Dynamic Service Placement in Geographically Distributed Clouds

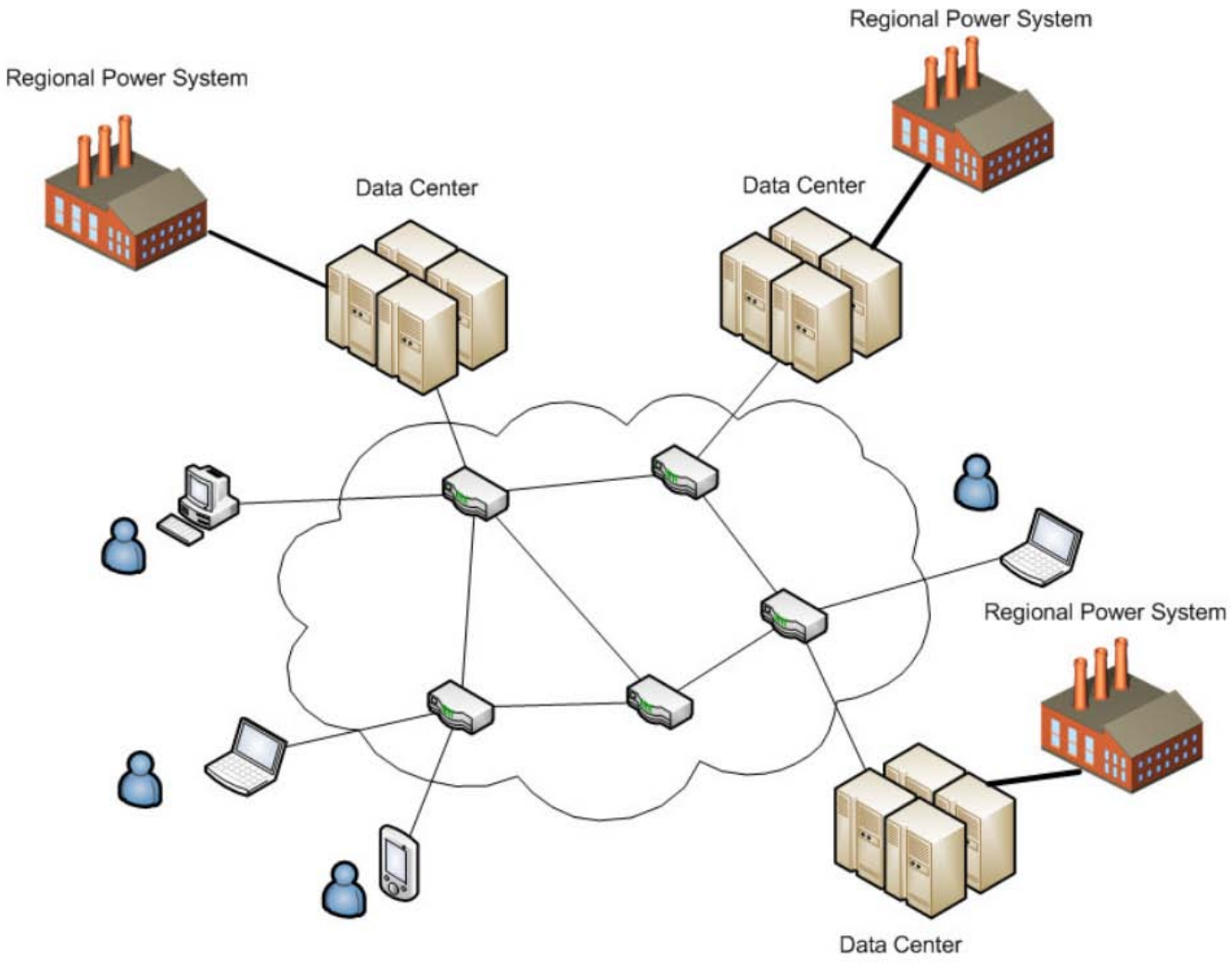
Presenter: Xuanjia Qiu

Nov. 28, 2012

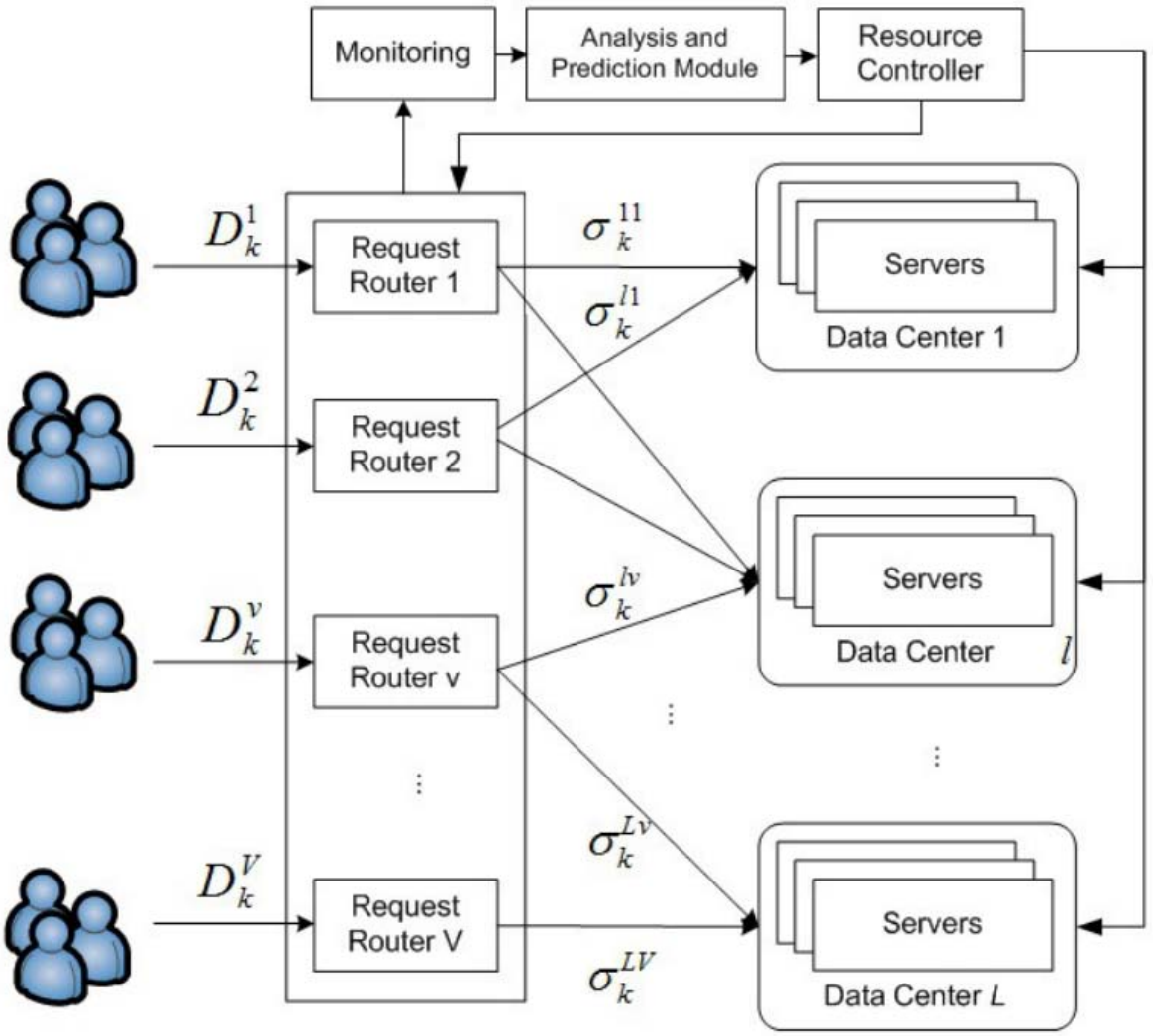
Background

- Authors:
  - Qi Zhang, Mohamed Faten Zhani: University of Waterloo, Canada
  - Quanyan Zhu: UIUC, U.S.
  - Raouf Boutaba: POSTECH, Korea
- Publication:
  - ICDCS, 2012

Problem Background



System Architecture



Model

- Target
  - Minimize operational cost
- Constraints:
  - Average delay of requests are bounded
  - Serving all demand
  - Net exceed data center capacity
- Feature:
  - Demand and cost are dyanmic

Modeling Cost

- Operational cost consists of
  - Resource cost

$$H_k = \sum_{l \in L} x_k^l p_k^l = \sum_{l \in L} \sum_{v \in V} x_k^{lv} p_k^l, \qquad \forall 0 \leq k \leq K$$

