Developing Dynamic Resource Management System in SoC Emulation

Seonchang Choi, Sangwoo Noh,
Seonghee Yim, Seonil Brian Choi

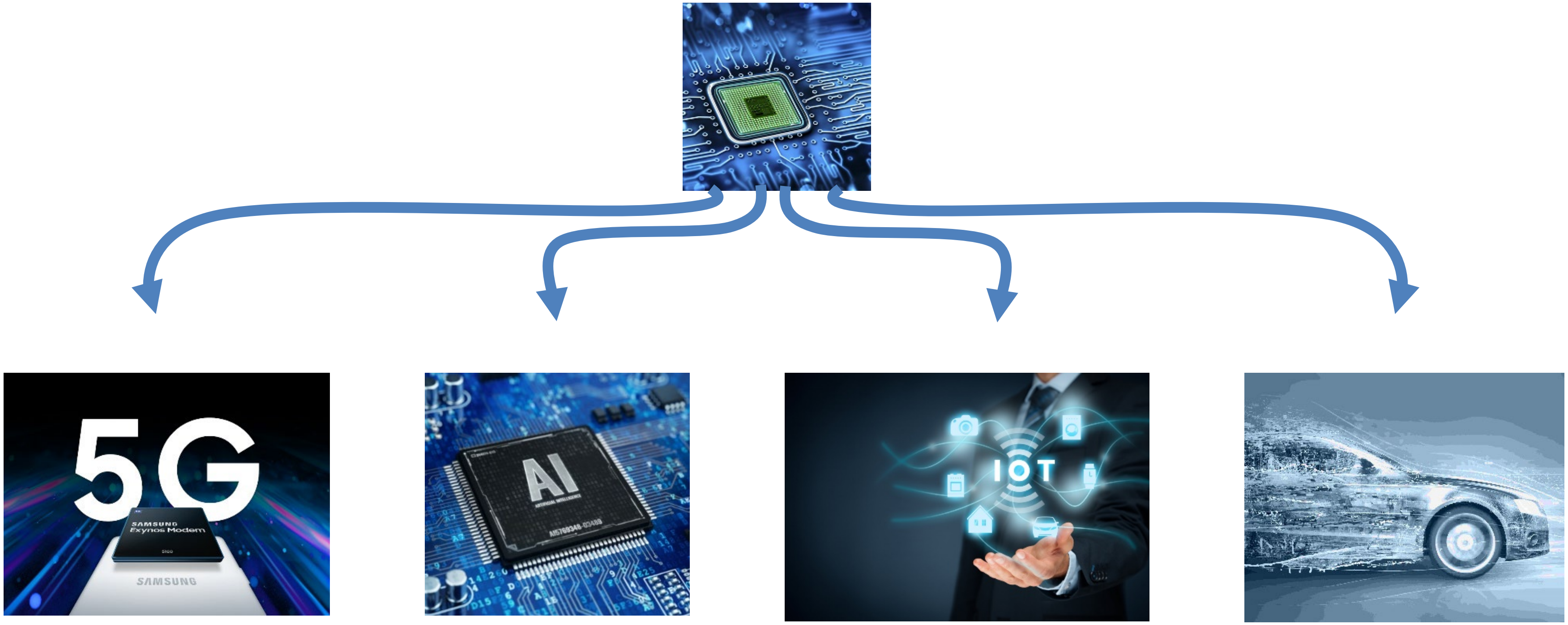
SAMSUNG ELECTRONICS

Index

- Backgrounds
- Emulator Queueing System
- Dynamic Resource Management
- Result
- Conclusions

Backgrounds

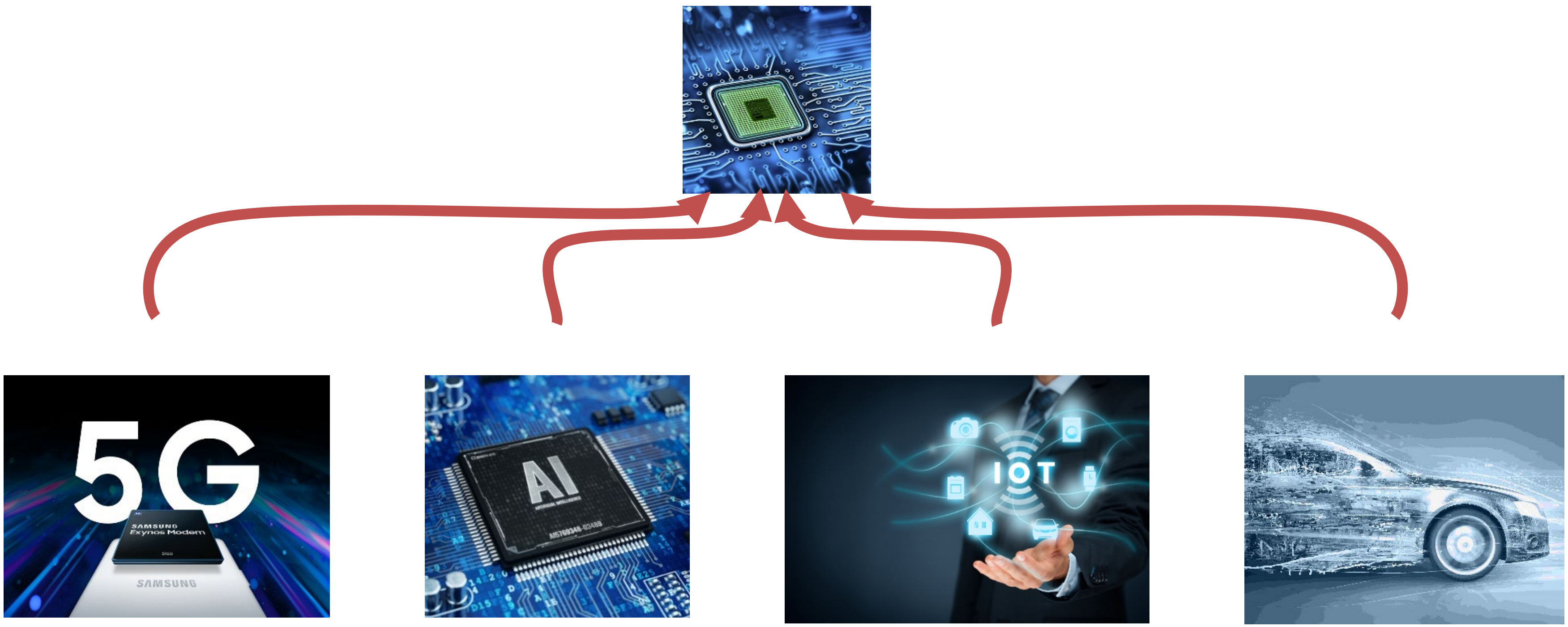
- Trends – SoCs are in EVERYWHERE!



5G network | AI | IoT | Automotive

Backgrounds

- Trends – SoCs must do EVERYTHING!



Modem | NPU / TPU | ISP / DSP | ADAS

Backgrounds

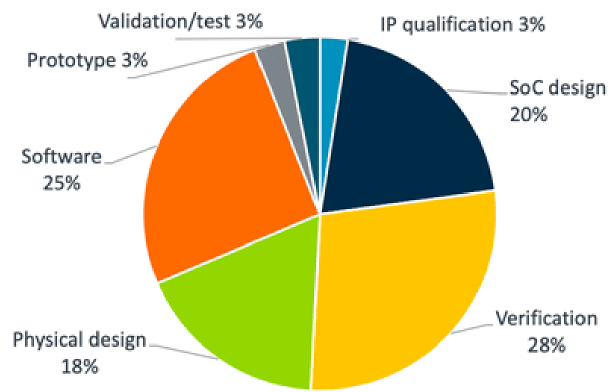
- Trends – Verification cost takes large portion of a pie!

