Towards the Tradeoff Between Service Performance and Information Freshness

Zhongdong Liu Bo Ji

Center for Networked Computing
Department of Computer and Information Sciences
Temple University

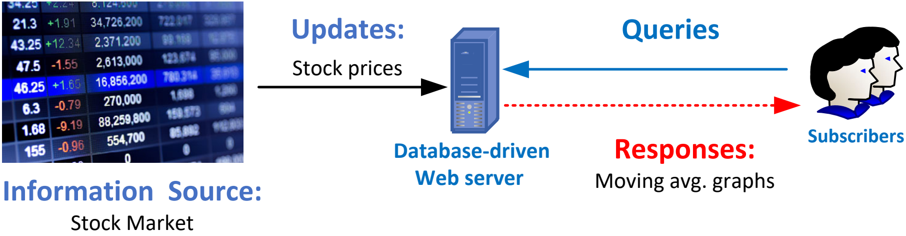
May 22, 2019

IEEE ICC 2019
Shanghai, China

Motivation

Processing updates ← The computing resources → Processing queries

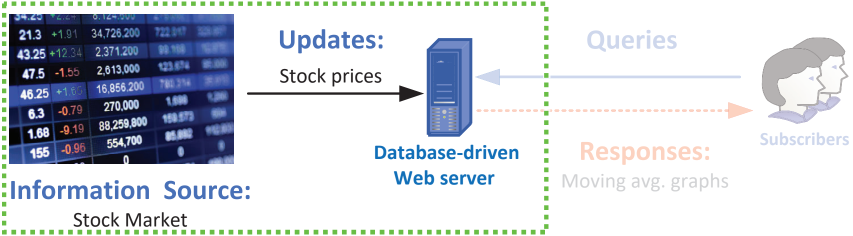
Example:



Motivation

Processing updates $\leftarrow$ The computing resources

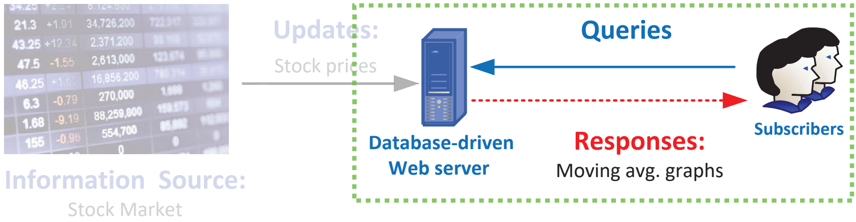
Example:



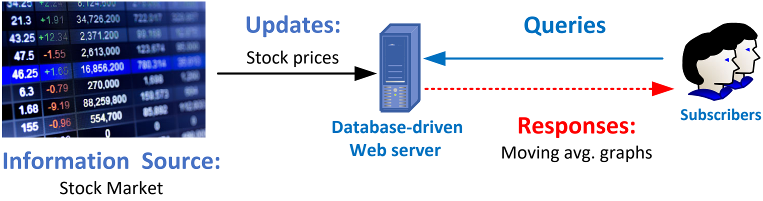
Motivation

The computing resources → Processing queries

Example:

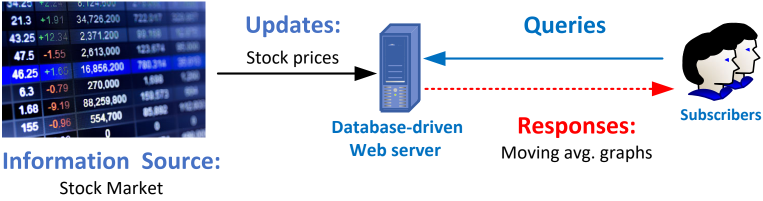


Key Tradeoff



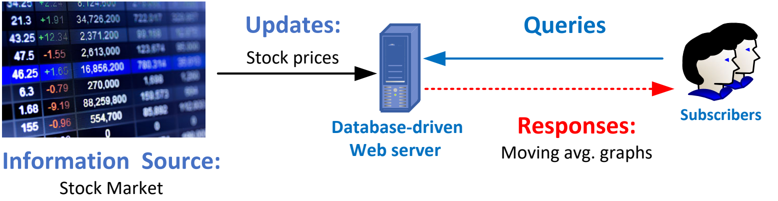
- The computing resources are **shared**!

