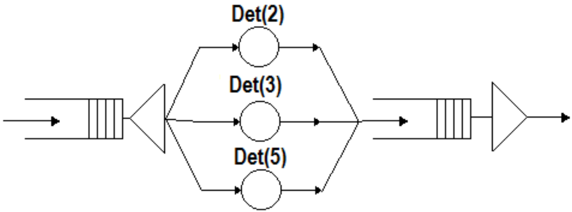
1. Consider a split–merge system with 3 service nodes

Numerical Results

1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Det(2), Det(3) and Det(5)

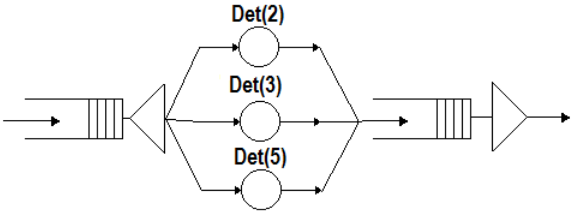
Numerical Results

1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Det(2), Det(3) and Det(5)



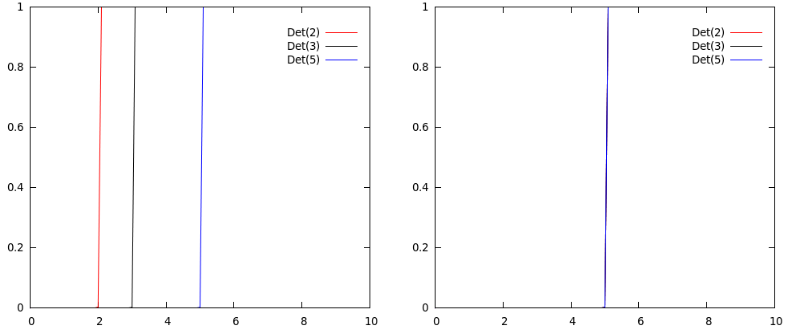
Numerical Results

1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Det(2), Det(3) and Det(5)

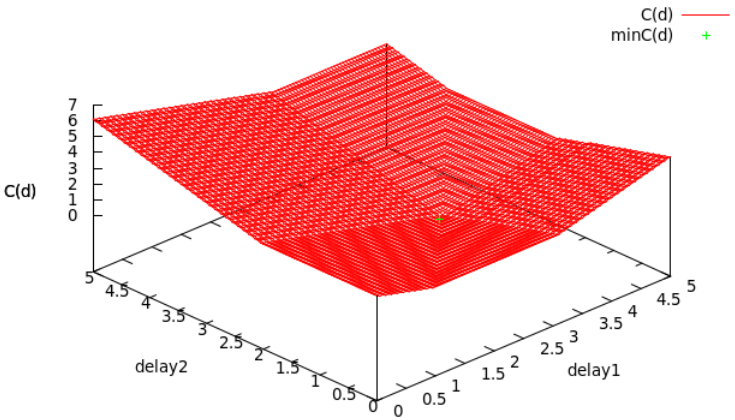


3. Vector of optimal delays $\widetilde{\mathbf{d}} = (3.0, 2.0, 0.0)$

CDFs of server service times before and after adding the optimal delays



Surface plot of cost function against delays



Numerical Results (non-trivial case)

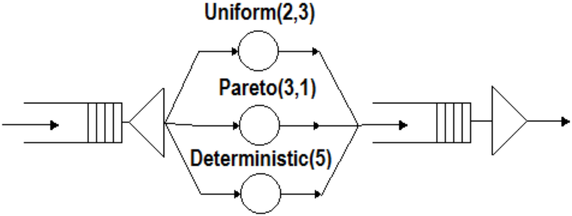
1. Consider a split–merge system with 3 service nodes

Numerical Results (non-trivial case)

1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Uniform(2,3), Pareto(3,1) and Det(5)

