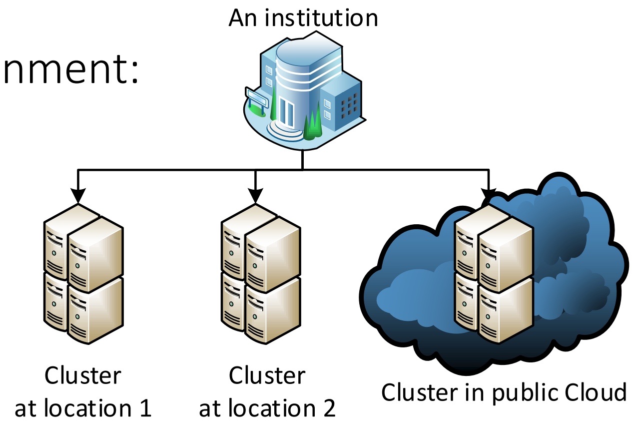
Deep Reinforcement Learning based Elasticity-compatible Heterogeneous Resource Management for Time-critical Computing

Zixia Liu, University of Central Florida
Liqiang Wang, University of Central Florida
Gang Quan, Florida International University

Background

- Expanding needs for data analytics call for greater scale computing infrastructure, multi-cluster computing environment shows its benefits and necessity in this.
  - Example: institution-owned geo-distributed clusters, hybrid-cloud, etc.
- An efficient resource management is needed.
- Many features to consider for resource management, also including cluster heterogeneity and elasticity.
- To consider features in an integration, We presents a DRL based resource management in such environment.

An example of a multi-cluster environment:

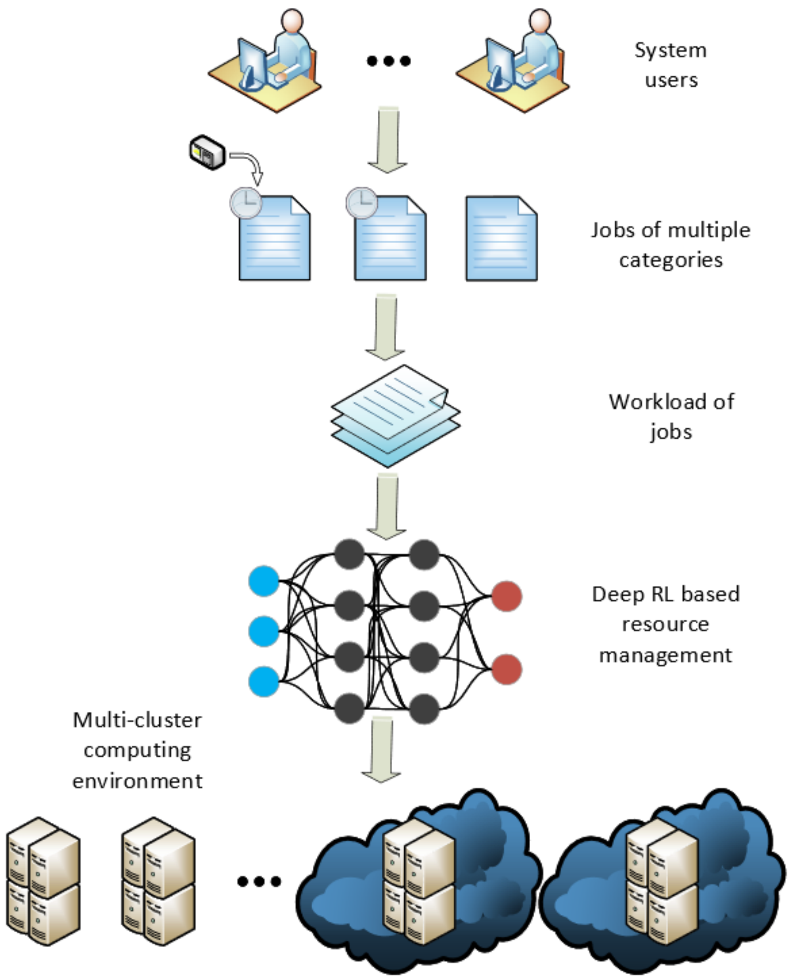


Contribution

- We propose a DRL based approach utilizing:
  - LSTM model and
  - multi-target regression with partial model sharing mechanism

  and compare its effectiveness with baselines and another RL approach.
- The approach is designed for distributed multi-cluster computing environments considering:
  - its heterogeneity and
  - being elasticity-compatible.
- It provides scheduling support for time-critical computing in such a multi-cluster environment.