Key Tradeoff



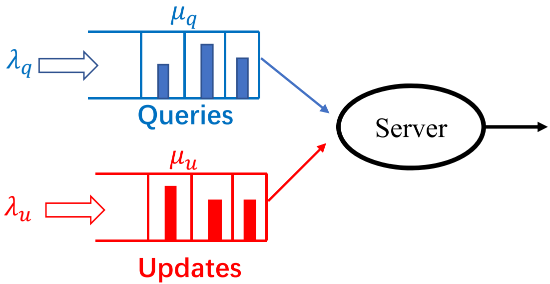
- The computing resources are **shared**!
- How to schedule the updates and queries jointly?

Key Tradeoff



- The computing resources are **shared**!
- How to schedule the updates and queries jointly?
- Tradeoff: **Service performance** vs. **Information freshness**
  - Serve updates first: **fresh** information, **long** response time
  - Serve queries first: **short** response time, **stale** information

Proposed System Model



- Two M/M/1 queues share one single server

Metric of service performance

- The response time of queries

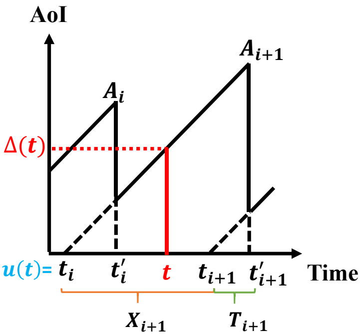
Metric of information freshness

- Age-of-Information (AoI) of updates: The time elapsed since the generation of the latest update:

$$\Delta(t) := t - u(t)$$

- Peak-Age-of-Information (PAoI) of updates:

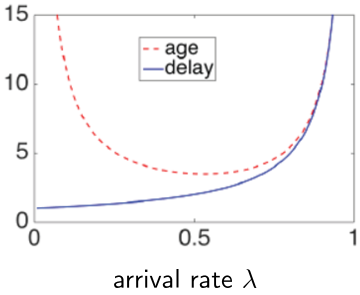
$$A_{i+1} = X_{i+1} + T_{i+1}$$



AoI vs. Throughput & Delay

M/M/1 FCFS queue: arrival rate λ, service rate $\mu = 1$

- Large arrival rate λ:
  - high throughput; large queueing delay; large AoI
- Small arrival rate λ:
  - low delay; large interarrival time; large AoI
- AoI depends on both:
  - queueing delay
  - inter-arrival time



(Image source: http://www.auburn.edu/∼yzs0078)

Related Work

Response time studies:

- Performance vs. Data freshness [Labrinidis et al. '04]
- Performance vs. Robustness [Osogami et al. '05]

AoI studies:

- AoI in M/M/1 under the FCFS policy [Kaul et al. '12]
- Poisson arrivals & single server [Costa et al. '14; Yates et al. '12]
- Pull model from user side [Sang et al. '17]

None of those work analyzes the tradeoff between the service performance and information freshness in a rigorous manner!

Related Work

Response time studies:

- Performance vs. Data freshness [Labrinidis et al. '04]
- Performance vs. Robustness [Osogami et al. '05]

AoI studies:

- AoI in M/M/1 under the FCFS policy [Kaul et al. '12]
- Poisson arrivals & single server [Costa et al. '14; Yates et al. '12]
- Pull model from user side [Sang et al. '17]

None of those work analyzes the tradeoff between the service performance and information freshness in a rigorous manner!

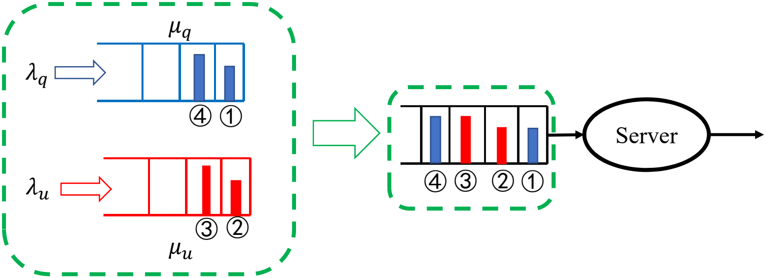
Our Contributions

- A single-server two-queue model for studying the key tradeoff
- Threshold-based scheduling policies that achieve better tradeoff