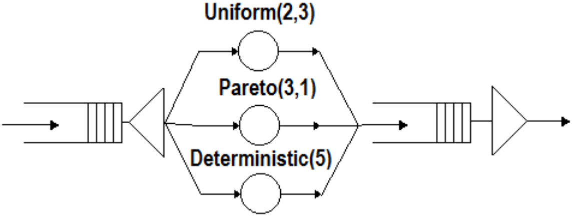
Numerical Results (non-trivial case)

1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Uniform(2,3), Pareto(3,1) and Det(5)



Numerical Results (non-trivial case)

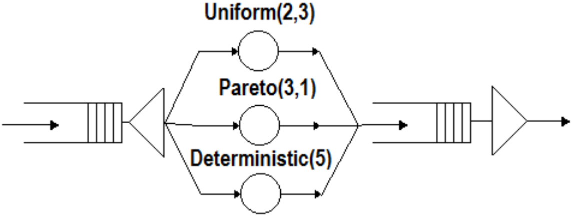
1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Uniform(2,3), Pareto(3,1) and Det(5)



3. Vector of optimal delays $\widetilde{\mathbf{d}} = (2.5317, 3.7154, 0.0)$

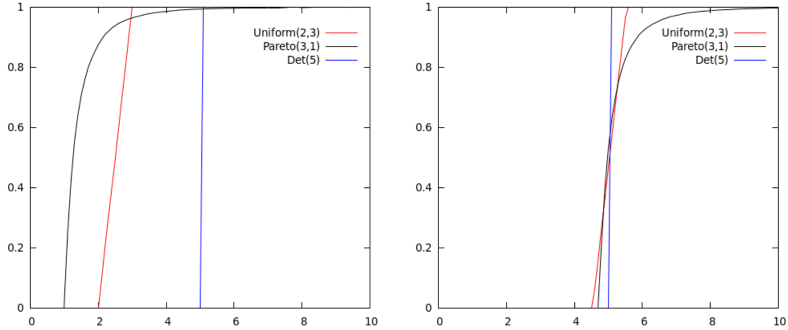
Numerical Results (non-trivial case)

1. Consider a split–merge system with 3 service nodes
2. With service time distribution functions: Uniform(2,3), Pareto(3,1) and Det(5)

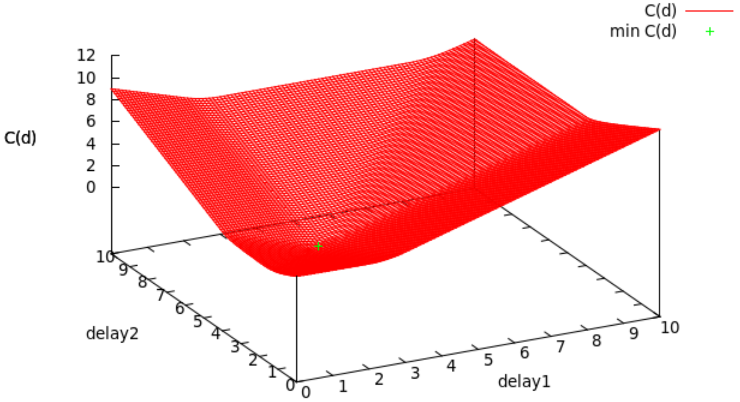


3. Vector of optimal delays $\widetilde{\mathbf{d}} = (2.5317, 3.7154, 0.0)$
4. Intuitively expected delays $\mathbf{d}_{intuit} = (2.5, 3.5, 0.0)$, $\widetilde{\mathbf{d}} \neq \mathbf{d}_{intuit}$

CDFs of server service times before and after adding the optimal delays



Surface plot of cost function against delays



Future Work

1. Parallel generation of cost function landscape

Future Work

1. Parallel generation of cost function landscape
2. Find appropriate algorithm for determining the vector of optimal delays

Future Work

1. Parallel generation of cost function landscape
2. Find appropriate algorithm for determining the vector of optimal delays
3. Investigate analogous optimisation of a fork-join system.

Thank you!