Problem Description

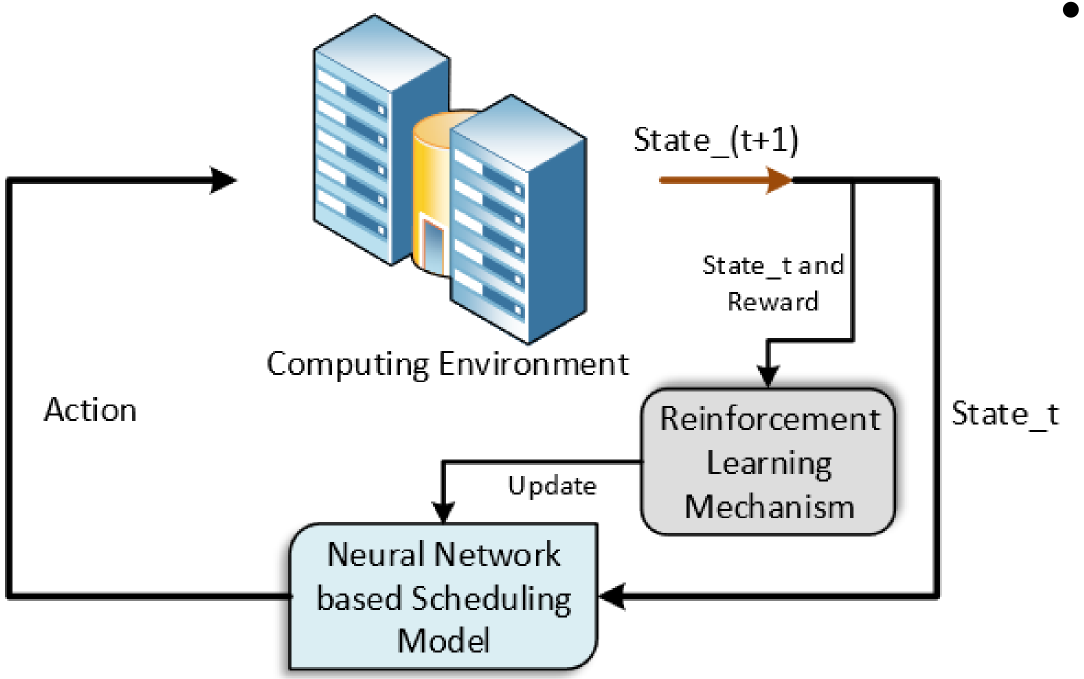


- Cluster in environment expresses its computing resources as the number of executors it could provide.
- Executors of different clusters may have different computing capabilities.
- Some clusters may be elastic.

- Goals for resource management:
  (1) Reducing occurrences of missing temporal deadline events.
  (2) Maintaining a low average execution time ratio for a hybrid workload containing multiple time-critical and general jobs.

DRL based Approach

- Brief introduction of Reinforcement learning



- We are using:
  - Reinforcement learning on deep neural networks
  - With neural networks serving as value estimators.

DRL based Approach

- Challenges:
  - How to represent system status and job information as state for such environment?
  - How should we define value?
  - Effective value estimator?

- Environment
- Action set
- Episode
- State
  - Computing system features and status
  - Scheduling job information

Table 1: State representation in our deep reinforcement learning model for 5 clusters

Vector Component	**Dimensions**
Cluster ($i = 1 \ldots 5$)	
Cluster sequence number	1
Normal capacity	1
Maximum capacity if elastic	1
Cluster heterogeneity factor	1
Occupation status (latest 105 steps)*	$105 \rightarrow 150$
Current total missing deadlines	1
	$775\ (155 \times 5)$
Job	
Category	1
Expected heterogeneity factor	1
Heterogeneity sensitivity	1
Discrete resource request distribution	10×5
Standard execution time	1×5
Execution deadline	1×5
Duration	1
	64
Overall state vector	**839 (775 + 64)**
Only * row includes temporal information	

DRL based Approach

- Value definition ideas:
  - Attend to causes of missing deadlines.
  - Attend to job's influence on resource competition.
  - Attend to mutual influences among jobs in cluster.
  - Attend to influences of heterogeneity and elasticity.
  - Attend to both missing deadlines and execution delay ratio.
- Value formula: $v^{(j)} = \frac{\eta_c \cdot m_{ih} \cdot m_{ic}}{-\eta_j} \left[M_j + \sum_{t=t_s}^{t_e} \beta^{D_t} \left(W_j^{(t)} + W_{cl}^{(t)} \right) \right] - \psi_{ih} \cdot \psi_{ic} \cdot R_j$

η_c: The heterogeneity factor of the cluster.

η_j: The expected heterogeneity factor of the job.

M_j: The number of missing deadlines of job j without resource waiting.

$W_j^{(t)}$: The happening of each missing deadline event of job j at moment t, if not in M_j.