A Simple Policy: FCFS

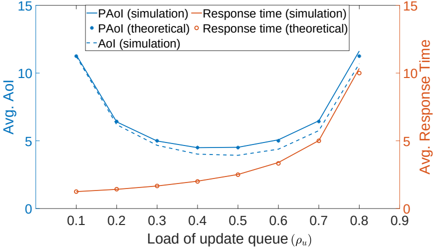
First-Come-First-Served (FCFS) policy

- Serving updates and queries according to the order of their arrivals

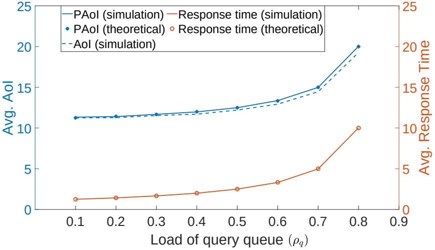


- Theoretical analysis for FCFS policy

Numerical Result: FCFS



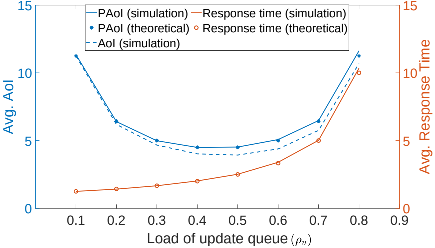
Fixed $\lambda_q = 0.1,\ \mu_q = \mu_u = 1,$ and $\rho_u = \lambda_u/\mu_u$



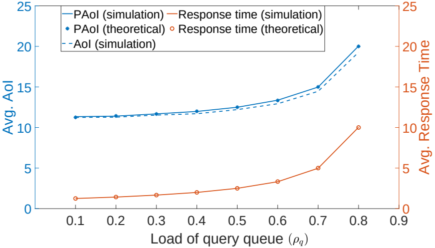
Fixed $\lambda_u = 0.1,\ \mu_q = \mu_u = 1,$ and $\rho_q = \lambda_q/\mu_q$

Both average PAoI and response time are **large** when the load is high!

Numerical Result: FCFS



Fixed $\lambda_q = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_u = \lambda_u/\mu_u$



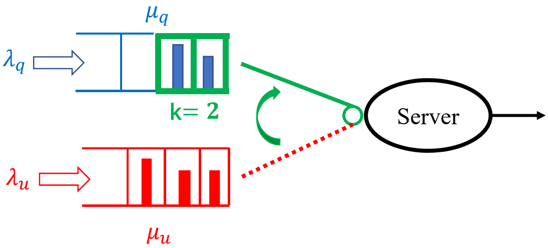
Fixed $\lambda_u = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_q = \lambda_q/\mu_q$

Both average PAoI and response time are **large** when the load is high!

The FCFS has no ability in controlling the tradeoff!

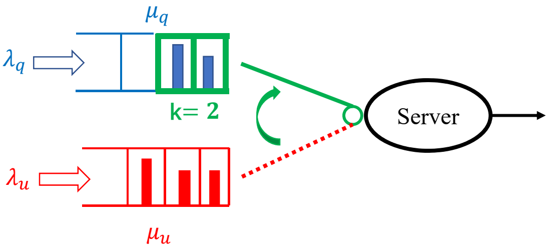
The Query-k Policy

- One single threshold k for the query queue
- Server switches condition:
  - The number of queries reaches the threshold k, or
  - The update queue becomes empty



The Query-k Policy

- One single threshold k for the query queue
- Server switches condition:
  - The number of queries reaches the threshold k, or
  - The update queue becomes empty



- Two special cases:
  - Threshold $k = 1$: the priority is **always** given to the query queue
  - Threshold $k = \infty$: exhaustive service at both queues

Proposition 2

Under the Query-1 policy, the expected response time is

$$\mathbb{E}\left[T_q\right] = \frac{1}{\mu_q} + \frac{\rho_q/\mu_q}{1-\rho_q},$$

and the expected PAoI is

$$\mathbb{E}\left[A\right] = \mathbb{E}\left[X_u\right] + \mathbb{E}\left[T_u\right] = \frac{1}{\lambda_u} + \frac{1/\mu_u}{1-\rho_q} + \frac{\rho_q/\mu_q + \rho_u/\mu_u}{\left(1-\rho_q\right)\left(1-\rho_q-\rho_u\right)}.$$

Proof sketch:

Equivalent to a preemptive priority queue with two classes of jobs [1].

