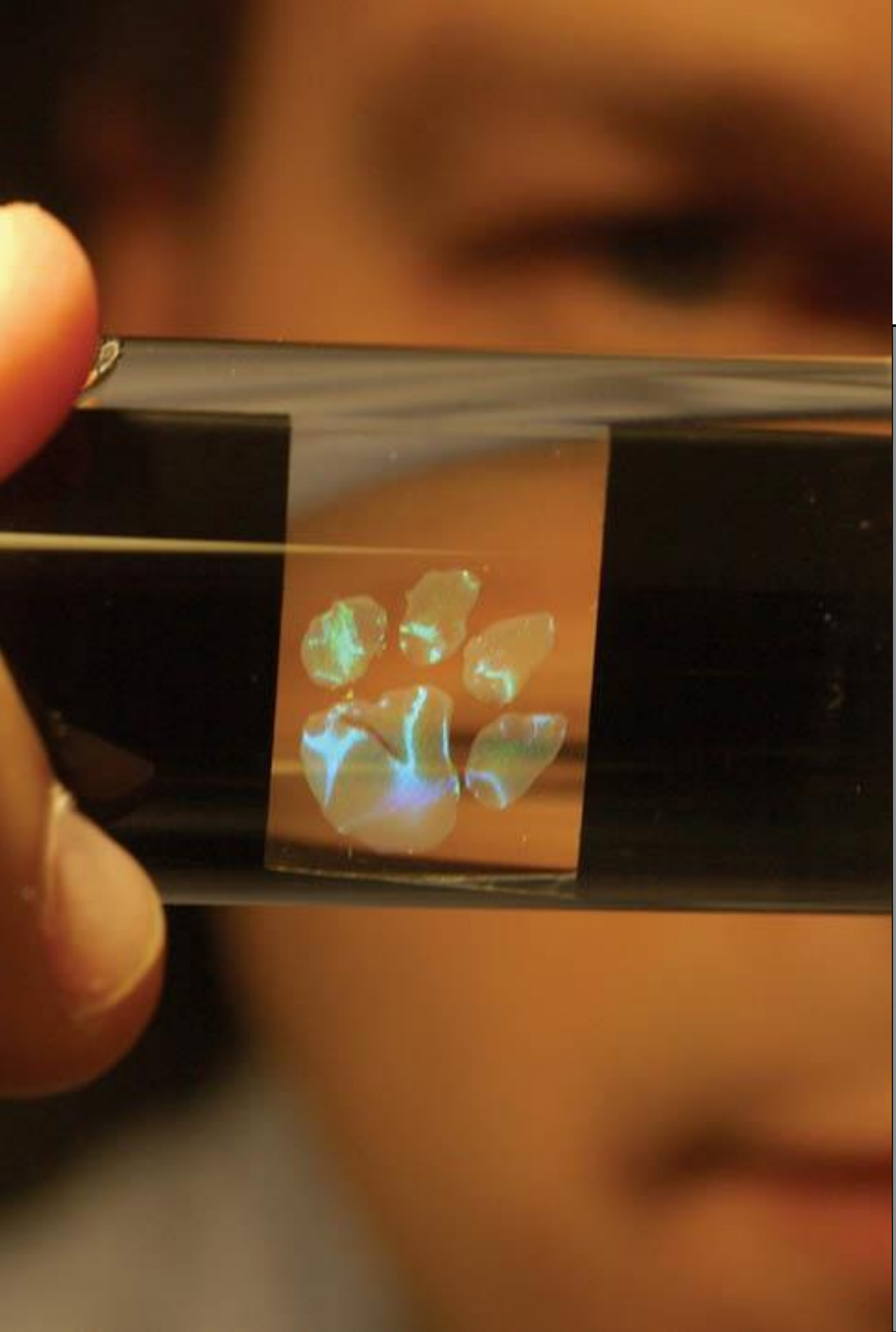


CLEMSON UNIVERSITY

CCRP: Customized Cooperative Resource Provisioning for High Resource Utilization in Clouds