$W_{cl}^{(t)}$: The number of missing deadlines of all jobs in the cluster at t if with resource waiting.

t_s and t_e: The deployment and termination moment of job j.

β: The decay factor.

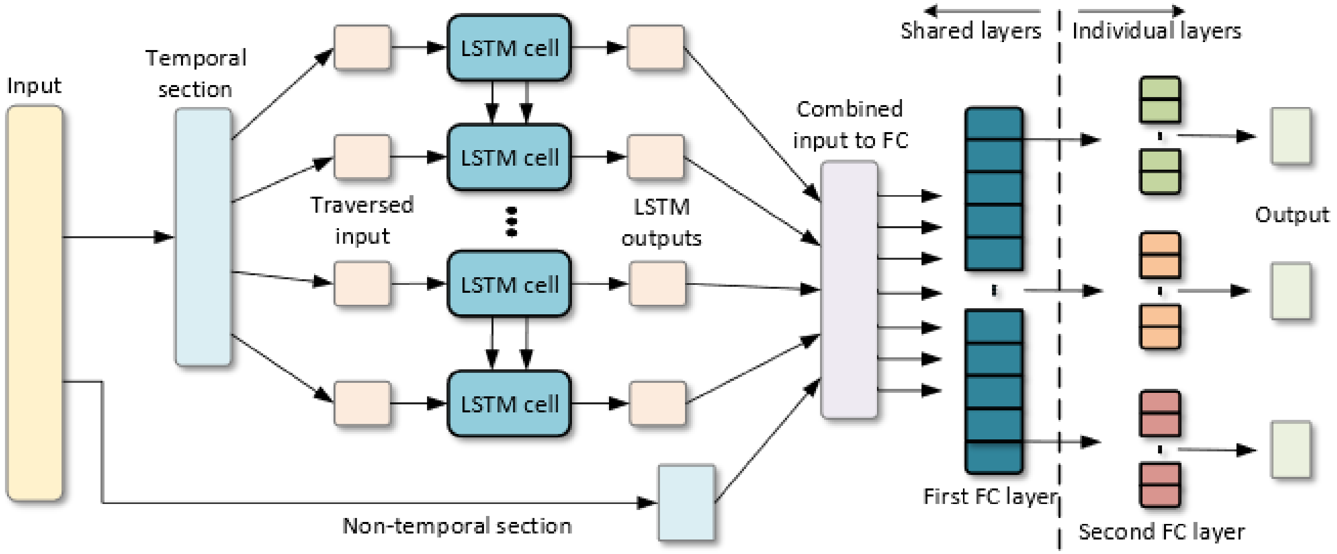
D_t: Number of new jobs deployed to the cluster after t_s, till moment t.

R_j: The overall average execution delay ratio of job j.

$m_{ih}, m_{ic}, \psi_{ih}$ and ψ_{ic}: penalty terms w.r.t. Improper Heterogeneity and Initial Competition.

DRL based Approach

- DRL model structure and value definition decomposition



$$v_1 = -\frac{\eta_c \cdot m_{ih} \cdot m_{ic}}{\eta_j}\left(M_j + \sum_{t=t_s}^{t_e} \beta^{D_t} W_j^{(t)}\right)$$

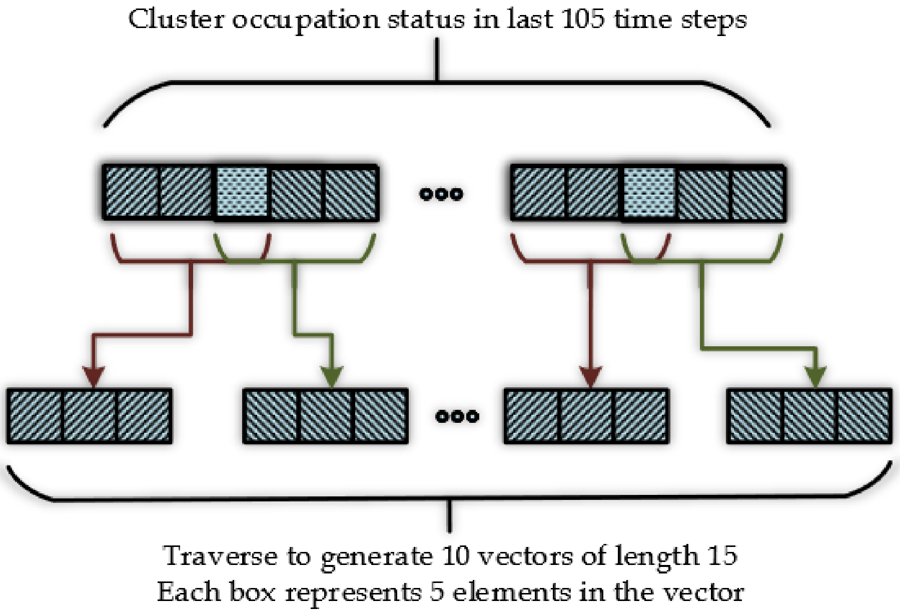
$$v_2 = -\frac{\eta_c \cdot m_{ih} \cdot m_{ic}}{\eta_j}\left(\sum_{t=t_s}^{t_e} \beta^{D_t} W_{cl}^{(t)}\right)$$