[1] Harchol-Balter, Mor. Performance modeling and design of computer systems: queueing theory in action. Cambridge University Press, 2013.

Proposition 3

Under the Query-k policy with $1 < k < \infty$, the expected response time is

$$\mathbb{E}[T_q] = \mathbb{E}[N_q]/\lambda_q,$$

and the expected PAoI is

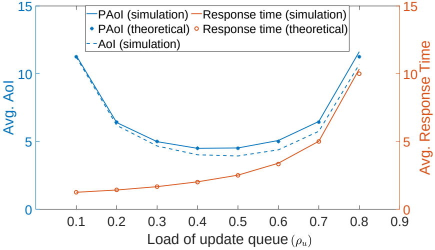
$$\mathbb{E}[A] = \mathbb{E}[X_u] + \mathbb{E}[T_u] = \frac{1}{\lambda_u} + \frac{\mu_{\mathrm{u}}}{\lambda_u} \cdot \left(\frac{\lambda_{\mathrm{q}}/\mu_{\mathrm{q}}^2 + \lambda_{\mathrm{u}}/\mu_{\mathrm{u}}^2}{1-\rho} - \frac{\mathbb{E}[N_q]}{\mu_{\mathrm{q}}} \right).$$

Proof sketch:

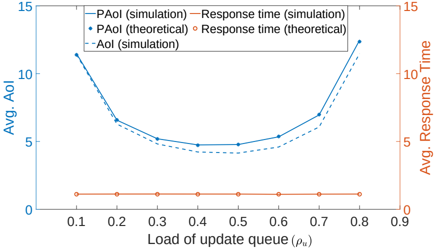
[2] provides a method for calculating $\mathbb{E}[N_q]$, then applying the Little's Law and Conservation Law.

[2] Boxma, Onno J., G. M. Koole, and Isi Mitrani. "A two-queue polling model with a threshold service policy." MASCOTS'95.

Numerical Results: The Query-1 Policy

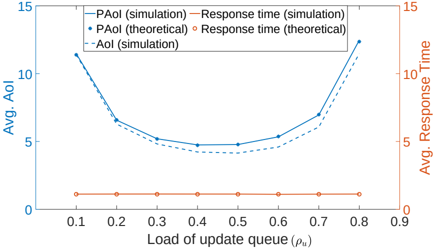


FCFS
(Fixed $\lambda_q = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_u = \lambda_u/\mu_u$)

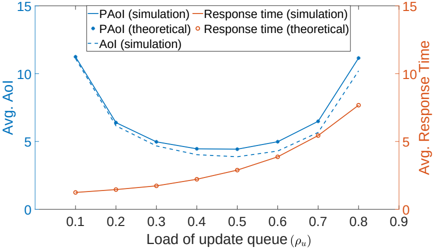


The Query-1
(Fixed $\lambda_q = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_u = \lambda_u/\mu_u$)

Query-1 policy achieves **much better response time** than FCFS!



The Query-1
(Fixed $\lambda_q = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_u = \lambda_u/\mu_u$)

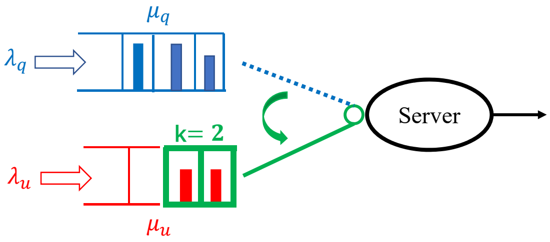


The Query-3
(Fixed $\lambda_q = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_u = \lambda_u/\mu_u$)

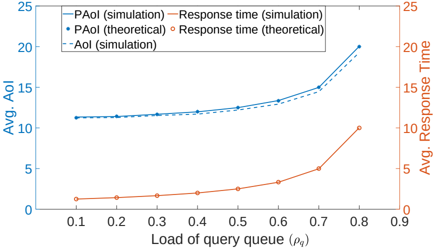
Query-3 has small improvement on the PAoI but **large response time**!