Jinwei Liu*, Haiying Shen† and Husnu S. Narman*

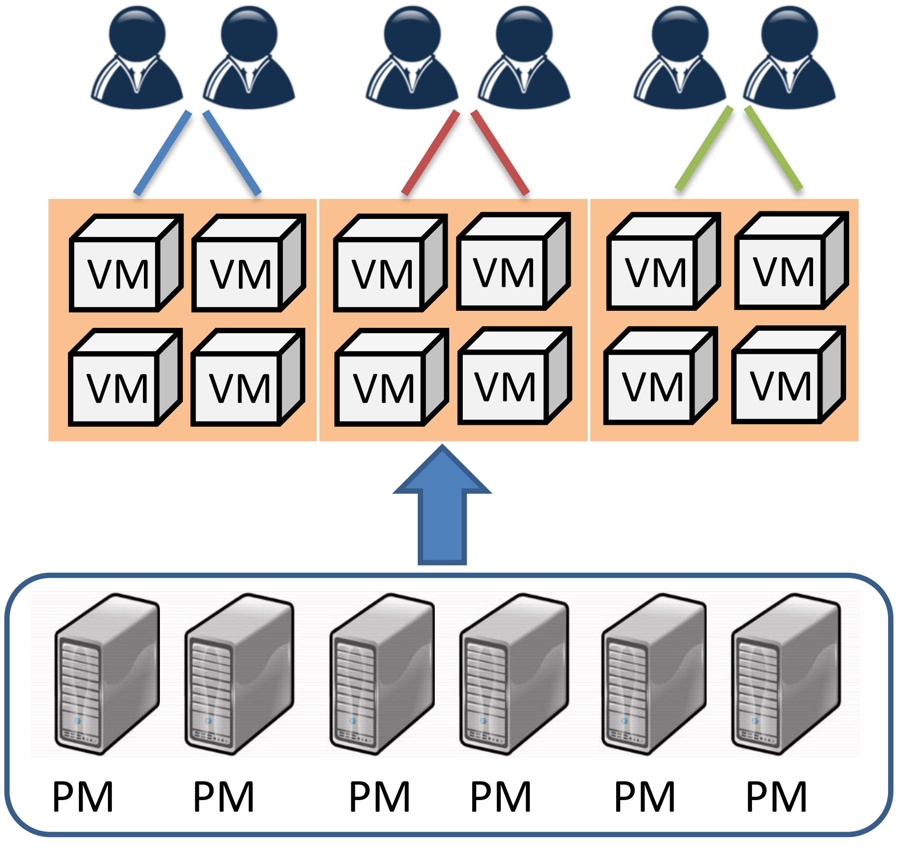
*Dept. of Electrical and Computer Engineering Clemson University, SC, USA
†Dept. of Computer Science, University of Virginia, Charlottesville, VA, USA

Outline

- Introduction
- Customized Cooperative Resource Provisioning (CCRP)
- Design of CCRP
- Performance Evaluation
- Conclusions

Introduction

- Resource allocation in cloud computing



Introduction (cont.)

- Resource allocation options
  - Reservation-based resource allocation
    - Amazon Reserved Instance

amazon web services

Amazon EC2 Reserved Instances

  - Demand-based resource allocation
    - Amazon On-demand Instances

Amazon Web Services EC2
On-Demand Instances

  - Opportunistic-based resource allocation
    - Amazon Spot Instances, [Carvalho, SoCC'14]

AMAZON EC2 SPOT INSTANCES

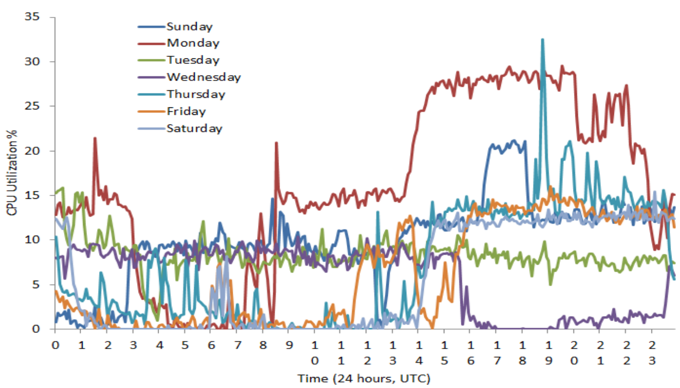
AWS.AMAZON.COM/EC2/SPOT

Motivation

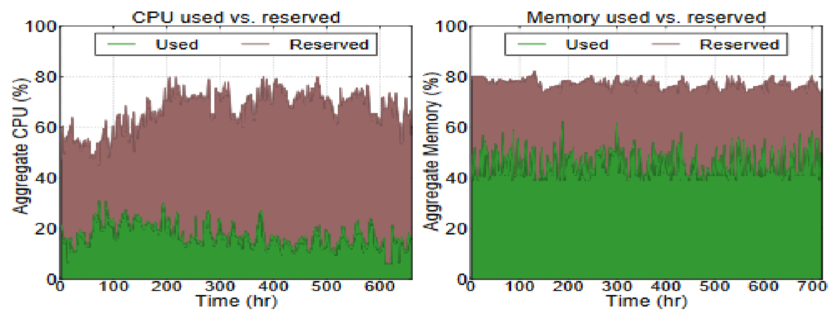
- Limitation of the three resource allocation options
  - Reservation-based resource allocation
    - Much more resources are wasted
    - Higher cost
  - Demand-based resource allocation
    - Resources cannot be fully utilized all the time
    - High cost
  - Opportunistic-based resource allocation
    - Either have no SLOs or cannot ensure SLO availability for both long jobs and short jobs

Motivation (cont.)