$$v_3 = -\psi_{ih} \cdot \psi_{ic} \cdot R_j$$

DRL based Approach

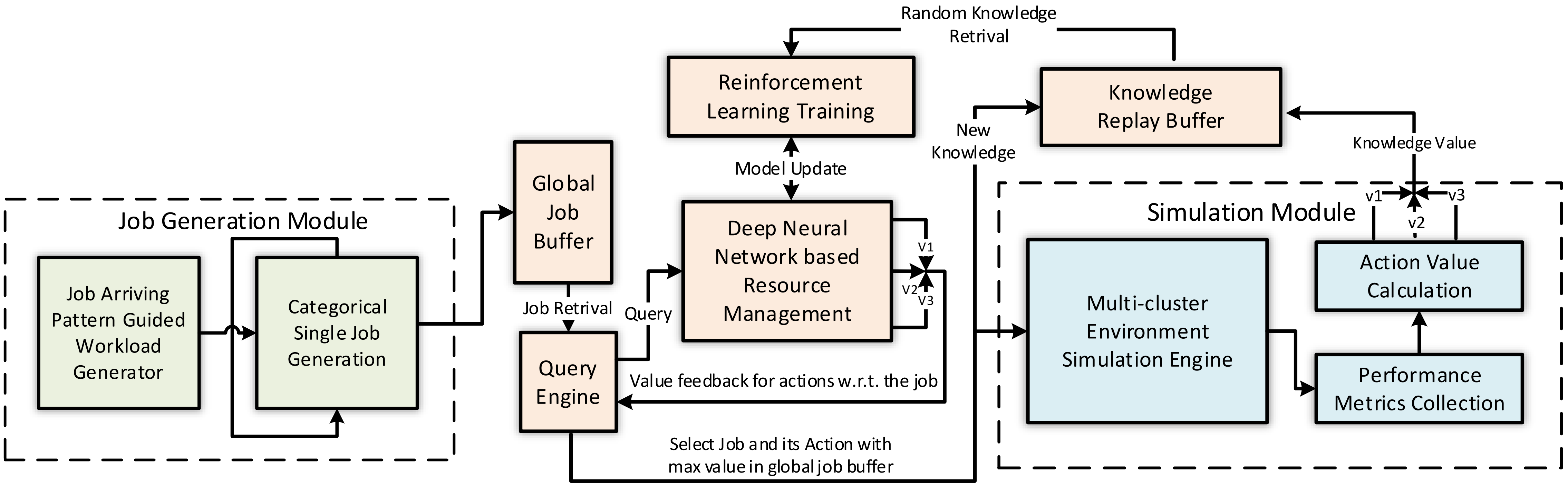
- Training Enhancement Skills
  - Cluster occupation status traverse.
    - Towards better cooperation with LSTM.
  - Training with decayed learning rate.
    - Towards finer model adjustment at later episodes in training.
  - Training with randomized workload.
    - Towards more general knowledge from various workloads.
  - Modified ε-greedy exploration.
    - Towards utilizing knowledge of rule-based model to partially guide exploration.
  - Solving multi-job selection dilemma
    - Towards coping with jobs in the job buffer.

Cluster occupation status traverse:



DRL based Approach

- Training architecture



Experiments

- Introduction
  - Experiment via simulation with a testing environment of 5 clusters. Clusters in this environment are heterogeneous and 2 of the clusters have elasticity as well.
- Elasticity controller
- Local intra-cluster scheduler