The Update-k Policy

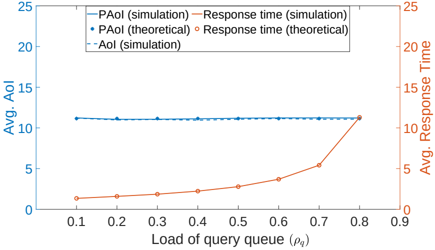
- Similar to the Query-k policy, except threshold k for the update queue
- Server switches condition:
  - The number of updates reaches the threshold k, or
  - The query queue becomes empty



Numerical Result: The Update-1 Policy



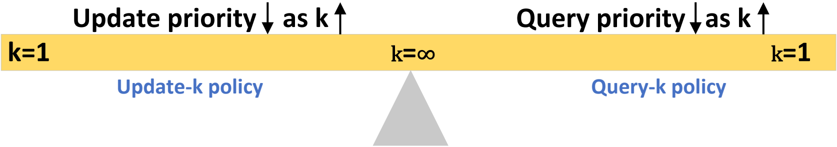
FCFS
(Fixed $\lambda_u = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_q = \lambda_q/\mu_q$)



The Update-1
(Fixed $\lambda_u = 0.1,\ \mu_q = \mu_u = 1,$
and $\rho_q = \lambda_q/\mu_q$)

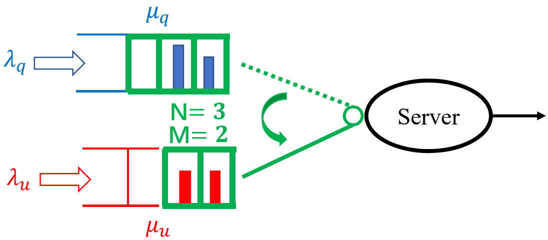
Update-1 achieves **much better PAoI** than FCFS!

Limitation of The Single-threshold-based Policies

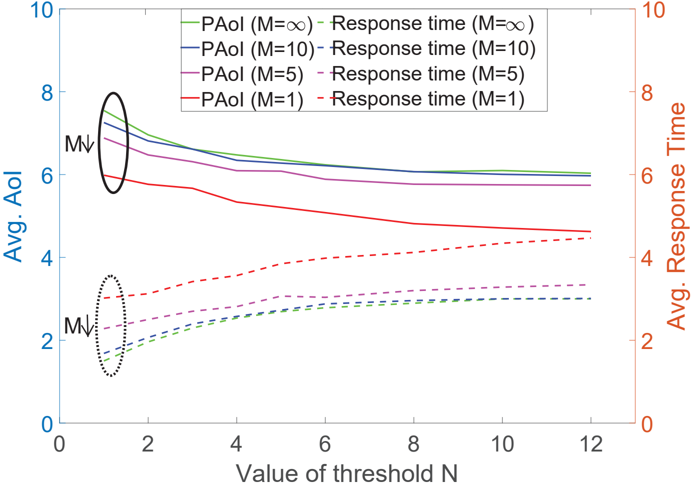


The priority is given to one queue only!

- Threshold M for the update queue; threshold N for the query queue
- Server swithes conditions
  - The threshold is reached, or
  - The other queue is empty



Simulation Result: The Joint-(M, N) Policy



Conclusion & Future Work

Conclusion:

- Proposed a simple single-server two-queue model
  - Tradeoff: **Service performance** vs. **Information freshness**
- Proposed threshold-based scheduling policies
  - The Query-k, the Update-k and the Joint-(M, N) policy
  - Analyze the response time and the PAoI rigorously

Future Work:

- Average PAoI vs. Average AoI
- The systematical analysis of the Joint-(M, N) policy
- Switching overhead

Thank You!

Questions?

Zhongdong Liu (zhongdong.liu@temple.edu)

Analysis: FCFS

Proposition 1

Under the FCFS policy, the expected response time is

$$\mathbb{E}\left[T_q\right] = \frac{\rho_u/\mu_u + \left(1-\rho_u\right)/\mu_q}{1-\rho_u-\rho_q},$$

and the expected PAoI is

$$\mathbb{E}\left[A\right] = \mathbb{E}\left[X_u\right] + \mathbb{E}\left[T_u\right] = \frac{1}{\lambda_u} + \frac{\rho_q/\mu_q + \left(1-\rho_q\right)/\mu_u}{1-\rho_u-\rho_q}.$$

Proof sketch:

For an update, its response time = total service time of updates + total service time of queries + its own service time.