UCC 2018
Dec 18, 2018

# Hierarchical and Frequency-Aware Model Predictive Control for Bare-Metal Cloud Applications

Yukio Ogawa
Muroran Institute of Technology,
Hokkaido, Japan

Go Hasegawa and Masayuki Murata
Osaka University,
Osaka, Japan



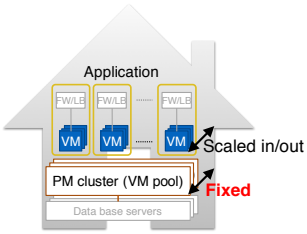
## Contents





## Background and Goal

- On-premises business applications use over-provisioned dedicated physical machines (PMs), which can be improved by migrating them to a bare-metal cloud.

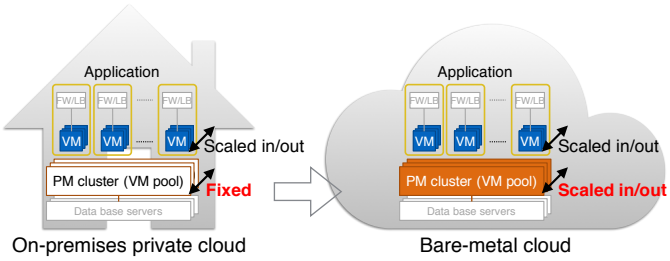



LB: load balancer, FW: fire wall, VM: virtual machine, PM: physical machine



## Background and Goal

- On-premises business applications use over-provisioned dedicated physical machines (PMs), which can be improved by migrating them to a bare-metal cloud.
- Goal: Development of a scaling mechanism for both PMs and virtual machines (VMs) in a bear-metal cloud

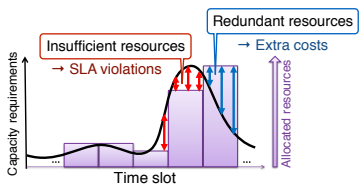



LB: load balancer, FW: fire wall, VM: virtual machine, PM: physical machine



## Resource Scaling Approach

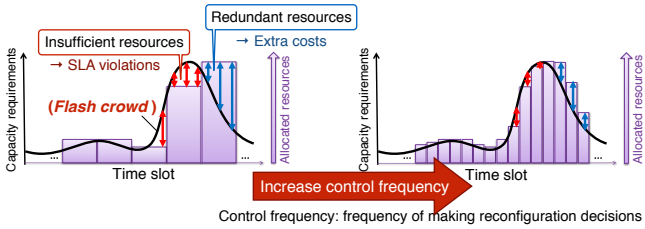
- Proactive resource allocation needs to predict the future demands, but prediction errors result in resource inefficiency.






## Resource Scaling Approach

- Proactive resource allocation needs to predict the future demands, but prediction errors result in resource inefficiency.
- High control frequency make resource reconfigurations adapt more quickly to demand changes, but it also increase reconfiguration overheads.
- *Model predictive control* (*MPC* ) is applied to balance between resource efficiency and reconfiguration overheads.

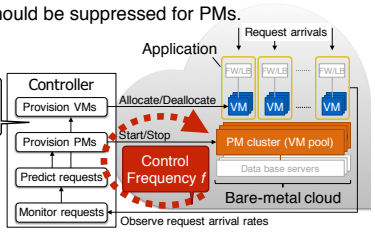



Control frequency: frequency of making reconfiguration decisions

# Challenges and Proposals

- PMs have larger reconfiguration overheads and need a longer lead time than VMs.
- Challenges
  - The controller should initiate the reconfiguration process of PMs before initiating that of VMs
  - Excessive reconfigurations should be suppressed for PMs.
- Proposal

**Two-level hierarchical MPC with two different reconfiguration weight factors**






# Scaling Model: Physical Machines

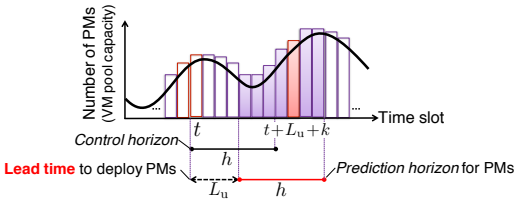
- The controller solves the optimization problem to balance the redundant PMs and the reconfiguration overhead.