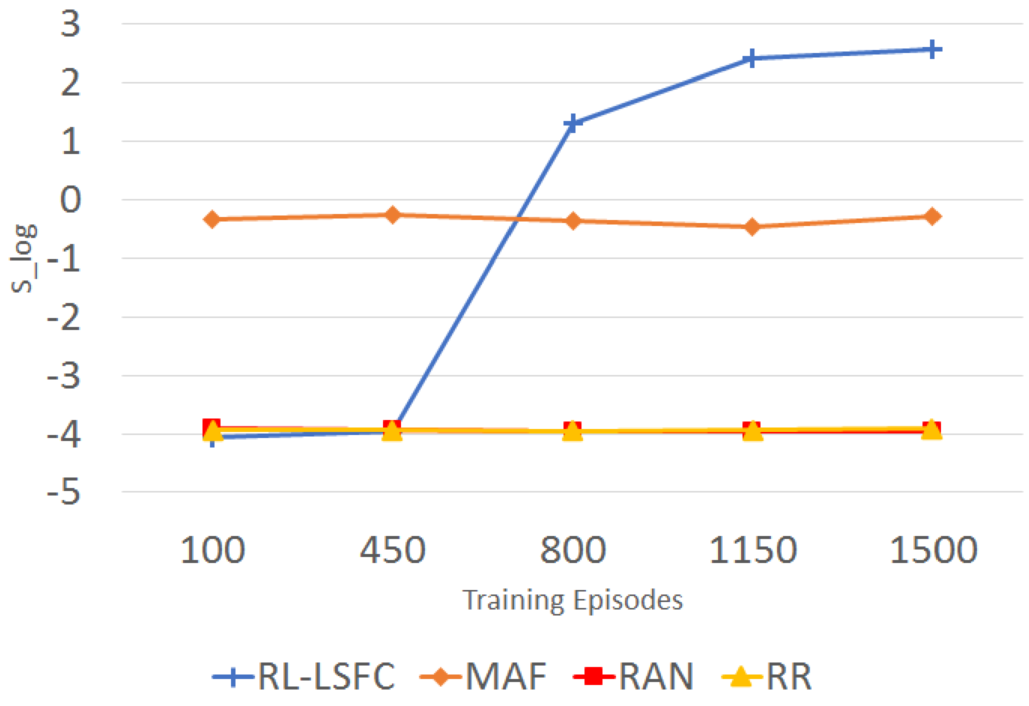
Experiments

- Comparison:
  - Rule-based baselines:
    - Random (RAN)
    - Round-Robin (RR)
    - Most Available First (MAF)
  - Another RL approach:
    - RL-FC
- Job arriving patterns:
  - Uniform, Bernoulli and Beta

- Performance metrics:
  - TMDL:
    - Total number of occurrences of missing deadlines for all jobs in all clusters during the execution of the workload.
  - AJER:
    - Average job execution time ratio among all clusters
  - S_log

$$S_{log} = sign(S) * \log_{10}(\max(|S|, 1)) \text{ as } S = -\mathrm{TMDL} + 50 * (100 - \mathrm{AJER})$$

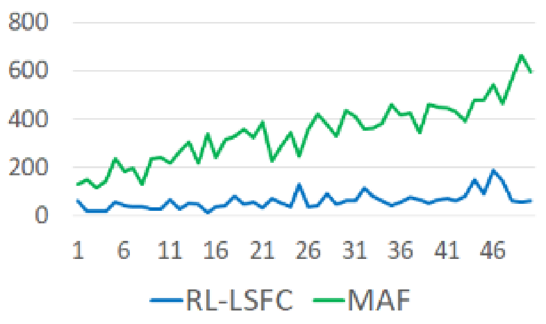
Experiments



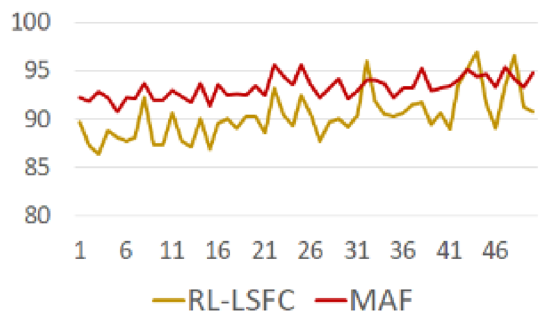
Performance comparison (S_{log}) of our deep RL approach RL-LSFC and baseline approaches in different training episodes.

Experiments

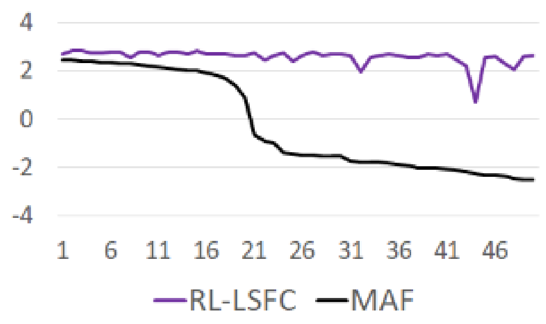
Comparison of RL-LSFC and MAF for 50 testing episodes. (L) lower is better. (H) higher is better. Fully-dominant(F), Semi-dominant(S) or Non-dominant(N) receives score 1 in an episode, if our approach is better than MAF in both, only one or none of the two metrics (TMDL and AJER).