SoC development cost breakdown

Verification
Verification IP
Models
Emulation/FPGA flows
Arm approved design Services
Support & training

Software
Debug tools
Compilers
Firmware
Apps
OS
Open source contributions
Models
FPGA/Emulation support
Software development boards

SoC development cycle cost ($%)



Physical design
Reference methodologies
Physical IP and POP
DFT/DFM readiness and flows
Arm approved design services

Silicon IC prototyping
Test methodologies
Models
Debug tools
Evaluation boards
Ready to use firmware or OS

IP qualification
Silicon proven IP
Functional integration kits
Reference platforms
Models for simulation
Models for software acceleration
Support

Architecture/design
Integration kits
Reference platforms
Models
Bus & interface standards
Arm system IP
Functional safety support
Security – IP, software and platform
Emulation

Source: ARM

Backgrounds

- Problems
  - Different interfaces for different types of emulators

Interface A	Interface B	Interface C

Company A

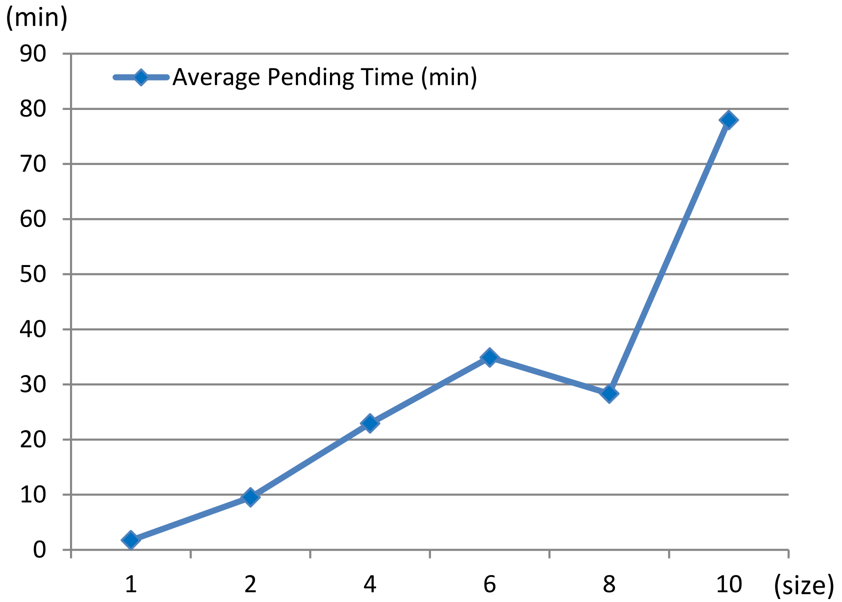
Company B

Company C

Backgrounds

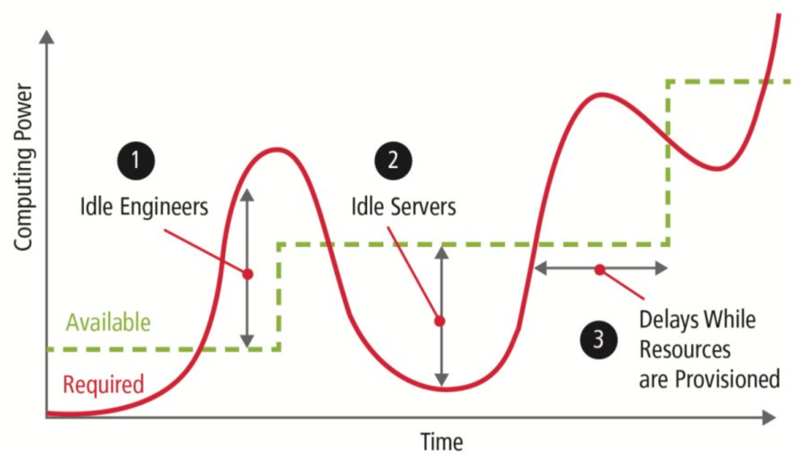
- Problems
  - Inefficient resource sharing