- Resource wastage & low resource utilization in existing cloud systems
  - Amazon EC2 servers have low resource utilization
  - Production cluster at Twitter has low resource utilization



Amazon EC2 servers

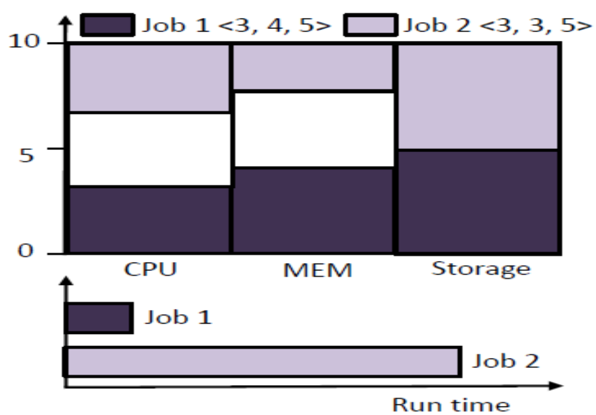


Production cluster at Twitter [1]

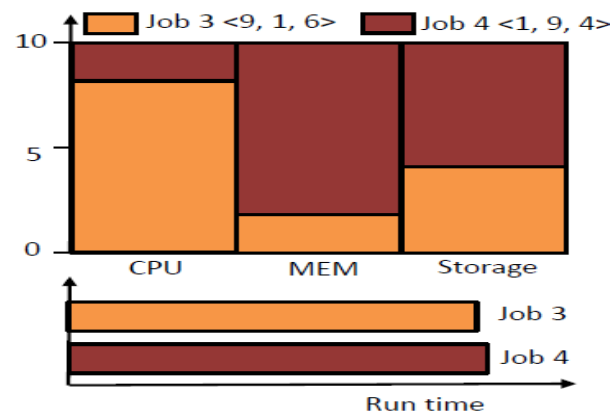
[1] C. Delimitrou and C. Kozyrakis. Quasar: Resource-Efficient and QoS-Aware Cluster Management. In *Proc. of ASPLOS*, 2014.

Motivation (cont.)

- Resource wastage caused by resource fragmentation
  - Neglect job packing or consider packing w/o avoiding fragmentation
  - Neglect job size for packing[1]



W/o avoiding fragmentation or considering job size



W/ avoiding fragmentation and considering job size

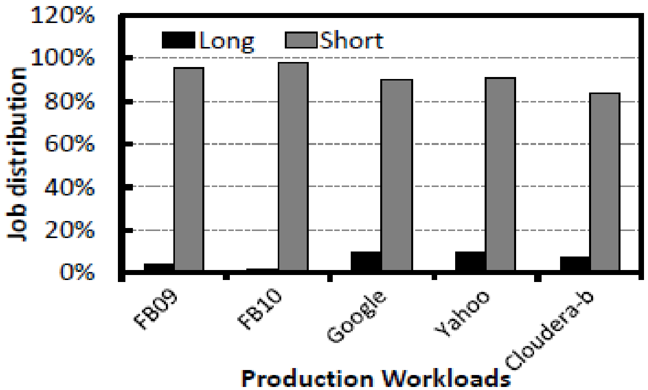
[1] To ensure higher availability SLO, CCRP uses a demand-based resource allocation for long jobs, and the VMs allocated to users' jobs cannot be reallocated to other jobs until users' jobs finish execution [2, 3].

[2] I. Pietri, G. Juve, E. Deelman, and R. Sakellariou. A performance model to estimate execution time of scientific workflows on the cloud. In *WORKS'14*, pages 11-19, 2014.

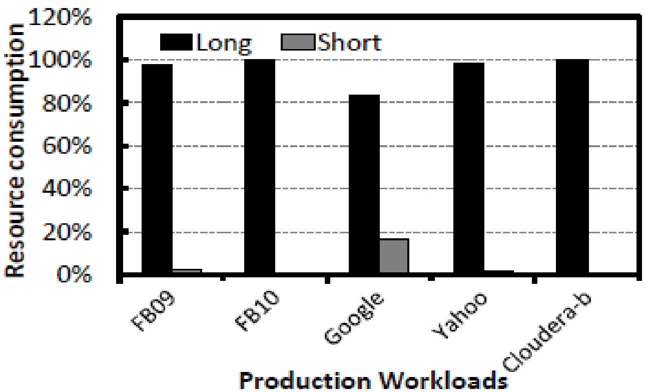
[3] M. Carvalho, W. Cirne, F. Brasileiro, and J. Wilkes. Long-term slos for reclaimed cloud computing resources. In *Proc. of SoCC*, 2014.