**Objective:** minimize

Redundant PMs

Reconfigured PMs

$$J_{\mathrm{u}} = \left(\frac{C_{\mathrm{u}}}{f}\right)^2 \sum_{l=L_{\mathrm{u}}}^{L_{\mathrm{u}}+h-1} \left(x(t+l|t) - x^{\min}(t+l|t)\right)^2 + W_{\mathrm{u}}^2 \sum_{k=0}^{h-1} u(t+k|t)^2$$






# Scaling Model: Physical Machines

- The controller solves the optimization problem to balance the redundant PMs and the reconfiguration overhead.

**Objective:** minimize

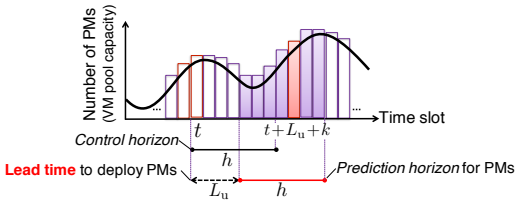
Redundant PMs

Reconfigured PMs

$$J_{\mathrm{u}} = \left(\frac{C_{\mathrm{u}}}{f}\right)^2 \sum_{l=L_{\mathrm{u}}}^{L_{\mathrm{u}}+h-1} \left(x(t+l|t) - x^{\min}(t+l|t)\right)^2 + W_{\mathrm{u}}^2 \sum_{k=0}^{h-1} u(t+k|t)^2$$

Allocated PMs

Minimum PMs






# Scaling Model: Physical Machines

- The controller solves the optimization problem to balance the redundant PMs and the reconfiguration overhead.

**Objective:** minimize

Redundant PMs

Reconfigured PMs

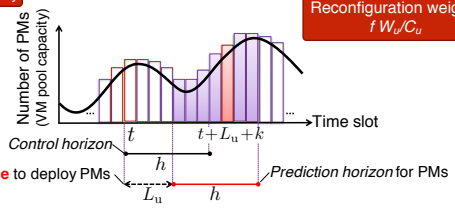
$$J_{\mathrm{u}} = \left(\frac{C_{\mathrm{u}}}{f}\right)^2 \sum_{l=L_{\mathrm{u}}}^{L_{\mathrm{u}}+h-1} \left(x(t+l|t) - x^{\min}(t+l|t)\right)^2 + W_{\mathrm{u}}^2 \sum_{k=0}^{h-1} u(t+k|t)^2$$

*f* : Control frequency

Allocated PMs

Minimum PMs

Reconfiguration weight: $f\,W_u/C_u$






# Scaling Model: Virtual Machines

- The controller solves the optimization problem to balance the risk of VM insufficiency the reconfiguration overhead.

**Objective:** minimize

Risk of VM insufficiency

Reconfigured VMs

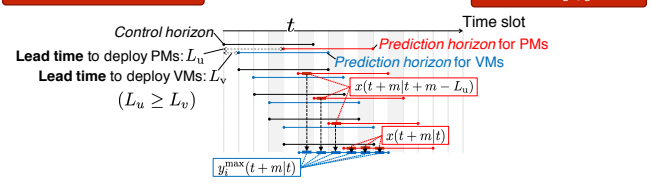
$$J_{\mathrm{v}i} = \left(\frac{C_{\mathrm{u}}}{fN}\right)^2 \sum_{m=L_{\mathrm{v}}}^{L_{\mathrm{v}}+h-1} \left(y_i^{\max}(t+m|t) - y_i(t+m|t)\right)^2 + W_{\mathrm{v}}^2 \sum_{k=0}^{h-1} v_i(t+k|t)^2$$

- *f* : Control frequency
- N: Number of VMs per PMs

Available VMs

Allocated VMs

Reconfiguration weight: $f\,NW_v/C_u$






# Evaluation

- Evaluate the proposed MPC using three HTTP traces from real-world web application: *World Cup*, *Campus* and *Video*.
- Predict future request arrival rates with *ARIMA* model.
- Find the optimal numbers of allocated resources using *Dynamic Programming*.

