FCD: Fast-Concurrent-Distributed Load Balancing under Switching Costs and Imperfect Observations

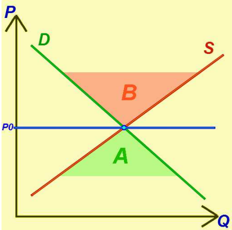
Furong Huang, Animashree Anandkumar

U.C. Irvine

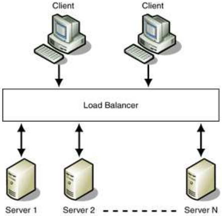
Presenter: Krishna Jagannathan

Load Balancing for Distributed Networks

Motivations



Cabel Supply Market Balance



Network Load Balance

Cloud Computing

Goal: Load Balancing

- A Nash-equilibrium: A state that no user has the incentive to change her current decision.

Convergence in the Networking System ASAP

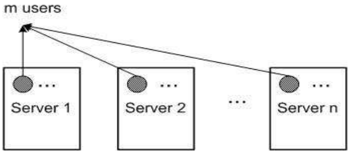
Outline

Outline

System Model

Notation

- Server load $X_i(t) :=$ # of users in server i at time slot t
- Load experience $Y_a(t)$ by user a at t
- ϵ-nash equilibrium: $\max_{ij} |X_i(t) - X_j(t)| \leq \epsilon \frac{m}{n}$



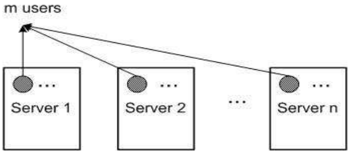
m users(unit bandwidth requirement), n servers,

$m \gg n$

System Model

Notation

- Server load $X_i(t)$:= # of users in server i at time slot t
- Load experience $Y_a(t)$ by user a at t
- ϵ-nash equilibrium: $\max_{ij} |X_i(t) - X_j(t)| \le \epsilon \frac{m}{n}$



m users(unit bandwidth requirement), n servers,

$m \gg n$

What's different from traditional load balancing?

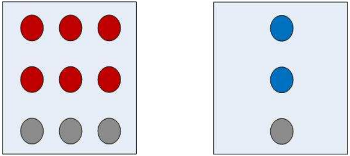
- No Dispatcher: Random local search in Distributed System
- Concurrently Implemented: Entails the design of load estimation mechanisms due to indirect observations

  * "Elementary Step System" vs Concurrent Process
- Switching Penalty:Entails a careful design of exploration probabilities, "trade-off"
- Extension to open systems: User dynamics, a wide class of arrival and departure processes

Outline

A Toy Case with 2 Servers

- Explore with $\gamma_{t_0} = \frac{2}{3}$



Initial State

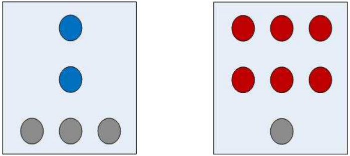
A Toy Case with 2 Servers

- Explore with $\gamma_{t_0} = \frac{2}{3}$
- Red users observe

$$Y_r(t_0) = X_1(t_0) = 9$$
$$Y_r(t_1) = X_2(t_1) = 7$$

Blue users observe

$$Y_b(t_0) = X_2(t_0) = 3$$
$$Y_b(t_1) = X_1(t_1) = 5$$



Exploration

A Toy Case with 2 Servers

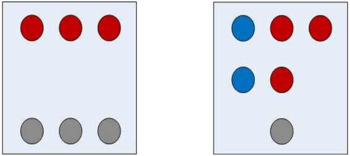
- Explore with $\gamma_{t_0} = \frac{2}{3}$
- Red users observe

$$Y_r(t_0) = X_1(t_0) = 9$$
$$Y_r(t_1) = X_2(t_1) = 7$$

Blue users observe

$$Y_b(t_0) = X_2(t_0) = 3$$
$$Y_b(t_1) = X_1(t_1) = 5$$

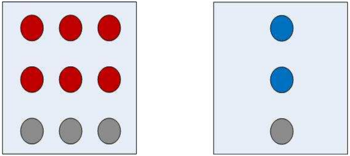
- Red users backtrack with prob. $= \frac{1}{2}$
- Blue users backtrack with prob. $= 1$