Motivation (cont.)

- Diversity on job size and resource consumption of jobs



Job distribution based on job size of various commercial workloads



Distribution of jobs' resource consumption based on job size of various of commercial workloads

- Therefore, it is important to consider job size for packing, which can reduce resource wastage
- Goal: design an efficient resource allocation scheme for heterogeneous jobs (both long jobs and short jobs) that can increase resource utilization while ensuring SLO availability

Outline

- Introduction
- Customized Cooperative Resource Provisioning (CCRP)
- Design of CCRP
- Performance Evaluation
- Conclusions

Customized Cooperative Resource Provisioning (CCRP) Problem

- Problem statement
  - Problem: Given a certain amount of resources, resource demands of each job, resource capacity constrains of VMs, how to allocate the VM resources to heterogeneous jobs to achieve higher resource utilization while avoiding SLO violations as much as possible?

Challenges of Resource Provisioning

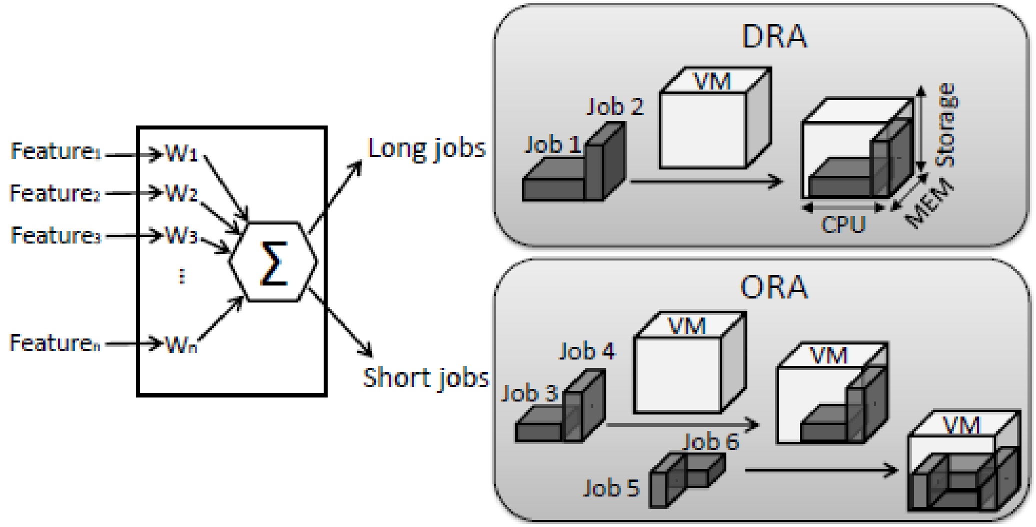
- Challenges of resource provisioning for heterogeneous jobs
  - How to determine the most appropriate resource provisioning strategy for jobs with different sizes
  - How to accurately predict the amount of unused resource of short jobs with resource fluctuations
  - How to reduce resource fragmentation in multi-resource allocation for heterogeneous jobs
  - How to fully utilize the resource while ensuring high SLO availability

Outline

- Introduction
- Customized Cooperative Resource Provisioning (CCRP)
- Design of CCRP
- Performance Evaluation
- Conclusions

Design of CCRP

- Key idea: Use opportunistic-based resource allocation for short jobs and demand-based resource allocation for long jobs to achieve high resource utilization and low SLO violation rate



Architecture of the hybrid resource allocation scheme CCRP: DRA (demand-based resource allocation for long jobs) and ORA (opportunistic-based resource allocation for short jobs)

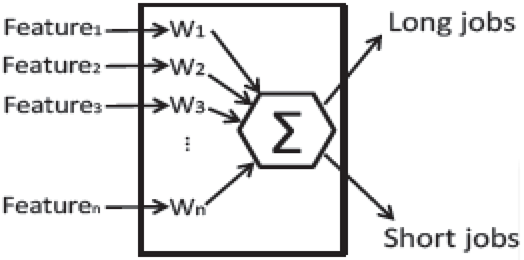
Job Classification

- Support Vector Machine (SVM) for job classification
  - Predicting execution time of jobs
    - Two types of features for predicting jobs' execution time