Starvation



Number of IP block per SoC

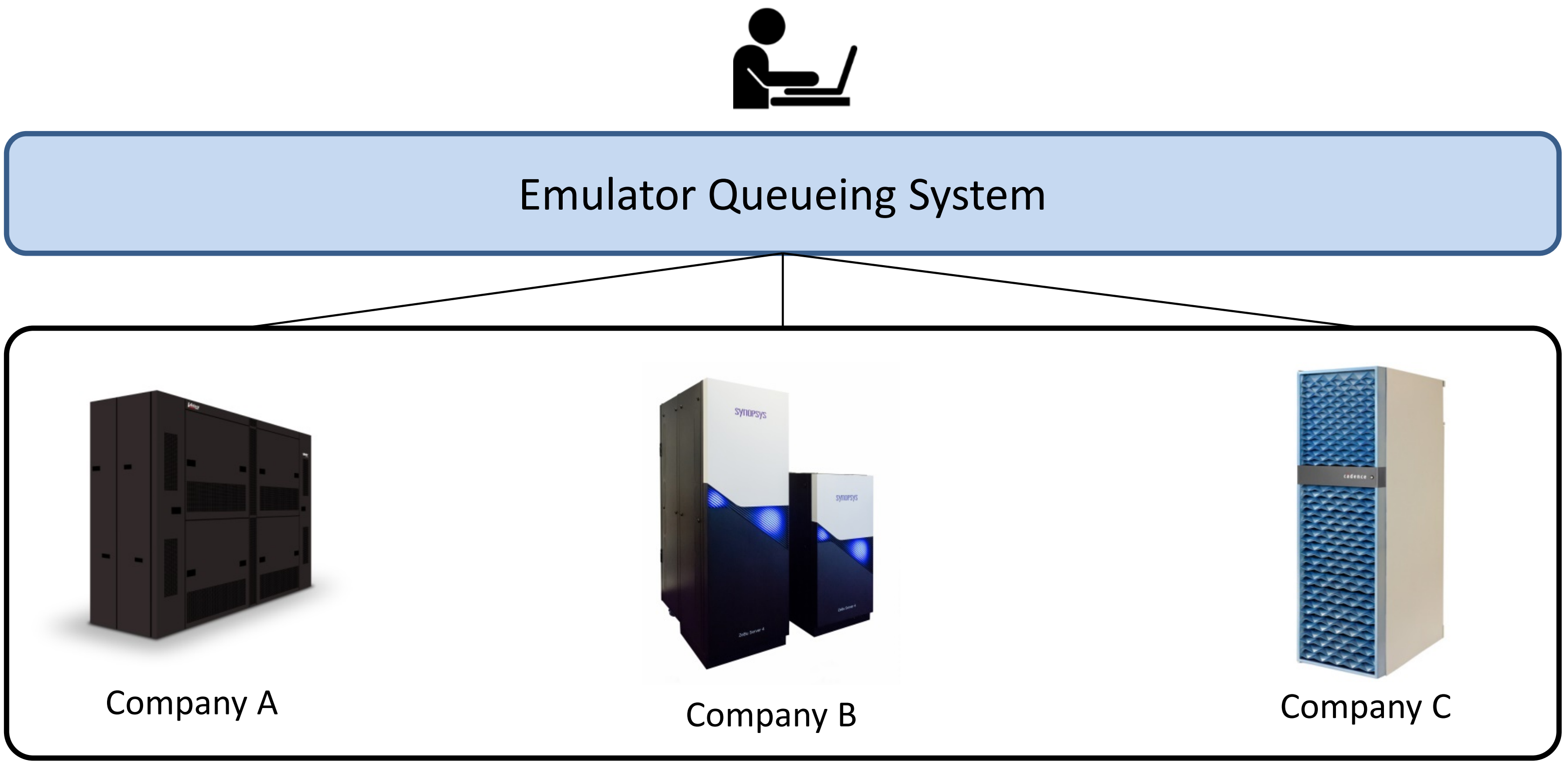
Low Utilization



Available resource and usage rate

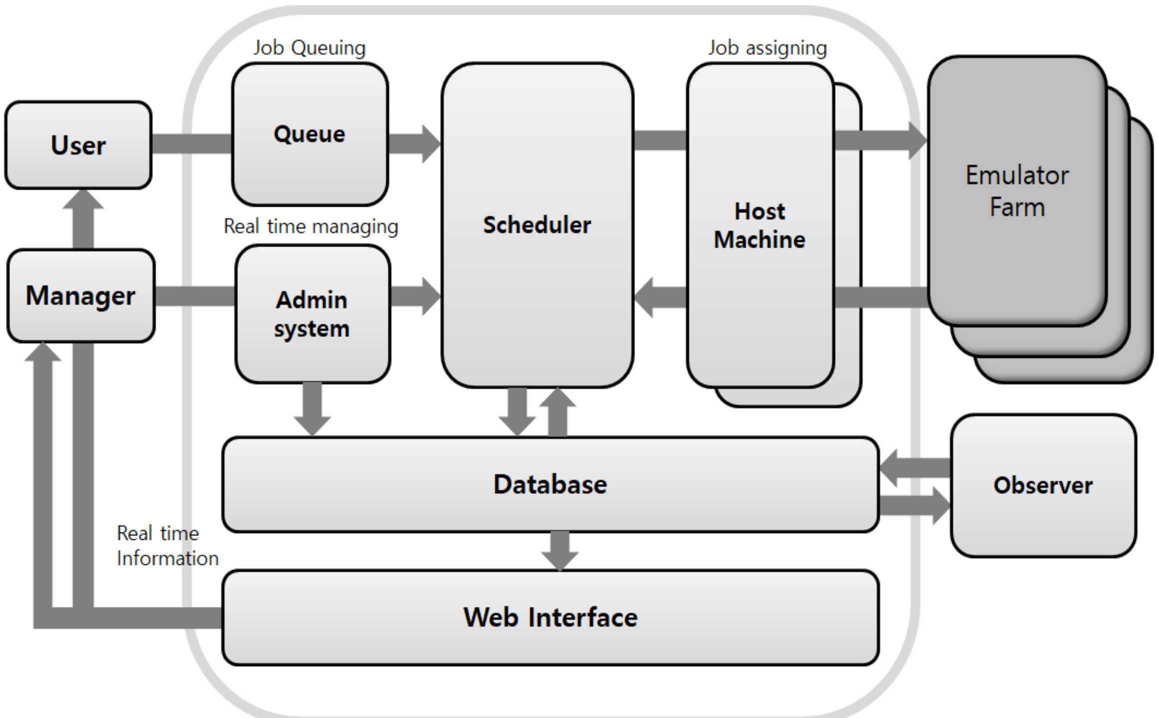
Emulator Queueing System

- Problems
  - Single interface for different types of emulators



Emulator Queueing System

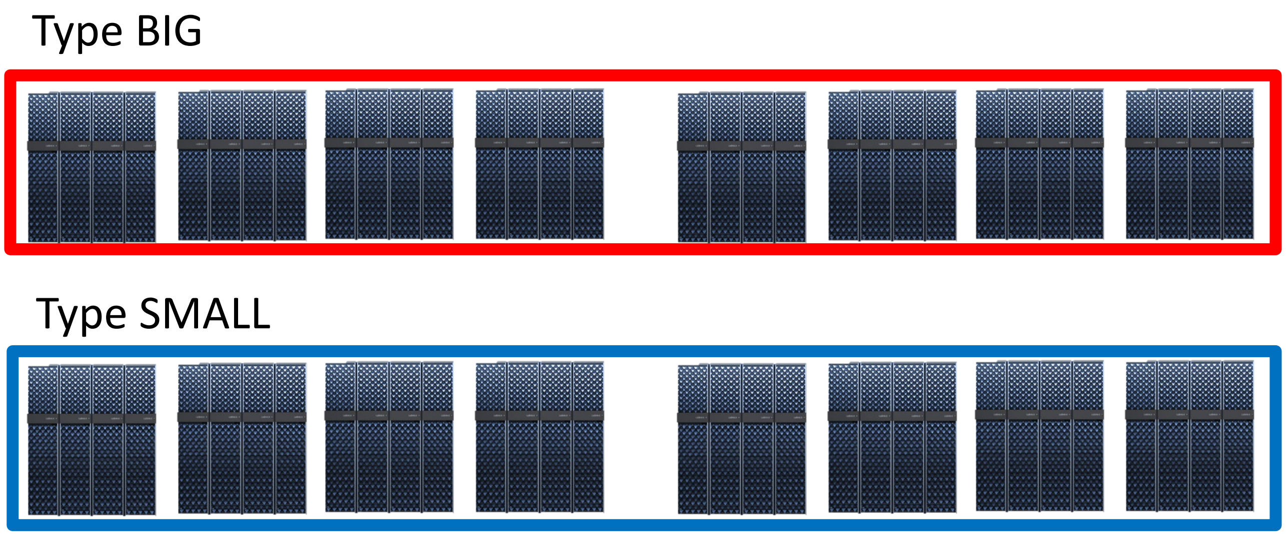
- Components
  - Queue
  - Scheduler
  - Host Machines
  - Observer
  - User Interface



System Architecture

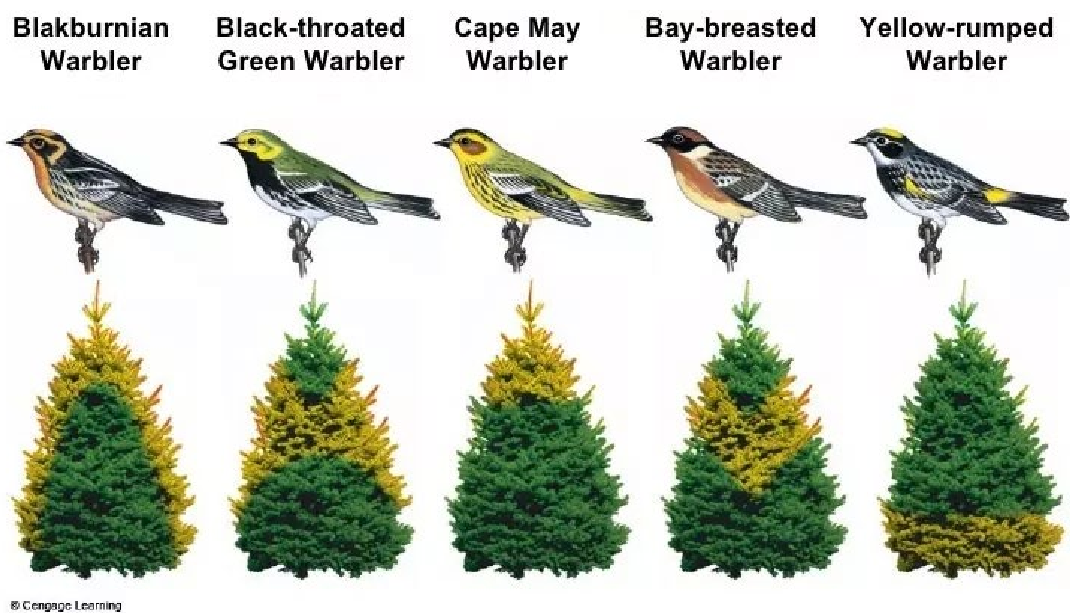
Emulator Queueing System

- Resource Partitioning
  - *N* Partitions with *N* type in an emulation farm
  - Dedicated type for each partition



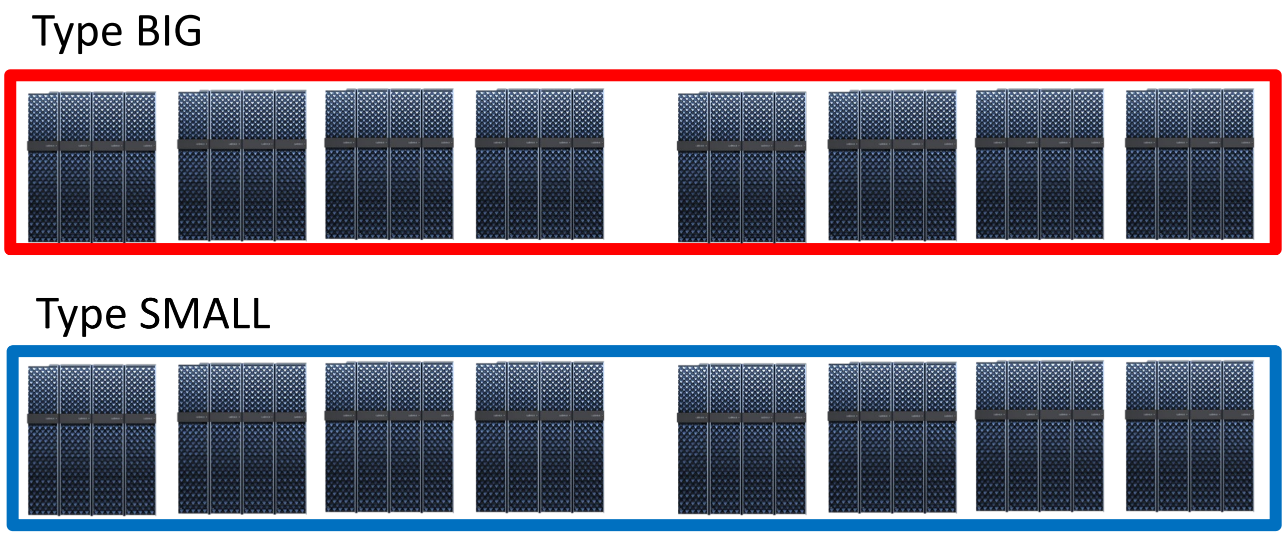
Emulator Queueing System

- Resource Partitioning
  - Solution for starvation in warblers' ecosystem
  - ***"Resource partitioning acts to promote the long-term coexistence of competing species."***