Backtracking

Exploration and Backtracking Probability

Exploration Probability γ_t

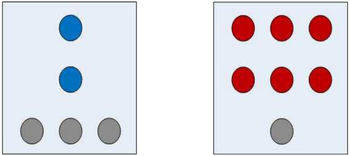


Initial State

Exploration and Backtracking Probability

Exploration Probability γ_t

- Why need a decaying exploration rate? Stable
- Binary search



Exploration

Exploration and Backtracking Probability

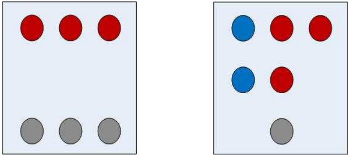
Exploration Probability γ_t

- Why need a decaying exploration rate? Stable
- Binary search

Backtracking Probability

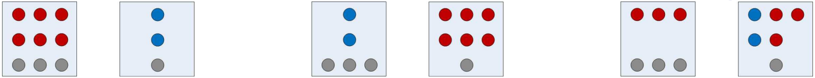
$f_{ji} := \{x_i(t-1), x_j(t), \gamma_t\}$

- Decision made based on observed server loads at $t-1$ and t for each user.



Backtracking

Parameter Estimation-backtracking prob.



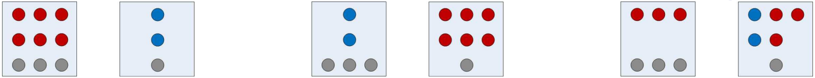
Initial　　Exploration　　Backtracking

The Design of Backtracking Probability

- Case 1: Blue User Switches from Lighter Load to Heavier Load

$$f_b(1) := f(Y_b(1), Y_b(0), \gamma_1) = 1$$

Parameter Estimation-backtracking prob.



Initial Exploration Backtracking

The Design of Backtracking Probability

- Case 1: Blue User Switches from Lighter Load to Heavier Load

$$f_b(1) := f(Y_b(1), Y_b(0), \gamma_1) = 1$$

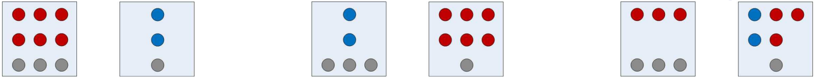
- Case 2: Red User Switches from Heavier Load to Lighter Load

$$f_r(1) := f(Y_r(1), Y_r(0), \gamma_1)$$

Criterion: Single Step Convergence Enforcement Strategy

$$\mathbb{E}[X_1(2)] = \mathbb{E}[X_2(2)]$$

Parameter Estimation-backtracking prob.



Initial Exploration Backtracking

The Design of Backtracking Probability

- Case 1: Blue User Switches from Lighter Load to Heavier Load

$$f_b(1) := f(Y_b(1), Y_b(0), \gamma_1) = 1$$

- Case 2: Red User Switches from Heavier Load to Lighter Load

$$f_r(1) := f(Y_r(1), Y_r(0), \gamma_1)$$

Criterion: Single Step Convergence Enforcement Strategy

$$\mathbb{E}[X_1(2)] = \mathbb{E}[X_2(2)]$$

Extension to n server case

Backtracking Probability be affected by other servers as well:partial information Idea: Mathematical induction + Information estimation

Outline

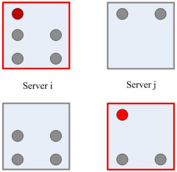
Outline

Closed system convergence time Guarantee

- Convergence time guarantee

$$\boxed{\mathbb{E}[T] \leq \max\left\{n\log n + n^{\frac{1}{\beta}}, \left[\left(\frac{n}{m}\right)^3 \log n\right]^{\frac{1}{\beta}}\right\}}$$

where exploration probability $\gamma_t = t^{-\beta}$, $\beta \in [0.5, 1]$ and $m \gg n$.

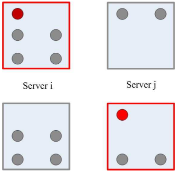


Closed system convergence time Guarantee

- Convergence time guarantee

$$\boxed{\mathbb{E}[T] \le \max\left\{ n \log n + n^{\frac{1}{\beta}}, \left[\left(\frac{n}{m}\right)^3 \log n \right]^{\frac{1}{\beta}} \right\}}$$

where exploration probability $\gamma_t = t^{-\beta}$, $\beta \in [0.5, 1]$ and $m \gg n$.

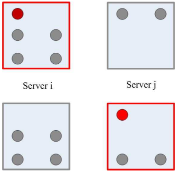


Closed system convergence time Guarantee

- Convergence time guarantee

$$\boxed{\mathbb{E}[T] \leq \max\left\{ n\log n + n^{\frac{1}{\beta}}, \left[\left(\frac{n}{m}\right)^3 \log n\right]^{\frac{1}{\beta}} \right\}}$$

  where exploration probability $\gamma_t = t^{-\beta}$, $\beta \in [0.5, 1]$ and $m \gg n$.