Feature	Description
Job-related features	
Required CPU	Amount of CPU resource required by the job
Required MEM	Amount of MEM resource required by the job
Required storage	Amount of storage resource required by the job
# of tasks	Number of tasks which the job contains
Priority of job	The priority of the job
System-related features	
CPU utilization	VM's CPU utilization
MEM utilization	VM's MEM utilization
Storage utilization	VM's storage utilization

Features for predicting jobs' execution time

  - Jos classification



Opportunistic-based Resource Allocation

- Predict unused resource
  - Predicting the amount of unused resource using deep learning with Semi-Markov Model (SMM)
    - Feed-forward evaluation

$$\phi_i(d) = F((\Sigma_{j=1}^{c} \omega_{ij}(d-1,d) \cdot \phi_j(d-1)) + b_i)$$

    - Back-propagation

$$E_i(d_h) = (t_i(d_h) - \phi_i(d_h)) \cdot F'(\phi_i(d_h))$$

$$E_i(d) = (\sum_{j=1}^{v} E_j(d+1) \cdot |\omega_{ji}(d,d+1)) \cdot F'(\phi_i(d))$$

    - Weight updates

$$\Delta\omega_{ij}(d-1,d) = \beta \cdot E_i(d) \cdot \phi_j(d-1),\ \forall j = 1, ..., c$$

Predict Unused Resource (cont.)

- Predicting fluctuations of the amount of unused resource using SMM
    - Why predict fluctuations of unused resource?
        - Short jobs usually do not exhibit certain resource utilization, which results in the fluctuations of the amount of unused resource
    - Solution: use SMM model to predict the fluctuations of the amount of unused resource
  - SMM model
    - Three states: Peak, Center, Valley

Predict Unused Resource (cont.)

- Error correction based on observation symbols
  - Predicted states:
    - Peak: $\hat{u}_{t+L} = \hat{\xi} + \min(h_{cpu} - m_{cpu}, m_{cpu} - l_{cpu})$
    - Valley: $\hat{u}_{t+L} = \hat{\xi} - \min(h_{cpu} - m_{cpu}, m_{cpu} - l_{cpu})$
      - h_{cpu}: the highest amount of unused resource
      - l_{cpu}: the lowest amount of unused resource
      - m_{cpu}: the average amount of unused resource
  - Rationale: 1) $h_{cpu} - m_{cpu}$ (or $m_{cpu} - l_{cpu}$) indicates the deviation between the amount of unused resource in peak (or valley) and the ave. of the amount of unused resource; 2) min is more conservative for ensuring sufficient resource being able to allocate to jobs

Resource Reallocation

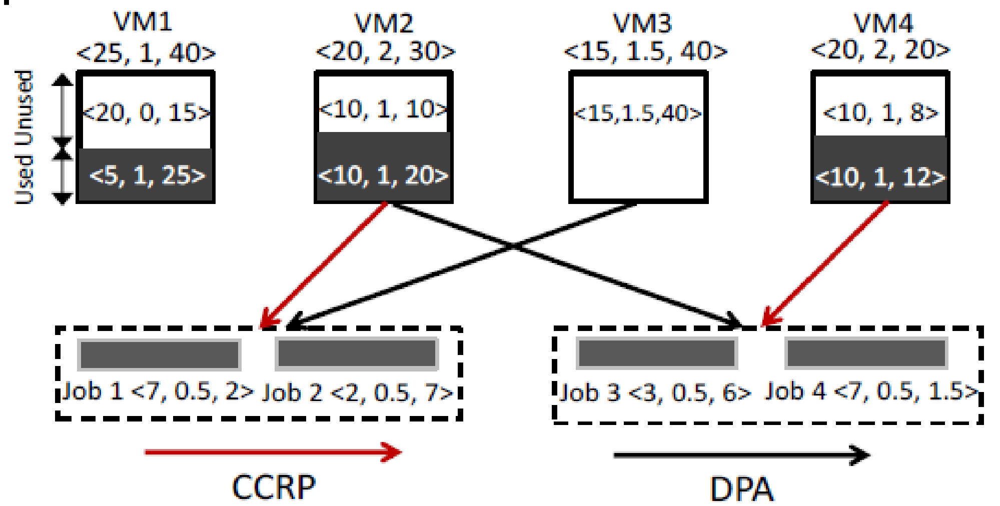
- Condition for resource reallocation

$$\Pr(0 \leq \delta_{t+L} \leq \epsilon) \geq \Pr^{th} \quad (1)$$

  - ϵ : pre-specified prediction error tolerance
  - $\Pr^{th}$: pre-defined probability threshold
  - δ_{t+L}: prediction error