(a) TMDL (L)



(b) AJER (L)

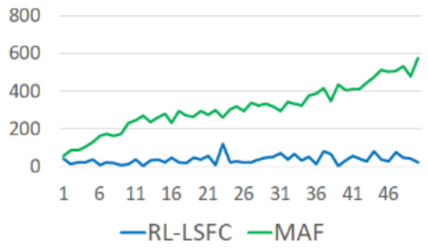


(c) S_{log} (H)

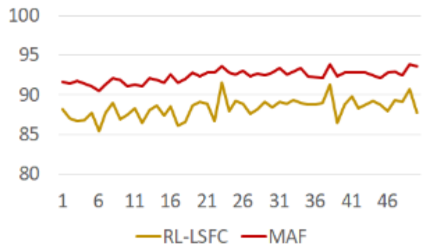
	TMDL (L)	AJER (L)	S_{log} (H)	F/S/N
RL-LSFC	**61.70**	**90.30**	**2.57**	46/4/0
MAF	343.84	93.22	−0.28	-

Experiments

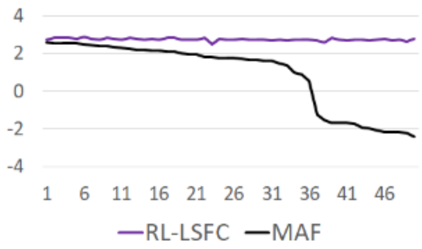
Comparison of RL-LSFC and MAF in variant workloads. (a)-(c) are related to b=36 scenario. (d)-(f) are related to b=40. Here b is a parameter in Uniform job pattern.



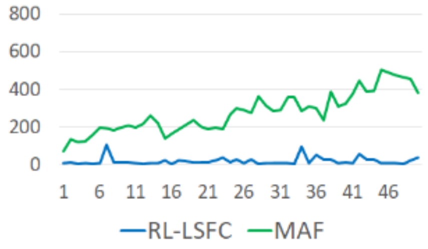
(a) TMDL (L)



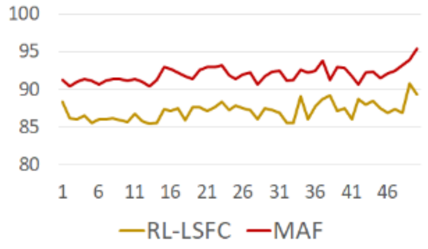
(b) AJER (L)



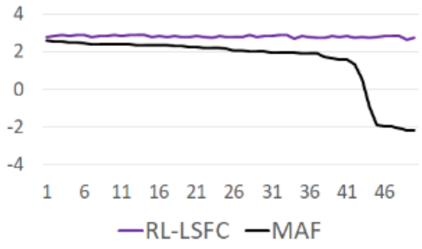
(c) S_{log} (H)



(d) TMDL (L)



(e) AJER (L)



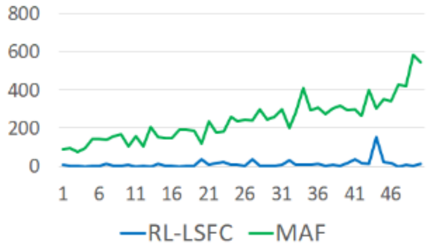
(f) S_{log} (H)

(a)-(c)	TMDL (L)	AJER (L)	S_{log} (H)	F/S/N
RL-LSFC	**37.66**	**88.32**	**2.73**	50/0/0
MAF	311.44	92.35	0.87	-

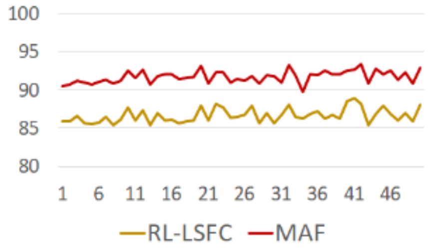
(d)-(f)	TMDL (L)	AJER (L)	S_{log} (H)	F/S/N
RL-LSFC	**19.76**	**87.12**	**2.79**	50/0/0
MAF	276.1	91.95	1.55	-

Experiments

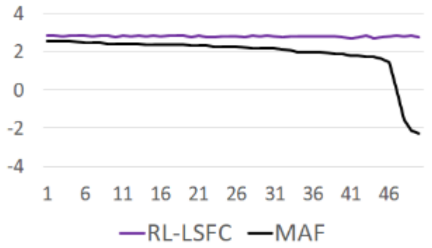
Comparison of RL-LSFC and MAF in other job arriving patterns. (a)-(c): Bernoulli pattern. (d)-(f): Beta pattern.