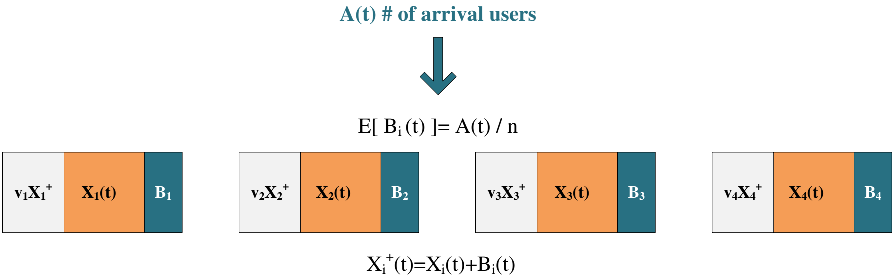


What if new users arrive or current users depart after the closed system starts?

Outline

Outline

Arrival and Departure Processes



Notation

- Users depart rate : $\nu_i(t) \propto X_i^+(t)$
- Users depart number : $\nu_i(t)X_i^+(t)$
- Real time load : $X_i^F(t) := X_i^+(t) - \nu_i(t)X_i^+(t)$

Outline

Convergence Time Guarantees

Open System

- Real time system load constraints

$$\boxed{\Pr\left\{\left|M(t) - \left(1 - \frac{\gamma_t}{n-1}\right) m\right| > n^{-\frac{1}{2}} m^2 \sqrt{\gamma_t}\right\} \leq \frac{\epsilon^2}{4n^2}}$$

where $M(t)$ is the real time system load.

Convergence Time Guarantees

Open System

- Real time system load constraints

$$\boxed{\Pr\left\{\left|M(t) - \left(1 - \frac{\gamma_t}{n-1}\right) m\right| > n^{-\frac{1}{2}} m^2 \sqrt{\gamma_t}\right\} \leq \frac{\epsilon^2}{4n^2}}$$

where $M(t)$ is the real time system load.

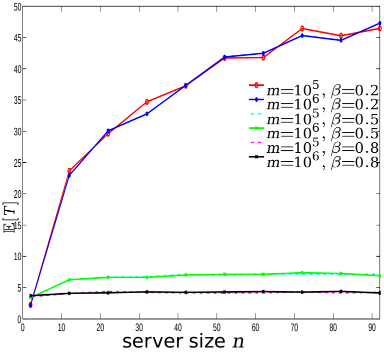
- Convergence time guarantee

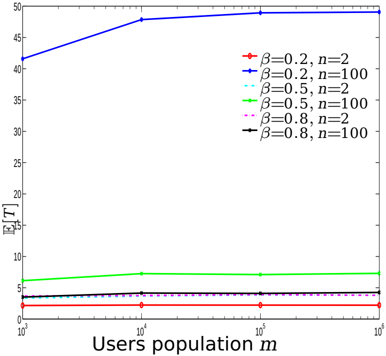
$$\boxed{\mathbb{E}[T] \leq \max\left\{n \log n + n^{\frac{1}{\beta}}, \left[\left(\frac{n}{m}\right)^3 \log n\right]^{\frac{1}{\beta}}, \left(\frac{n^4}{\epsilon m^3} \log n\right)^{\frac{1}{\beta}}\right\}}$$

where $\beta \in [0.5, 1]$.

Outline

$\mathbb{E}[\mathbf{T}]$ under different configurations

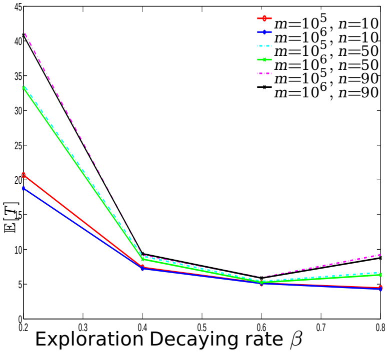




Converge linearly with n

Convergence time is robust with m

$\mathbb{E}[\mathbf{T}]$ under different configurations



Converge slower with smaller β

Outline

Conclusion

Summary

- Proposed fast and concurrent randomized local search algorithm for distributed system
- Convergence time guarantees are analyzed in closed/open systems
- Robustness of the algorithm with dynamic users in open system

furongh@uci.edu