Job Packing

- Leverage complementarity of jobs' requirements on different resource types



An example of allocating resources to jobs using packing strategy

$$DV(j,i) = \sum_{k=1}^{l}((d_{jk} - \frac{d_{jk}+d_{ik}}{2})^2 + (d_{ik} - \frac{d_{jk}+d_{ik}}{2})^2) \quad (2)$$

$$VOL_j = \sum_{i=1}^{l} \frac{\hat{R}_{ji}}{C_i^*} \quad (3)$$

$$vol_i = \sum_{k=1}^{l} \frac{r_{ik}}{C_k^*} \quad (4)$$

Outline

- Introduction
- Customized Cooperative Resource Provisioning (CCRP)
- Design of CCRP
- Performance Evaluation
- Conclusions

Performance Evaluation

- Methods for comparison
  - RCCR [3]: Opportunistic-based resource allocation based on time series forecasting

    [3] M. Carvalho, W. Cirne, F. Brasileiro, and J. Wilkes. Long-term SLOs for reclaimed cloud computing resources. In *Proc. of SoCC*, Seattle, 2014.
  - CloudScale [4]: Demand-based resource allocation based on FFT (fast Fourier transform) for prediction

    [4] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes. Cloudscale: Elastic resource scaling for multi-tenant cloud systems. In *Proc. of SoCC*, Cascais, Oct. 2011.
  - DDA [5]: Demand-based resource allocation based on monitoring

    [5] G. Shanmuganathan, A. Gulati, and P. Varman. Defragmenting the cloud using demand-based resource allocation. In *Proc. of SIGMETRICS*, pages 67-80, Pittsburgh, June 2013.
  - Tetris [6]: Dot product-based resource packing

    [6] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella. Multi-resource packing for cluster schedulers. In *Proc. of SIGCOMM*, 2014.

Experiment Setup

- Trace-driven experiments on a real cluster Palmetto & Amazon EC2
  - Nodes deployment
    - 50 Nodes in Palmetto [7]
    - 30 Nodes in Amazon EC2 [8]
  - Trace from Google [9]
  - Set CPU & MEM consumption based on Google trace [9]

[7] Palmetto cluster. http://citi.clemson.edu/palmetto/.
[9] Amazon EC2. http://aws.amazon.com/ec2.
[9] Google trace. https://code.google.com/p/googleclusterdata/.

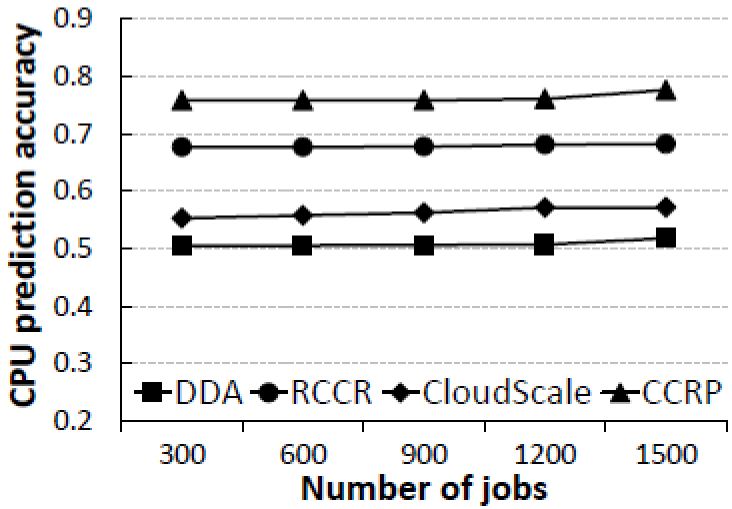
Experiment Setup (cont.)

- Parameter settings

Parameter	Meaning	Setting	Parameter	Meaning	Setting
N_p	# of servers	30-50	h	# of layers in DNN	4
N_v	# of VMs	100-400	N_n	# of units per layer	50
$\|J\|$	# of jobs	300-1500	M	# of states in SMM	3
l	# of resc. types	3	α	Significance level	5%-30%
$\Pr^{th}$	Prob. threshold	0.95	η	Confidence level	50%-90%