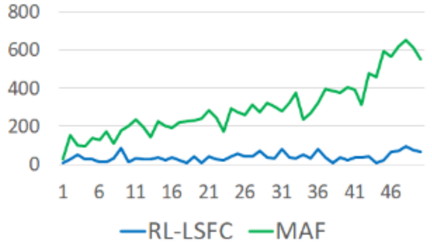
(a) TMDL (L)



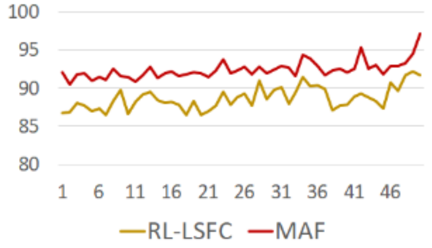
(b) AJER (L)



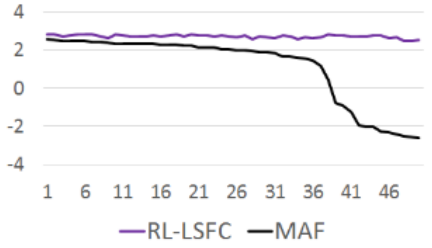
(c) S_{log} (H)



(d) TMDL (L)



(e) AJER (L)



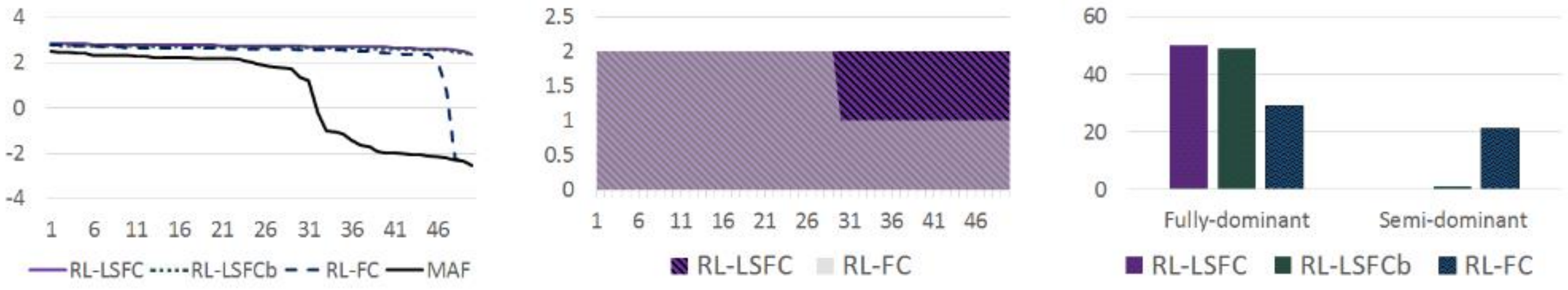
(f) S_{log} (H)

(a)-(c)	TMDL (L)	AJER (L)	S_{log} (H)	F/S/N
RL-LSFC	**13.32**	**86.67**	**2.81**	50/0/0
MAF	243.18	91.75	1.92	-

(d)-(f)	TMDL (L)	AJER (L)	S_{log} (H)	F/S/N
RL-LSFC	**39.14**	**88.68**	**2.71**	50/0/0
MAF	295.66	92.41	1.11	-

Experiments

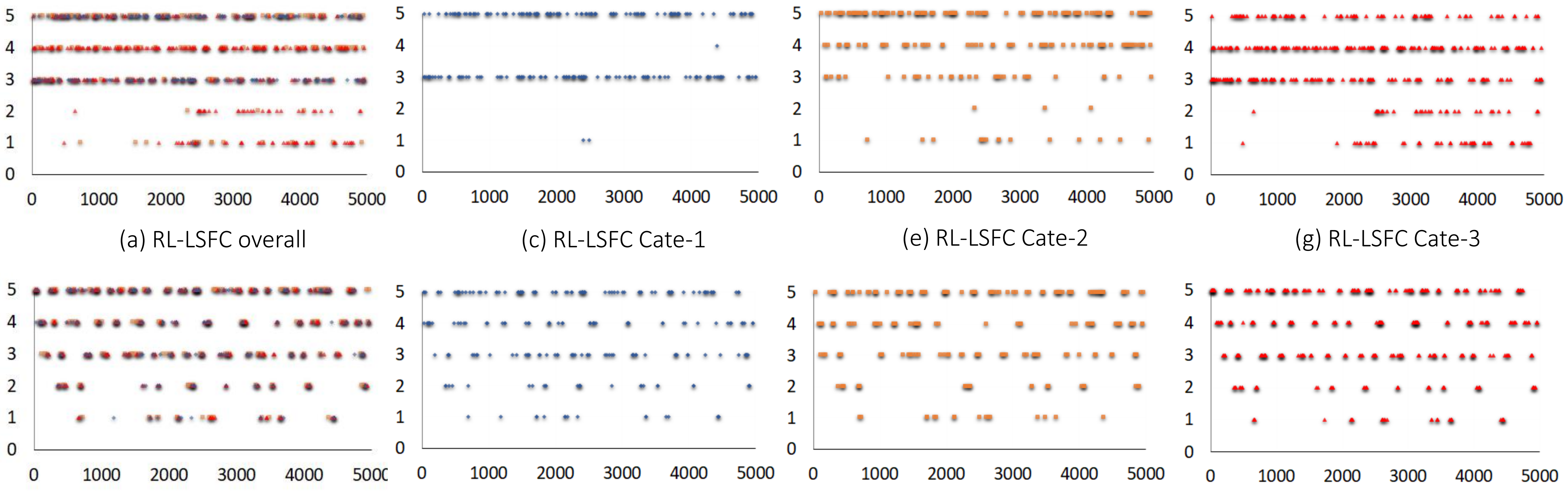
Comparison of three RL models w.r.t. MAF. In (b), we give F:2, S:1 and N:0 for scoring to show a dominant area (larger is better) of RL-LSFC (RL-LSFCb is very similar to RL-LSFC here, so omitted for viewing) and RL-FC.