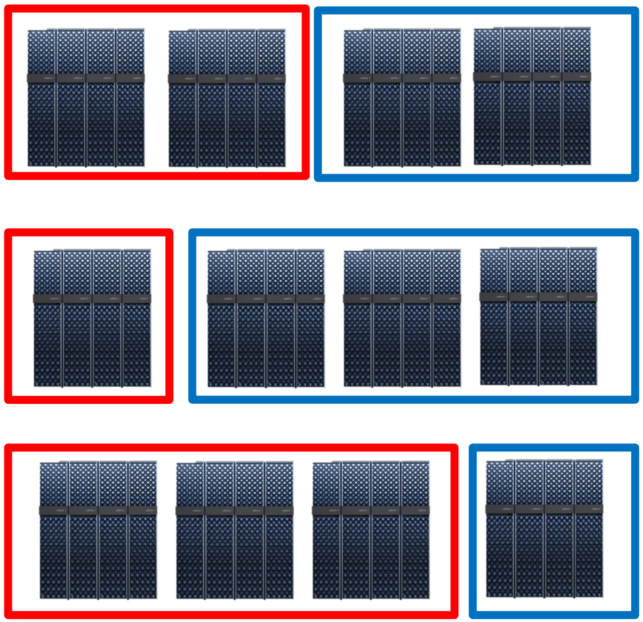
Emulator Queueing System

- Resource Partitioning
  - *N* Partitions with *N* types in an emulation farm
  - Dedicated type for each partition



Emulator Queueing System

- Dynamic Resource Management
  - Dynamically configure the size of partitions



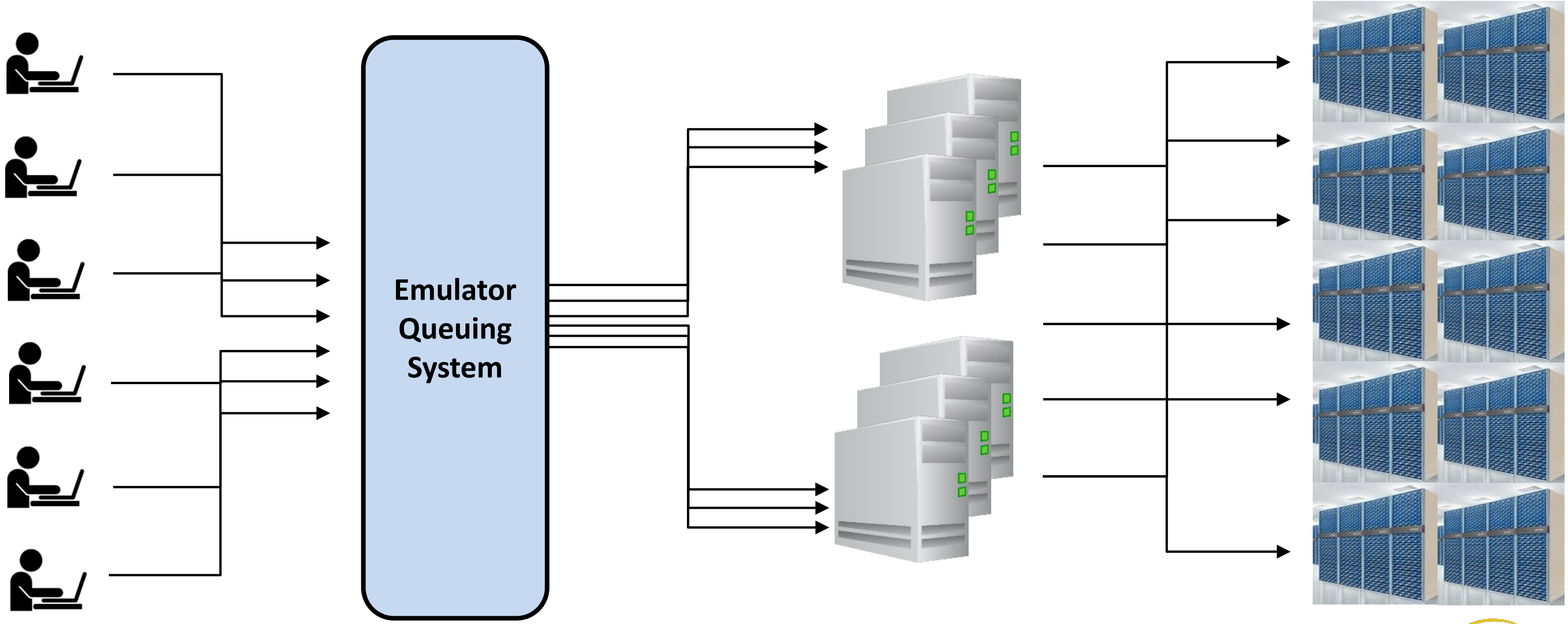
When **type A and B requires similar resources** resources