Evaluation of CCRP

- Experimental results on the real cluster
  - Prediction accuracy

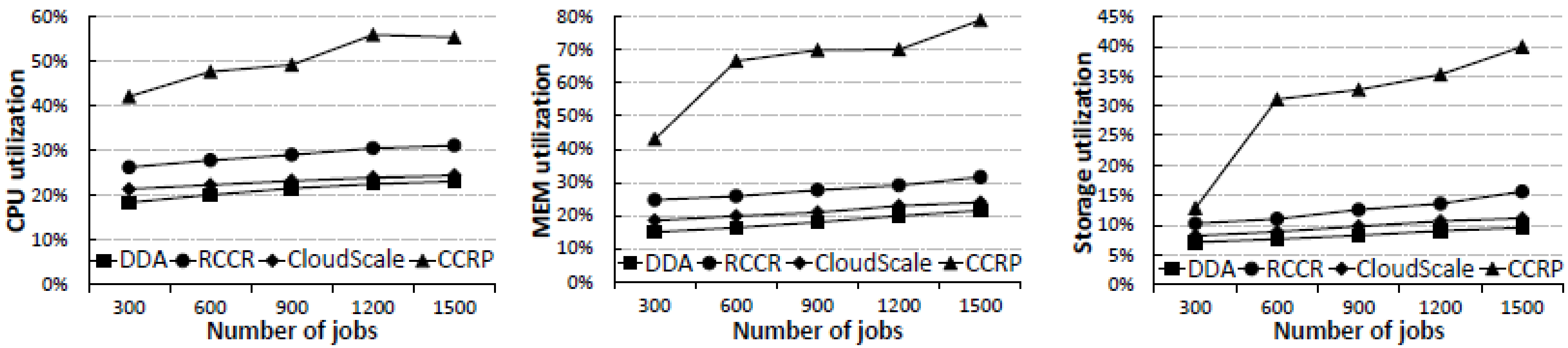


Prediction accuracy of CPU

  - Prediction accuracy: CCRP > RCCR > CloudScale > DDA
  - Reasons: advantages of deep learning; CCRP considers fluctuations of amount of unused resource caused by short jobs' time-varying resource demands, uses SMM to correct prediction errors

Evaluation of CCRP

- Experimental results on the cluster
  - Resource utilization

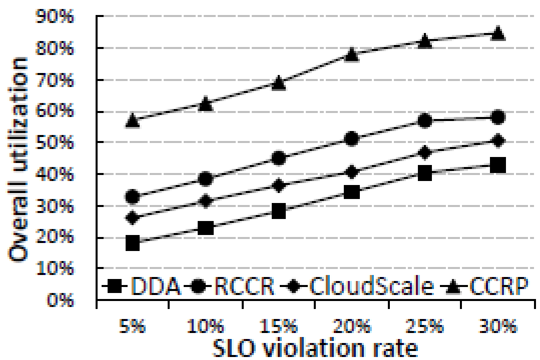


(a) CPU **(b) Memory** **(b) Storage**

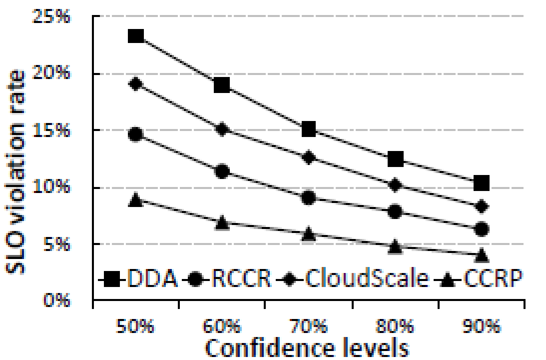
  - Utilization of CPU, MEM and storage: CCRP > RCCR > CloudScale > DDA
  - Reasons: opportunistic-based resource allocation for short jobs; use deep learning & SMM to accurately predict the unused resource and avoid over-provisioning; use job packing to reduce resource fragmentation

Evaluation of CCRP

- Experimental results on the cluster
  - Overall resource utilization & SLO violation rate



(a) Resource utilization vs. SLO violation rate

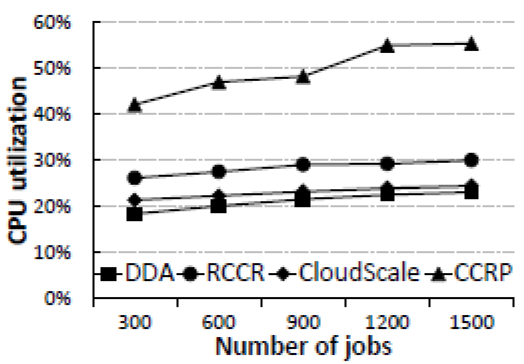


(b) SLO violation rate vs. confidence levels

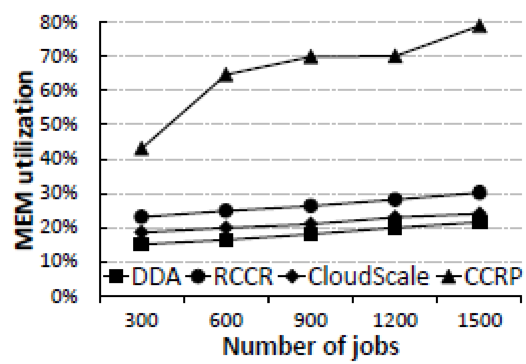
  - Observation: resource utilization increases as SLO violation rate increases; overall resource utilization: CCRP > RCCR > CloudScale > DDA
    SLO violation rate: CCRP < RCCR < CloudScale < DDA
  - Reasons: the higher the SLO violation rate, the lower the probability of over-provisioning occurring; advantages of deep learning; CCRP considers fluctuations, uses SMM to correct prediction errors