  - Reconfiguration cost, modeled as a quadratic penalty function on the change of number of servers

$$G_k = \sum_{l \in L} \sum_{v \in V} S(u_k^{lv}) = \sum_{l \in L} \sum_{v \in V} c^l (u_k^{lv})^2, \forall\ 0 \leq k \leq K.$$

Modeling Delay

- Model demand assigned to a data center as M/M/1 queue
- Average queueing delay

$$q(x_k^{lv}, \sigma_k^{lv}) = \frac{1}{\mu - \lambda} = \frac{1}{\mu - \frac{\sigma_k^{lv}}{x_k^{lv}}}.$$

- Bound the average delay

$$d_{lv} + q(x_k^{lv}, \sigma_k^{lv}) \leq \bar{d}_{lv}, \quad \forall v \in V, l \in L, 0 \leq k \leq K.$$

- To simplify symbols, define

$$a^{lv} = \begin{cases} \frac{1}{\mu - (\bar{d}_{lv} - d_{lv})^{-1}}, & \text{if } \bar{d}_{lv} - d_{lv} > 0, \\ \infty, & \text{otherwise}, \end{cases}$$

Problem Formulation

$$
\begin{aligned}
\min \quad & \sum_{k=0}^{K} \sum_{v \in V} \sum_{l \in L} x_k^{lv} p_k^{lv} + c^l (u_k^{lv})^2 \\
\text{s.t.} \quad & \sum_{l \in L} \frac{x_k^{lv}}{a^{lv}} \geq D_k^v, && \forall v \in V, 0 \leq k \leq K \\
& x_{k+1}^{lv} = x_k^{lv} + u_k^{lv}, && \forall l \in L, v \in V, 0 \leq k < K \\
& \sum_{v \in V} x_k^{lv} \leq C^l, && \forall l \in L, k \in K, \\
& x_k^{lv} \in \mathbb{R}_+, u_k^{lv} \in \mathbb{R}, && \forall l \in L, v \in V, 0 \leq k \leq K
\end{aligned}
$$

- General MIP is NP-hard. Considering the data centers are large, approximating the problem to be LP. Rounding the solution to near integers.

MPC Algorithm for DSPP for One SP

- MPC (Model Predictive Control)

1: Provide initial state $\mathbf{x}_0$, $k \leftarrow 0$
2: **loop**
3: At beginning of control period k:
4: Predict $\mathbf{D}^l_{k+i|k}$ for horizons $i = 1, \cdots, K$ using a demand prediction model
5: Solve DSPP to obtain $\mathbf{u}_{k+t|k}$ for $t = 0, \cdots, W-1$
6: Change the resource allocation according to $\mathbf{u}_{k|k}$
7: Update demand assignment policy of request routers according to equation (13)
8: $k \leftarrow k+1$
9: **end loop**

  - Key points:
    - Run it in each time slot
    - Prediction horizon
    - Use the solution of the current time slot only

Competition among Multiple SPs

- The placement configuration of each SP is kept private from other SPs
- Model as a multi-person non-cooperative game
- One key challenge is the modeling of the data center capacity constraints
  - How to guarantee the total capacity constraint of each data center is not violated, when multiple SPs make decisions independently?

- Approximation
  - Optimally scheduling VMs with heterogeneous resource requirements is NP-hard bin-packing problem
  - Cloud Providers generally design VM sizes to match physical machine capacities, e.g., VM sizes are multiples of each other.
  - Bin-packing can be solved optimally using First-Fit-Decrease policy

$$J^i(\mathbf{u}^i, \mathbf{u}^{-i}) = \sum_{k=0}^{K} \sum_{v \in V} \mathbf{p}_k \mathbf{x}_k^{iv} + \mathbf{u}_k^{iv\top} \mathbf{R}^i \mathbf{u}_k^{iv}$$

$$\begin{aligned}
\text{s.t.} \quad & \mathbf{a}_k^{i\top} \mathbf{x}_k^{iv} \geq D_k^{iv}, && \forall i \in \mathcal{N}, v \in V, 0 \leq k \leq K, \\
& \sum_{i \in \mathcal{N}} \sum_{v \in V} \mathbf{s}^i \mathbf{x}_k^{iv} \leq \mathbf{C}, && 0 \leq k \leq K, \\
& \mathbf{x}_{k+1}^{iv} = \mathbf{x}_k^{iv} + \mathbf{u}_k^{iv}, && \forall i \in \mathcal{N}, v \in V, 0 \leq k \leq K-1, \\
& \mathbf{x}_k^{iv} \in \mathbb{R}_+^L \mathbf{u}_k^{iv} \in \mathbb{R}^L, && \forall i \in \mathcal{N}, v \in V, 0 \leq k \leq K.
\end{aligned}$$