When **type B requires more** resources than Type A

When **type A requires more** resources than Type B

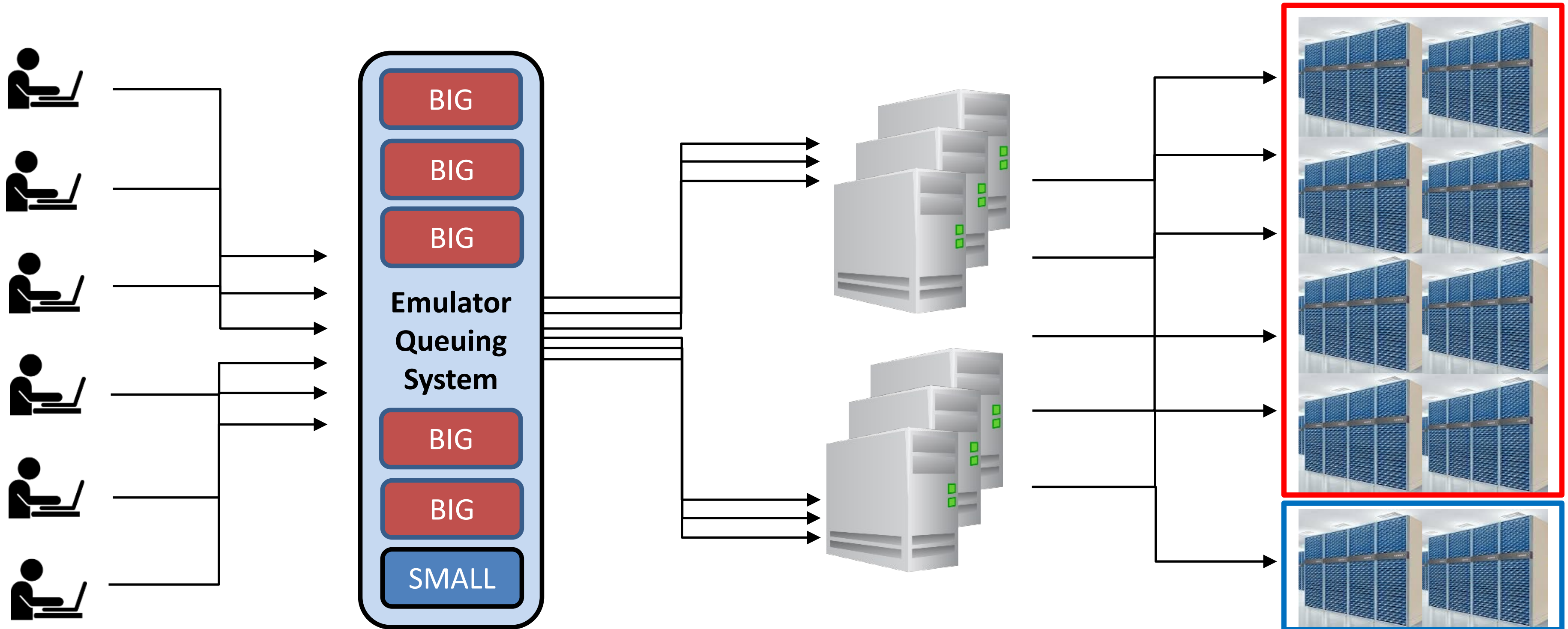
Emulator Queueing System

- Advantages
  - Easy to scale-out



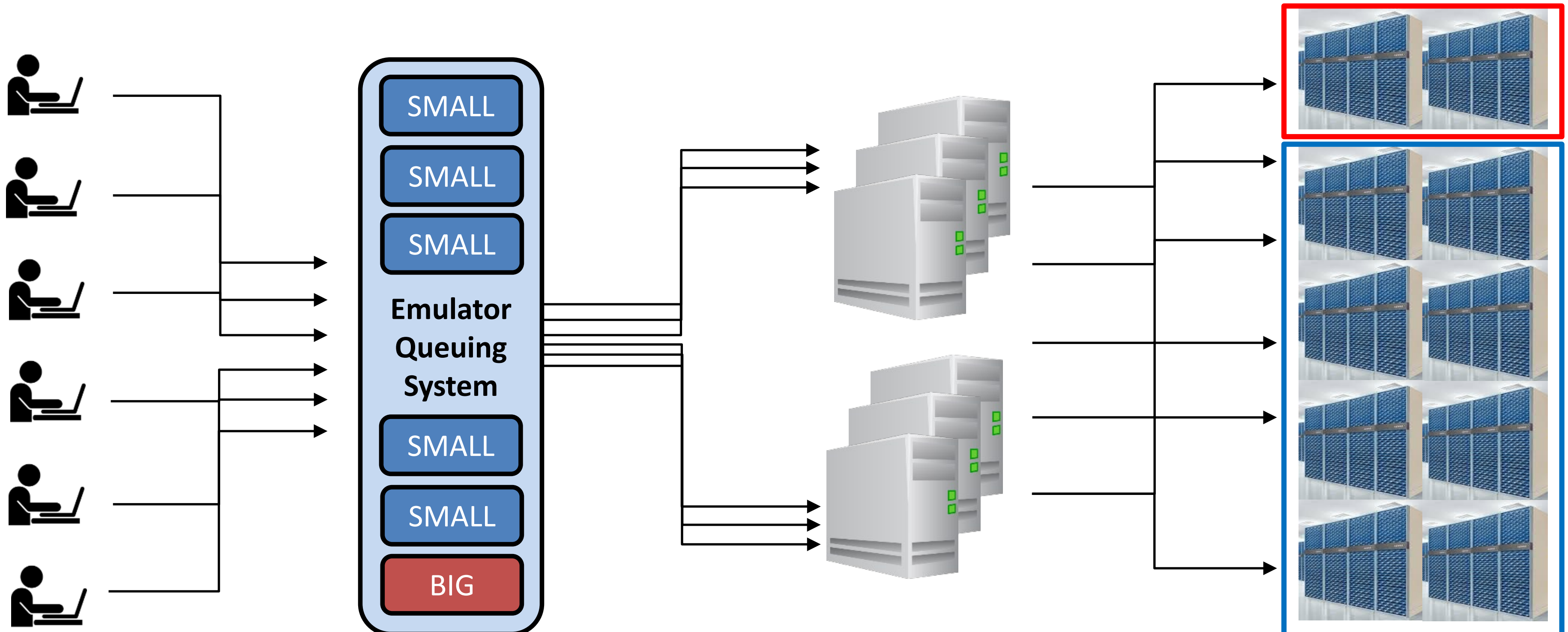
Emulator Queueing System

- Advantages
  - Increase resource utilization



Emulator Queueing System

- Advantages
  - Increase resource utilization



Dynamic Resource Management

- What policy should be applied to make a decision?

Dynamic Resource Management

- What policy should be applied to make a decision?

1. **Machine learning based policy**
   - Reinforcement learning – Deep Q Network
2. **Heuristic based policies**
   - Quality of Service (QoS)
   - Greedy
   - Fair share

Dynamic Resource Management: Machine Learning Policy

- Why do we use reinforcement learning?

Dynamic Resource Management: Machine Learning Policy

- Why do we use reinforcement learning?

✓ **Markov Decision Process**

- *Markov decision process (MDP) is a* ***discrete time*** *stochastic control process.*
- *Mathematical framework for* ***<u>modeling decision making</u>*** *in situations where outcomes are partly random and partly under the control of a decision maker.*
- *MDPs are useful for studying* ***optimization problems*** *solved via dynamic programming and* ***reinforcement learning.***

Bellman, R. (1957). "A Markovian Decision Process". Journal of Mathematics and Mechanics

Dynamic Resource Management: Machine Learning Policy

- How does the reinforcement learning find out the optimal solution?

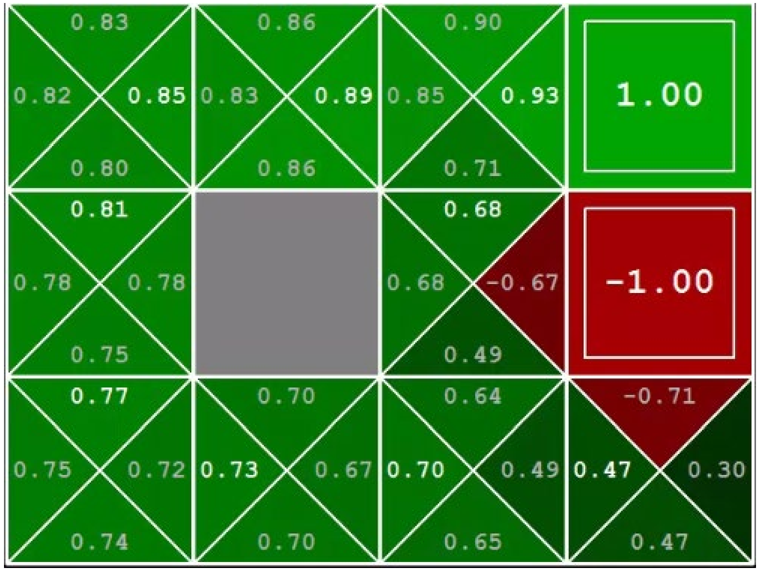
Dynamic Resource Management: Machine Learning Policy

- How does the reinforcement learning find out the optimal solution?

✓ **Bellman Equation**

$$V^{\pi*}(s) = \max_a \{R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^{\pi*}(s')\}$$

- V(s): value function
- s: state
- s': next state
- a: action
- R(s,a): reward function
- *r*: discounted rate
- P(s|s,a): conditional probability



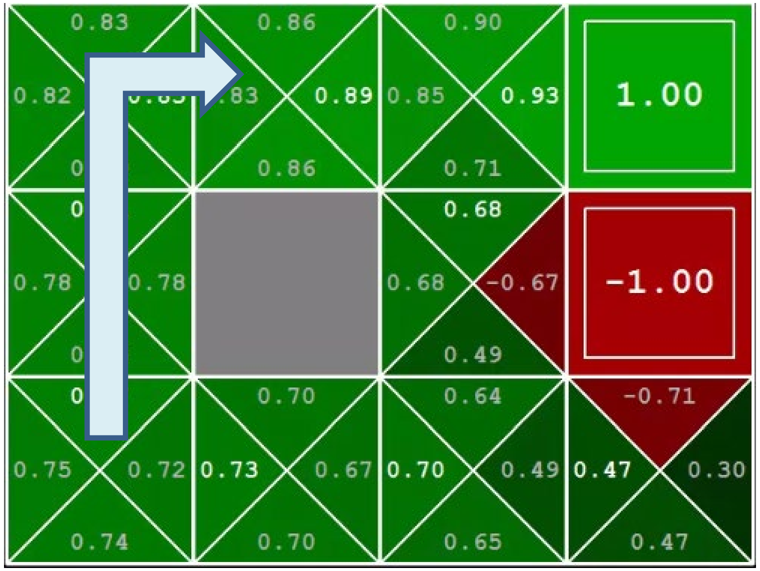
Dynamic Resource Management: Machine Learning Policy

- How does the reinforcement learning find out the optimal solution?