# Evaluation

- Evaluate the proposed MPC using three HTTP traces from real-world web application: *World Cup*, *Campus* and *Video*.
- Predict future request arrival rates with *ARIMA* model.
- Find the optimal numbers of allocated resources using *Dynamic Programming*.
- Focus on clarifying the effect of high-frequency control with various reconfiguration weights for PMs.
  - Experimental setup

| Control frequency | | $f$ | 1 - 16 time(s) per hour |
|---|---|---|---|
| Lead time | PMs | $L_u$ | 1 - $f$ time slots |
| | VMs | $L_v$ | |
| Length of control and prediction horizon | | $h$ | $f$ time slots (fixed) |
| Reconfiguration weight | PMs | $W_u/C_u$ | **0.1 - 1.4** |
| | VMs | $W_v/C_u$ | |
| Number of VMs per PM | | $N$ | |



# Evaluation

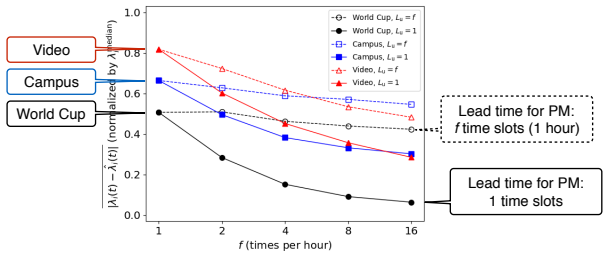
- Evaluate the proposed MPC using three HTTP traces from real-world web application: *World Cup*, *Campus* and *Video*.
- Predict future request arrival rates with *ARIMA* model.
- Find the optimal numbers of allocated resources using *Dynamic Programming*.
- Focus on clarifying the effect of high-frequency control with various reconfiguration weights for PMs.
  - Experimental setup

| Control frequency | | $f$ | 1 - 16 time(s) per hour |
|---|---|---|---|
| Lead time | PMs | $L_u$ | 1 - $f$ time slots |
| | VMs | $L_v$ | 1 time slot (fixed) |
| Length of control and prediction horizon | | $h$ | $f$ time slots (fixed) |
| Reconfiguration weight | PMs | $W_u/C_u$ | **0.1 - 1.4** |
| | VMs | $W_v/C_u$ | 0.01 (**fixed at the lower**) |
| Number of VMs per PM | | $N$ | 4 (fixed) |



# Prediction Errors

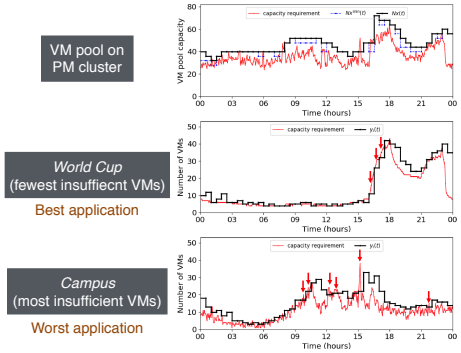
- The prediction errors are mainly caused
  - by a large spike lasting a few hours in the case of *World Cup*
  - by a fluctuation during several tens of minutes in the cases of *Campus* and *Video*