Evaluation of CCRP

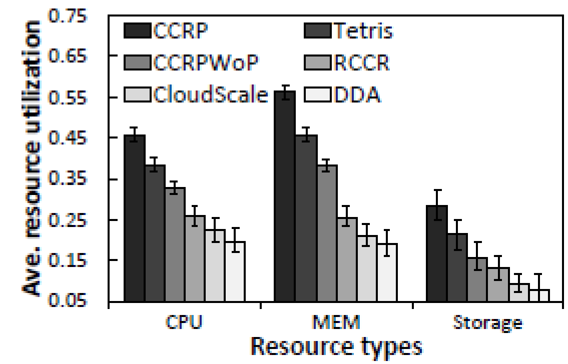
- Experimental results on Amazon EC2
  - Resource utilization



(a) CPU



(b) Memory

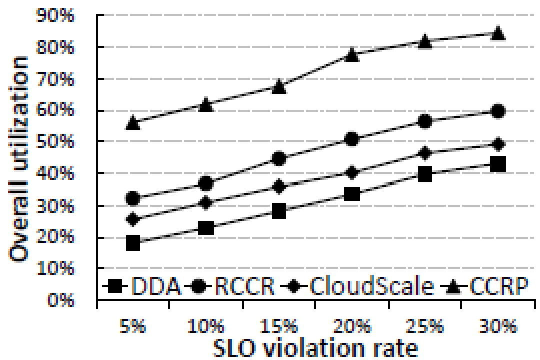


(b) Ave. resource utilization

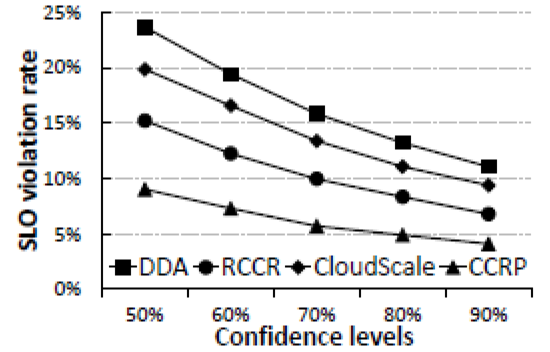
  - Observation: Utilization of CPU, MEM and Ave. resource: CCRP > RCCR > CloudScale > DDA
  - Reasons: opportunistic-based resource allocation for short jobs; use deep learning & SMM to accurately predict the unused resource and avoid over-provisioning; use job packing to reduce resource fragmentation

Evaluation of CCRP

- Experimental results on Amazon EC2
  - Overall resource utilization & SLO violation rate



(a) Resource utilization vs. SLO violation rate on Amazon EC2



(b) SLO violation rate vs. confidence levels on Amazon EC2

- Observation:

  a) resource utilization increases as SLO violation rate increases; given an SLO violation rate, overall resource utilization follows CCRP > RCCR > CloudScale > DDA

  b) SLO violation rate decreases as the confidence levels increases; SLO violation rate follows CCRP < RCCR < CloudScale < DDA

Outline

- Introduction
- Customized Cooperative Resource Provisioning (CCRP)
- Design of CCRP
- Performance Evaluation
- Conclusions

Conclusions

- Our contributions
  - Present a customized cooperative resource provisioning scheme for high resource utilization and low SLO violation rate by using a hybrid resource allocation for heterogeneous jobs
  - Predict jobs' execution time and classify jobs into two types; accurately predict the amount of unused resource using deep learning and SMM model with considering the fluctuations of the amount of unused resource
  - Consider jobs' complementary resource requirements and jobs' heterogeneity in job size, and present a job packing strategy to reduce the resource fragmentation
  - Extensive experimental results based on a real cluster and Amazon EC2 validate the performance of CCRP

- Future work
  - The fluctuation of the workloads

Thank you!

Questions & Comments?

Jinwei Liu, PhD

jinweil@clemson.edu

Electrical and Computer Engineering

Clemson University