✓ **Bellman Equation**

$$\boxed{V^{\pi*}(s) = \max_{a}\{R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^{\pi*}(s')\}}$$

- V(s): value function
- s: state
- s’: next state
- a: action
- R(s,a): reward function
- *r*: discounted rate
- P(s|s,a): conditional probability



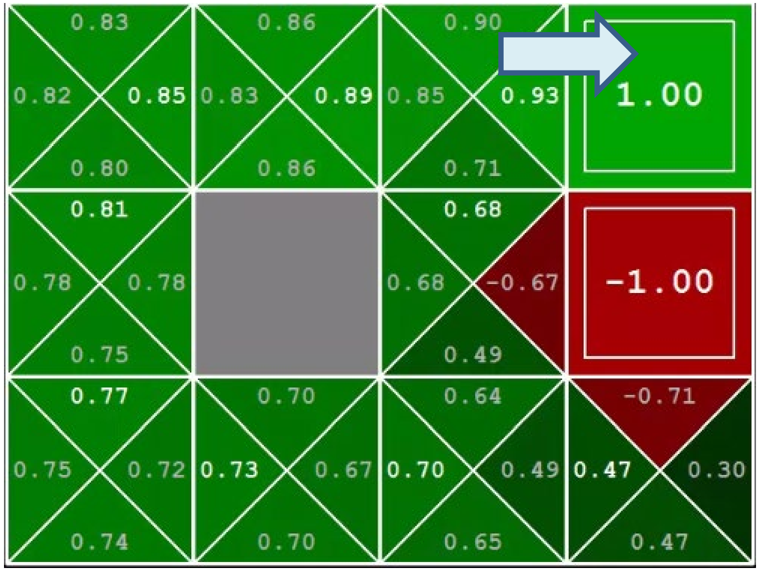
Dynamic Resource Management: Machine Learning Policy

- How does the reinforcement learning find out the optimal solution?

✓ **Bellman Equation**

$$V^{\pi*}(s) = \max_a \{R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^{\pi*}(s')\}$$

- V(s): value function
- s: state
- s': next state
- a: action
- R(s,a): reward function
- *r*: discounted rate
- P(s|s,a): conditional probability



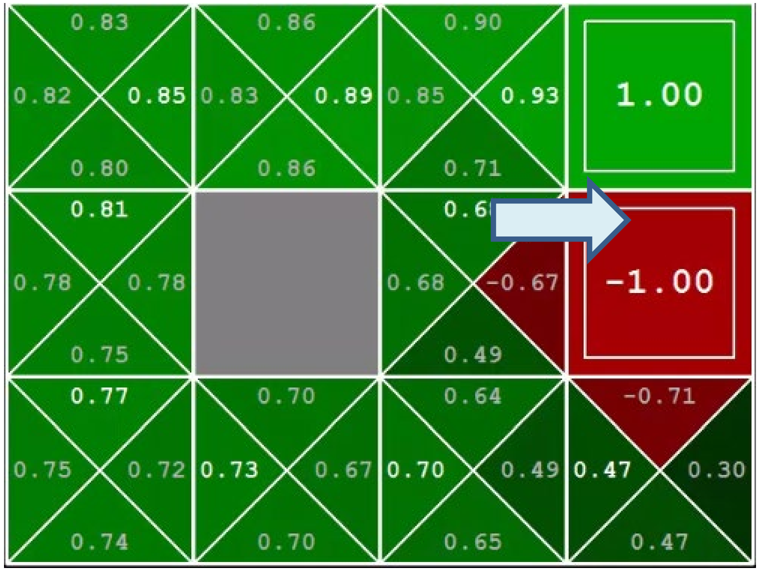
Dynamic Resource Management: Machine Learning Policy

- How does the reinforcement learning find out the optimal solution?

✓ **Bellman Equation**

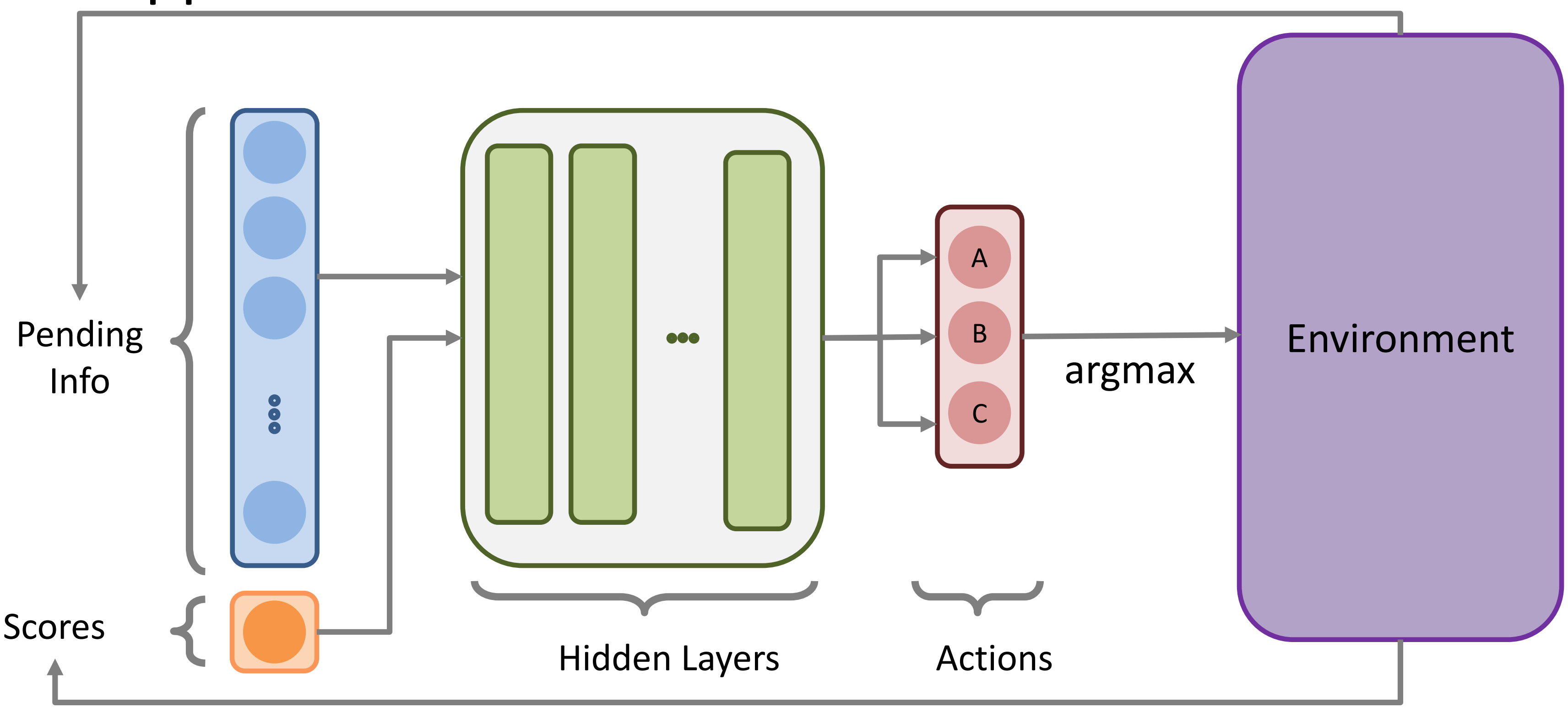
$$\boxed{V^{\pi*}(s) = \max_{a}\{R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^{\pi*}(s')\}}$$

- V(s): value function
- s: state
- s’: next state
- a: action
- R(s,a): reward function
- *r*: discounted rate
- P(s|s,a): conditional probability



Dynamic Resource Management: Machine Learning Policy

- Application

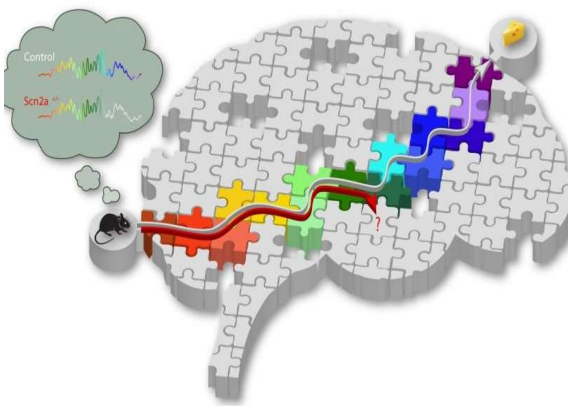


Dynamic Resource Management: Machine Learning Policy

- How did we improve learning speed?