Game analysis

- Generally the Nash equilibrium refers to the stable outcome of the competition, where no SP can improve its cost by unilaterally changing its server allocation over time
- As the controller relies on MPC framework for dynamic resource allocation, a new version of Nash equilibrium for control strategies using MPC framework needs to be defined

Definition of Nash equilibrium

Definition 1 (η-Nash Equilibrium [29]). *Let $\mathcal{I}_k^i$ be the information set of a SP i at time k under a given information structure η^i, and Γ^i is the set of all admissible policies of SP i under η^i. The policy $\{\gamma^{i*}, i \in \mathcal{N}\}$ is an $\eta-$Nash equilibrium of the game Ξ, where $\mathbf{u}^i = \gamma^{i*}(\mathcal{I}_k^i)$ and $\eta = \{\eta^i, i \in \mathcal{N}\}$ if $J^i(\gamma^{i*}, \gamma^{-i*}) \leq J^i(\gamma^i, \gamma^{-i*})$, for all admissible policies $\gamma^i \in \Gamma^i$ and for all $i \in \mathcal{N}$, where $\gamma^{-i*} = \{\gamma^j, j \neq i, j \in \mathcal{N}\}$.*

- Note:
  - Policy is a function. Information set is the parameter for the function of policy.
  - Decision is the value of the function of the optimal policy at information set.

Definition of Nash equilibrium (cont.)

Definition 2 (**W**−MPC Nash Equilibrium)**.** *Let W^i be the prediction window of SP i and every SP adopts MPC as outlined in Algorithm 1. The dynamic non-cooperative game Ξ admits* **W**−*MPC Nash Equilibrium,* $\mathbf{W} = \{W^i, i \in \mathcal{N}\}$, *if the sequences* $\mathbf{u}^{iv*} := \{\mathbf{u}_k^{iv*}, 0 \leq k \leq K\}$ *obtained under MPC algorithms satisfy* $J^i(\mathbf{u}^{i*}, \mathbf{u}^{-i*}) \leq J^i(\mathbf{u}^i, \mathbf{u}^{-i*})$, *for all admissible sequences* $\mathbf{u}^i \in \mathcal{U}^i$ *and for all* $i \in \mathcal{N}$, *where* $\mathcal{U}^i$ *is the set of admissible control sequences under MPC algorithms, and* $\mathbf{u}^{-i*} = \{\mathbf{u}^j, j \neq i, j \in \mathcal{N}\}$.

Social Welfare Problem

$$
\begin{aligned}
\min_{\{\mathbf{u}^1,\ldots,\mathbf{u}^N\}} \quad & \sum_{i\in\mathcal{N}} J^i(\mathbf{u}^1,\ldots,\mathbf{u}^N) \\
\text{s.t.} \quad & \mathbf{a}_k^{i\top}\mathbf{x}_k^{iv} \geq D_k^{iv}, && \forall i\in\mathcal{N}, v\in V, 0\leq k\leq K \\
& \sum_{i\in\mathcal{N}}\sum_{v\in V} \mathbf{s}^i\mathbf{x}_k^{iv} \leq \mathbf{C}, && 0\leq k\leq K, \\
& \mathbf{x}_{k+1}^{iv} = \mathbf{x}_k^{iv} + \mathbf{u}_k^{iv}, && \forall i\in\mathcal{N}, v\in V, 0\leq k< K, \\
& \mathbf{x}_k^{iv}\in\mathbb{R}_+^L \mathbf{u}_k^{iv}\in\mathbb{R}^L, && \forall i\in\mathcal{N}, v\in V, 0\leq k\leq K.
\end{aligned}
$$

(SWP)