(a) S_{log} (H) (b) Dominant area (c) F/S/N distribution

	TMDL (L)	AJER (L)	S_{log} (H)	F/S/N
RL-LSFC*	36.84	**88.71**	**2.71**	**50/0/0**
RL-LSFCb*	**14.74**	90.13	2.67	49/1/0
RL-FC	15.28	92.79	2.24	29/21/0
MAF	305.02	92.67	0.65	-

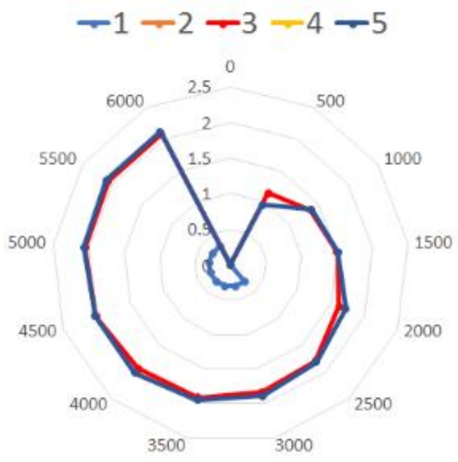
Experiments



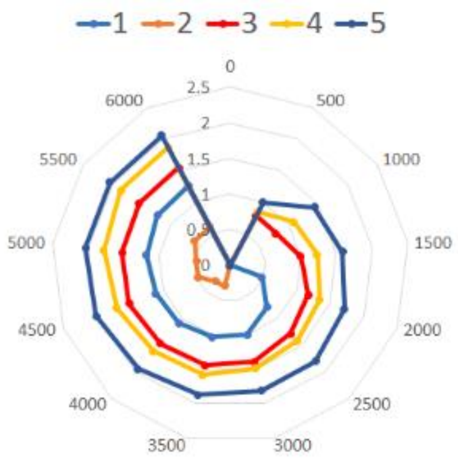
(b) MAF overall (d) MAF Cate-1 (f) MAF Cate-2 (h) MAF Cate-3

Job-Cluster scheduling patterns for RL-LSFC and MAF in one testing episode. One point for each job and one color for each job category. Vertical axis 1-5 is referring to cluster sequence number. Horizontal axis is time slice.

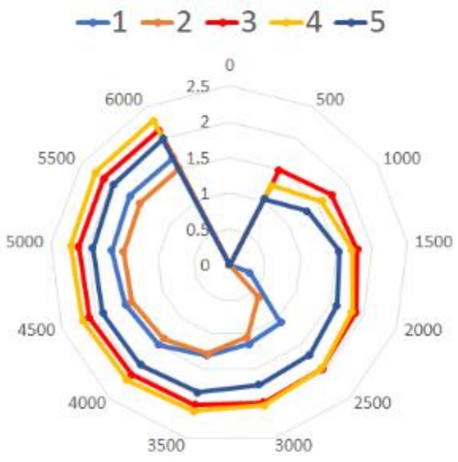
Experiments



RL-LSFC Cate-1



RL-LSFC Cate-2



RL-LSFC Cate-3

Comparison of Job-Cluster scheduling pattern with respect to different job categories under RL-LSFC control. Value axis is on logarithmic scale of job counts; angle axis is time slice. One color for each cluster.

Conclusion

- Obtained an elasticity-compatible resource management via DRL for a heterogeneous multi-cluster environment.
- Comparing to the best baseline, it
  - reduces the occurrence of missing execution deadline events for workloads of 1000 jobs by around 5x to 18x,
  - and reduces average execution time ratio by around 2% to 5%.
- Also shows better performance than a previous reinforcement learning based approach with fully-connected layers.