✓ Stay action when there is no pending jobs

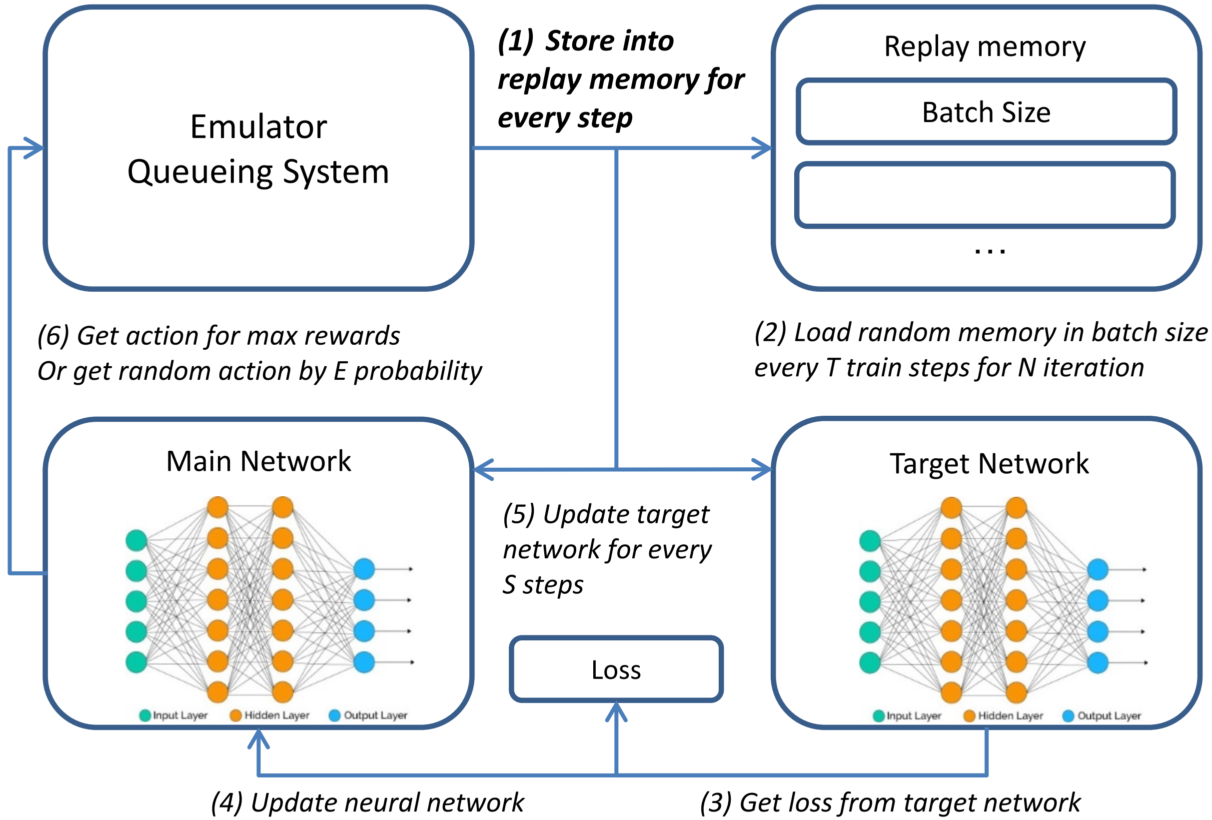
✓ Replay memory & Target network



** The concept is presented in the paper 'Play Atari with deep reinforcement learning' by Deepmind*

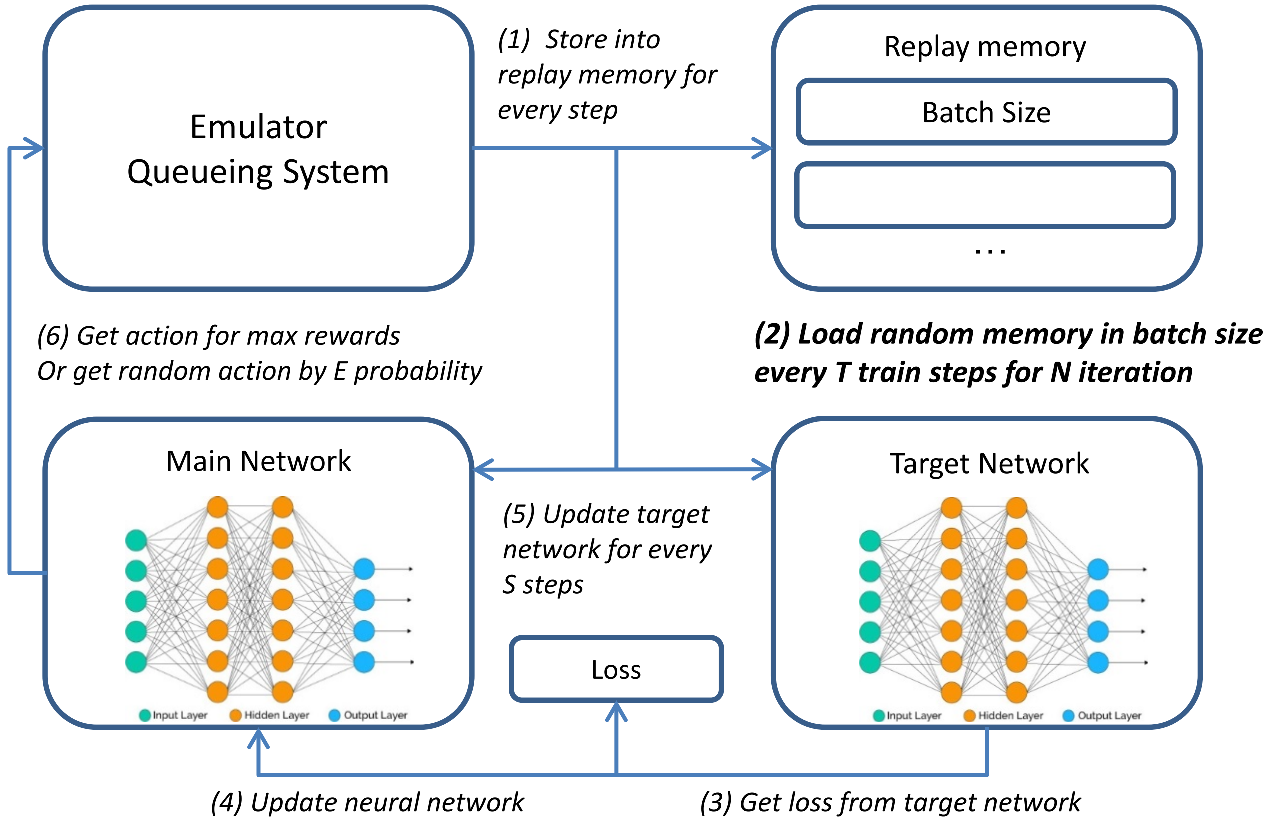
Dynamic Resource Management: Machine Learning Policy