# Example of Allocated Resources

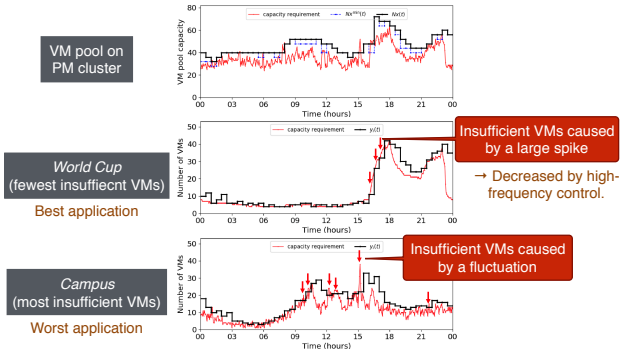
- Capacity requirements and allocated resources over a 1-day period ($f = 2$, $W_u/C_u = 0.7$, $L_u = 1$)






# Example of Allocated Resources

- Capacity requirements and allocated resources over a 1-day period ($f = 2$, $W_u/C_u = 0.7$, $L_u = 1$)

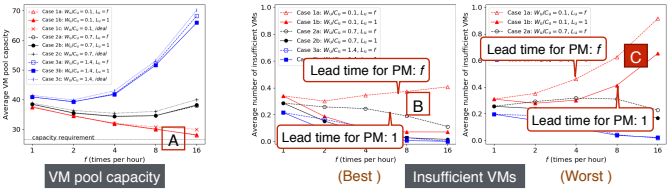





# Effect of Control Frequency

- When control frequency ($f$) increases (from 1 to 16) ↗,

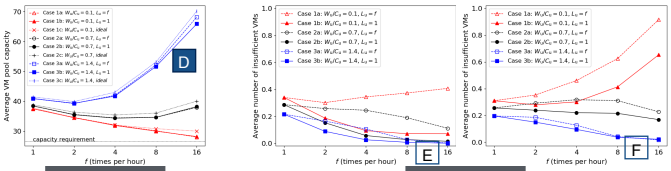
| | Reconfiguration weight for PMs ($W_u/C_u$) | | | | |
|---|---|---|---|---|---|
| | Lower ($W_u/C_u = 0.1$) | | | Upper ($W_u/C_u = 1.4$) | |
| | PMs | VMs (Best) | VMs (Worst) | PMs | VMs (Best) | VMs (Worst) |
| Reconfigurations | ↗ | ↗ | ↗ | | | |
| VM pool capacity | A | | | | | |
| Insufficient VMs | | B | C | | | |



# Effect of Control Frequency (Continued)

- When control frequency ($f$) increases (from 1 to 16) ↗,

| | Reconfiguration weight for PMs ($W_u/C_u$) | | | | | |
|---|---|---|---|---|---|---|
| | Lower ($W_u/C_u$ = 0.1) | | | Upper ($W_u/C_u$ = 1.4) | | |
| | PMs | VMs | | PMs | VMs | |
| | | (Best) | (Worst) | | (Best) | (Worst) |
| Reconfigurations | ↗ | ↗ | ↗ | ↘ | ↗ | ↗ |
| VM pool capacity | ↘ | | | ↗ D | | |
| Insufficient VMs | | → | ↗ | | ↘ E | ↘ F |




VM pool capacity (Best ) Insufficient VMs (Worst )

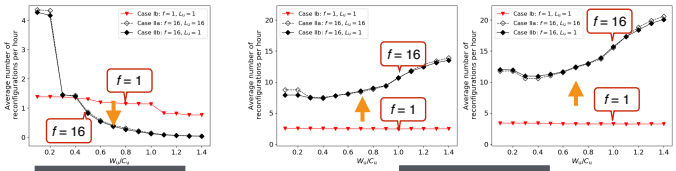


# Effect of Reconfiguration Weight for PMs

- How much should weight fo factor ($W_u/C_u$) be given to high-frequency control?

| | | Reconfiguration weight for PMs ($W_u/C_u$) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 |
| Reconfigurations | PMs | + | | - | - | - | - | - |
| | VMs (best) | + | + | + | + | + | + | + |
| | VMs (worst) | + | + | + | + | + | + | + |