- Price of Anarchy (PoA) and Price of Stablity (PoS)

$$\rho_{MPC} := \inf_{\mathbf{u}^* \in \mathcal{U}^*} \frac{\sum_{i \in \mathcal{N}} \sum_{v \in V} J_v^i(\mathbf{u}^{i\circ})}{\sum_{i \in \mathcal{N}} \sum_{v \in V} J_v^i(\mathbf{u}^{i*})},$$

$$\xi_{MPC} = \sup_{\mathbf{u}^* \in \mathcal{U}^*} \frac{\sum_{i \in \mathcal{N}} \sum_{v \in V} J_v^i(\mathbf{u}^{i\circ})}{\sum_{i \in \mathcal{N}} \sum_{v \in V} J_v^i(\mathbf{u}^{i*})},$$

Theorem 1. *Assume that the prediction horizon of each SP $i, i \in \mathcal{N}$, is the same, i.e., $W^i = \bar{W}$ and $\bar{W}$ is also the prediction window used for (SWP). Then, the price of stability ξ_{MPC} of the game Ξ is always equal to 1, i.e., there exists a Nash equilibrium solution yields no efficiency loss under the common knowledge of the capacity constraint.*

- In some equilibriums, there exists loss of efficiency due to selfish of the SPs. But the social optimality is achievable.

- An algorithm that converges to the equilibrium solution whose efficiency is optimal:
  - Based on dual decomposition technique
  - Key points (detailed in next slide):
    - Cloud infrastructure providers is responsible for coordinating the allocation of resources when demand exceeds capacity
    - Iterative process

Iterative Algorithm for computing the Best Nash Equilibrium

Provide initial state $\mathbf{x}_0$, $k \leftarrow 0$, Initialize $\mathbf{C}^i \in \mathbb{R}_+^L$, $\bar{\mathbf{x}}_k^i \leftarrow 0$, $\forall k \in \mathcal{K}$ $\bar{J}(\mathbf{u}^1, ..., \mathbf{u}^N) \leftarrow \infty$, $converged \leftarrow$ **false**

repeat

  for $i = 1 \rightarrow N$ **do**

    $\mathbf{u}^i \leftarrow$ solution of DSPP^i with capacity vector $\mathbf{C}_i, \forall k \in \mathcal{K}$

    $\lambda^{il} \leftarrow$ the dual variable of capacity constraint for DSPP^i of DC $l \in L$, $\forall k \in \mathcal{K}$

  end for

  $\bar{\mathbf{C}}^i := (\mathbf{C^i} + \alpha[\lambda^{i1}, ..., \lambda^{iL}]^\top)$

  $\mathbf{C}^i := \bar{\mathbf{C}}^i \cdot \frac{\mathbf{C}}{\sum_{i \in \mathcal{N}} \bar{\mathbf{C}}^i}$

  $J(\mathbf{u}^1, ..., \mathbf{u}^N) = \sum_{i \in \mathcal{N}} J^i(\mathbf{u}^1, ..., \mathbf{u}^N)$

  if $|J(\mathbf{u}^1, ..., \mathbf{u}^N) - \bar{J}(\mathbf{u}^1, ..., \mathbf{u}^N)| \leq \epsilon \bar{J}(\mathbf{u}^1, ..., \mathbf{u}^N)$ **then**

    $converged \leftarrow$ **true**

  end if

  $\bar{J}(\mathbf{u}^1, ..., \mathbf{u}^N) \leftarrow J(\mathbf{u}^1, ..., \mathbf{u}^N)$