Difference of high frequency control ($f$=16) to the existing settings (i.e., $f$ =1)



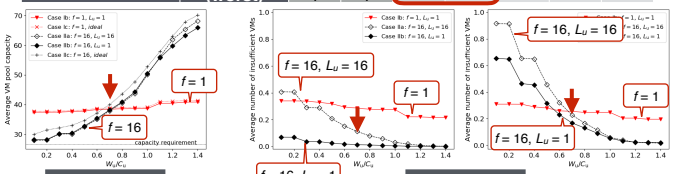

PM Reconfigurations (Best ) VM Reconfigurations (Worst )



# Effect of Reconfiguration Weight for PMs (Continued)

- How much should reconfiguration weight for PMs ($W_u/C_u$) be given?

| | | Reconfiguration weight factor ($W_u/C_u$) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 |
| Reconfigurations | PMs | + | | - | - | - | - | - |
| | VMs (best) | + | + | + | + | + | + | + |
| | VMs (worst) | + | + | + | + | + | + | + |
| VM pool capacity | | - | - | - | | + | + | + |
| Insufficient VMs | (best) | + | | - | - | - | - | - |
| | (worst) | + | + | | - | - | - | - |




VM pool capacity (Best ) Insufficient VMs (Worst )



# Conclusion

- High-frequency control of hierarchical and frequency-aware MPC
  - Improve the timing of the PM reconfigurations, and increase the VM reallocations to adjust the redundant capacity among the applications
  - Lead to the reduction of VM insufficiency without increasing the resource redundancy level
- Future work
  - Evaluations with various control options
  - Evaluations with different combinations of the three applications; each of the combinations has different request arrival characteristics



**Thank you!**
**Questions?**

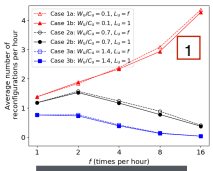


Backup Slides

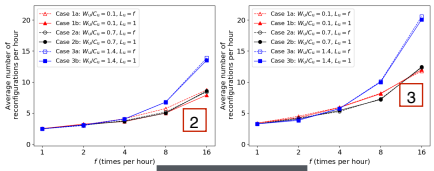
# Effect of Control Frequency

- When control frequency ($f$) increases (from 1 to 16) ↗,

| | Reconfiguration weight for PMs ($W_u/C_u$) | | | | | |
|---|---|---|---|---|---|---|
| | Lower ($W_u/C_u$ = 0.1) | | | Upper ($W_u/C_u$ = 1.4) | | |
| | PMs | VMs | | PMs | VMs | |
| | | (Best) | (Worst) | | (Best) | (Worst) |
| Reconfigurations | ↗ 1 | ↗ 2 | ↗ 3 | | | |




PM Reconfigurations




VM Reconfigurations

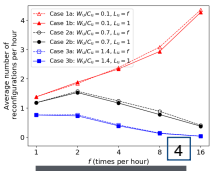


# Effect of Control Frequency

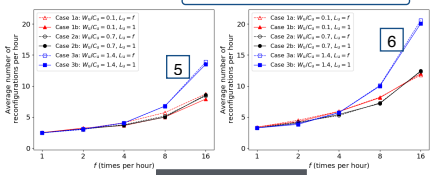
- When control frequency ($f$) increases (from 1 to 16) ↗,

| | Reconfiguration weight for PMs ($W_u/C_u$) | | | | | |
|---|---|---|---|---|---|---|
| | Lower ($W_u/C_u$ = 0.1) | | | Upper ($W_u/C_u$ = 1.4) | | |
| | PMs | VMs | | PMs | VMs | |
| | | (Best) | (Worst) | | (Best) | (Worst) |
| Reconfigurations | ↗ | ↗ | ↗ | ↘ 4 | ↗ 5 | ↗ 6 |

VM pool capacity is held, but VM reallocations become active.




PM Reconfigurations




VM Reconfigurations