until $converged =$ **true**

Impact of Electricity prices

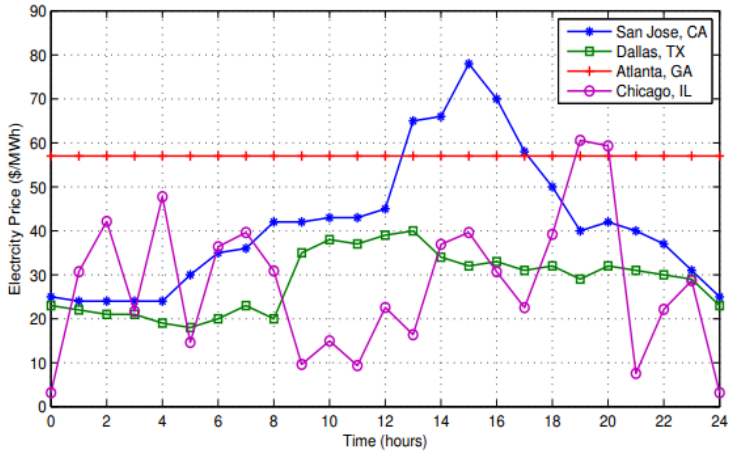


Figure 3. Prices of electricity used in the experiments

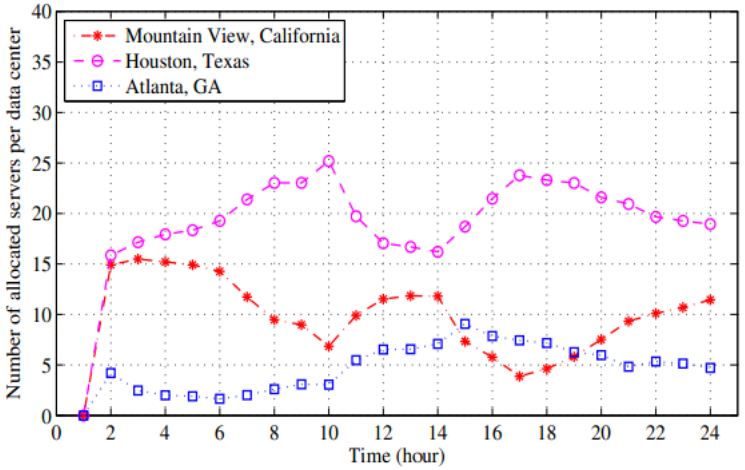


Figure 5. Impact of price on resource allocation

Convergence rate

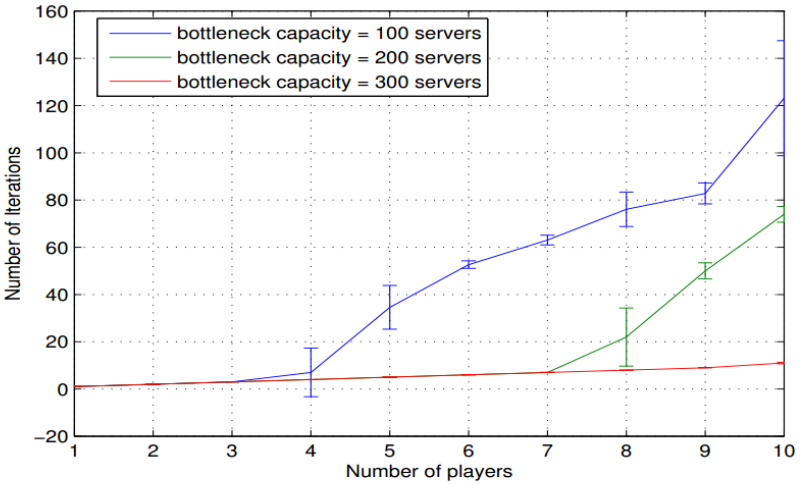


Figure 7. Impact of number of players on the convergence rate

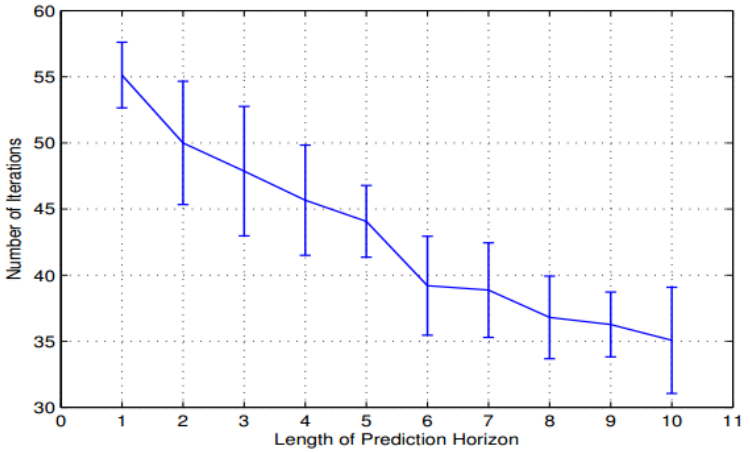


Figure 8. Impact of prediction horizon length the speed of convergence

Prediction Horizon Length

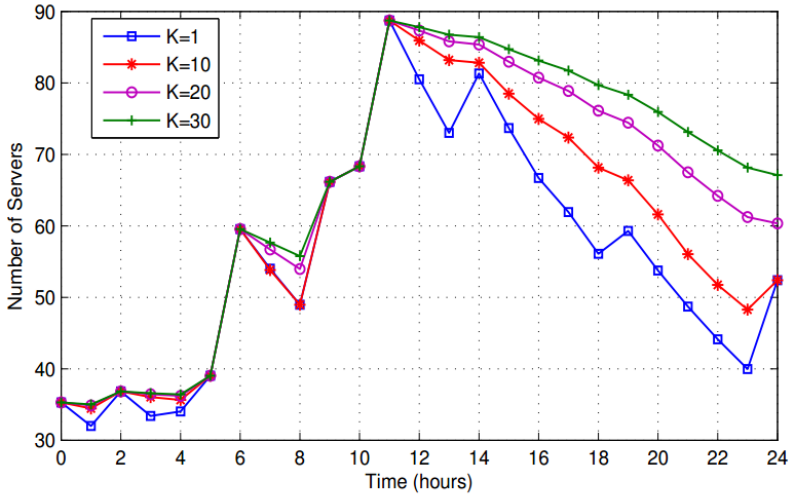


Figure 6. Effect of prediction horizon on the number of servers

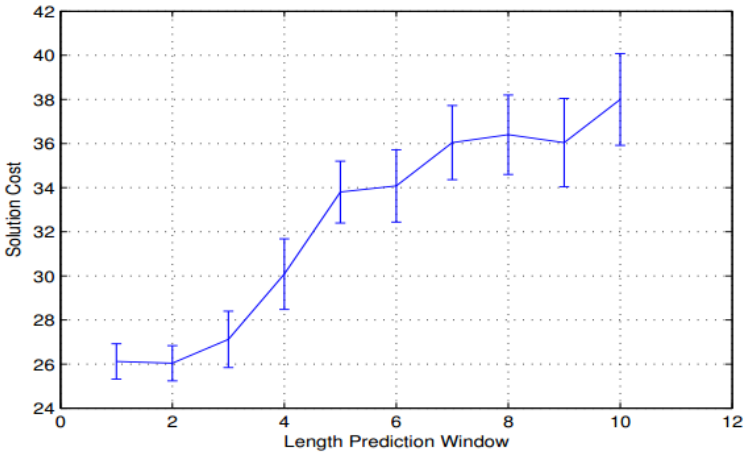
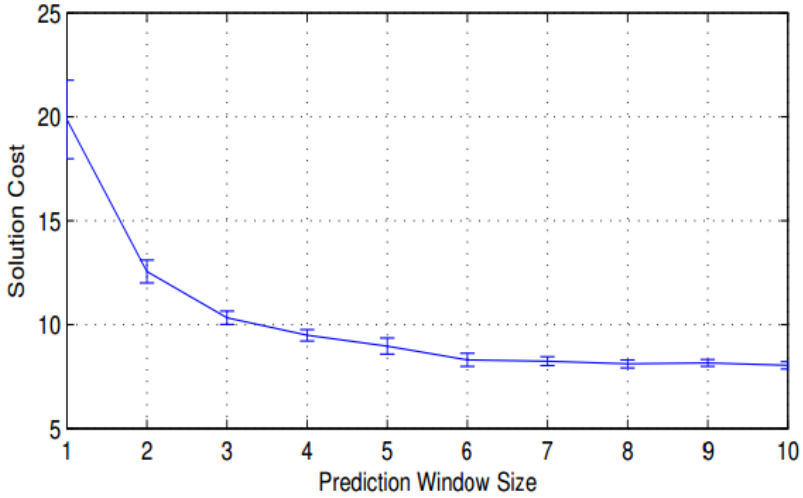


Figure 9. Impact of prediction horizon length the cost



What is Learned from this Paper

- Flow:
  - Model a service placement (or resource allocation) problem as a bipartite graph
  - Model as an linear programming problem
  - Propose an algorithm based on MPC framework
  - Extend it to a multi-person non-cooperative game
    - Define the equilibrium
    - Prove the price of stability is 1
    - Design an algorithm that converges to the equilibrium that is socially optimal

What is Learned from the paper (cont.)

- Ways to model delays
- Reconfiguration cost as a type of cost for preemption of jobs
- Algorithms to solve a DSPP
  - Lyapunov optimization
  - MPC algorithm
  - Approximation algorithm