- Architecture



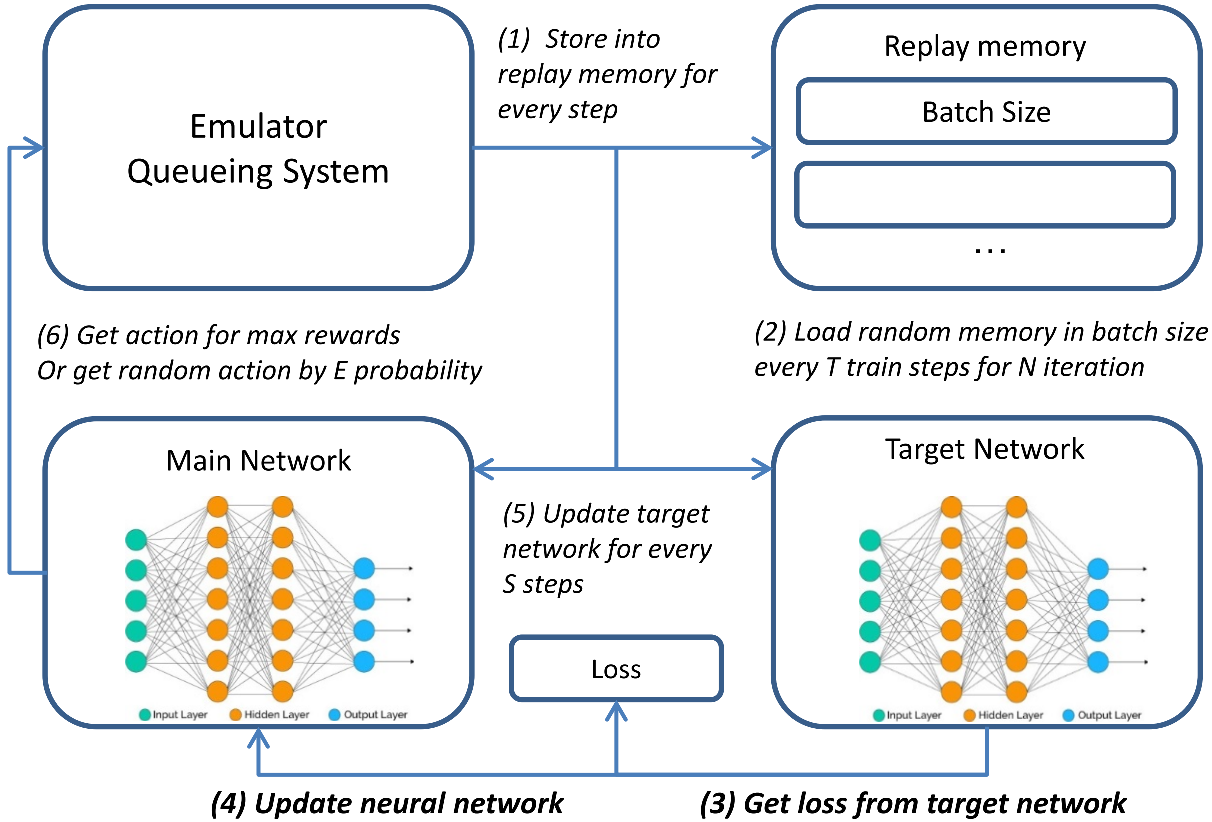
Dynamic Resource Management: Machine Learning Policy

- Architecture



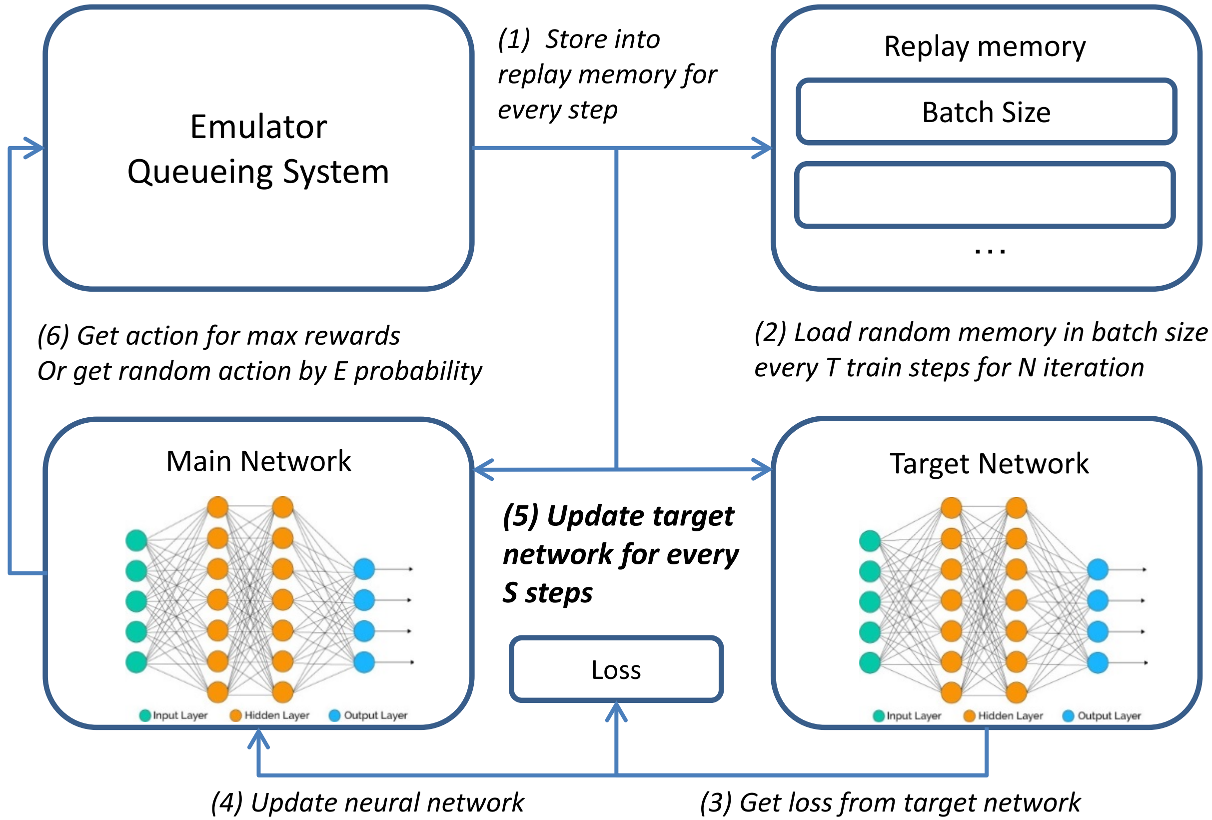
Dynamic Resource Management: Machine Learning Policy

- Architecture



Dynamic Resource Management: Machine Learning Policy

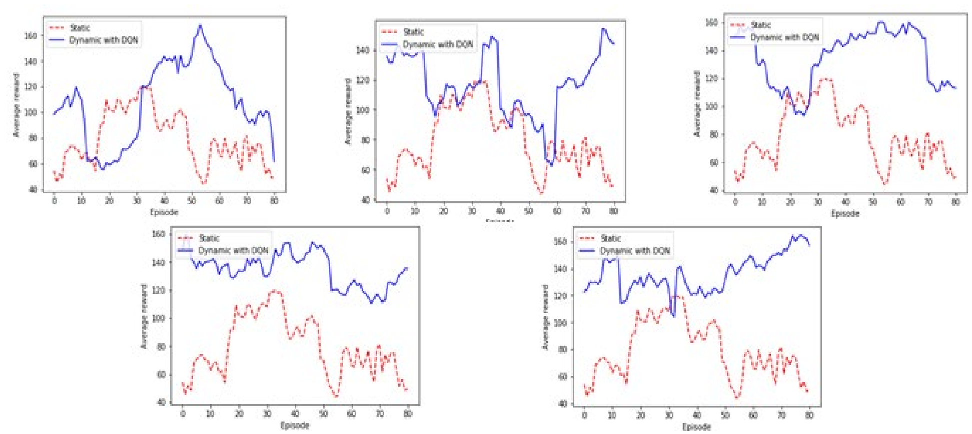
- Architecture



Dynamic Resource Management: Machine Learning Policy

- Number of trains : 10,000
  - 5,000 episodes, 10 iterations

(A) Average reward while training



(B) Average reward for 500 tests

Dynamic Resource Management: Heuristic Policies

Quality of Service **Greedy** **Fair Share**

Dynamic Resource Management: Heuristic Policies

- **QoS – Max pending time**

Algorithm 1 QoS policy

IF max(*big*-type job pending time) > max(*small*-type job pending time) **THEN**
    IF max(*big*-type job pending time) > BIG MAX **THEN**
        allocate to big
  END IF
ELSE IF max(*small*-type job pending time) > SMALL MAX **THEN**
    IF max(*small*-type job) > BIG MAX **THEN**
        allocate to small
  END IF

END IF
** BIG MAX and SMALL MAX is a constant number.*

** Allocation scheme : Max-machine, max-unit*

*** Decision interval : 5 min*

Dynamic Resource Management: Heuristic Policies

- **Greedy → Current queue status**

Algorithm 2 Greedy policy

```
IF big/total ratio > RATIO THEN
    IF big total pending time > BIG TOTAL MAX THEN
        allocate to big
  END IF
ELSE IF small/total ratio > RATIO THEN
    IF small total time > SMALL TOTAL MAX THEN
        allocate to small
  END IF
END IF
```

** BIG RATIO, SMALL RATIO, BIG TOTAL MAX and SMALL TOTAL MAX are constant numbers.*

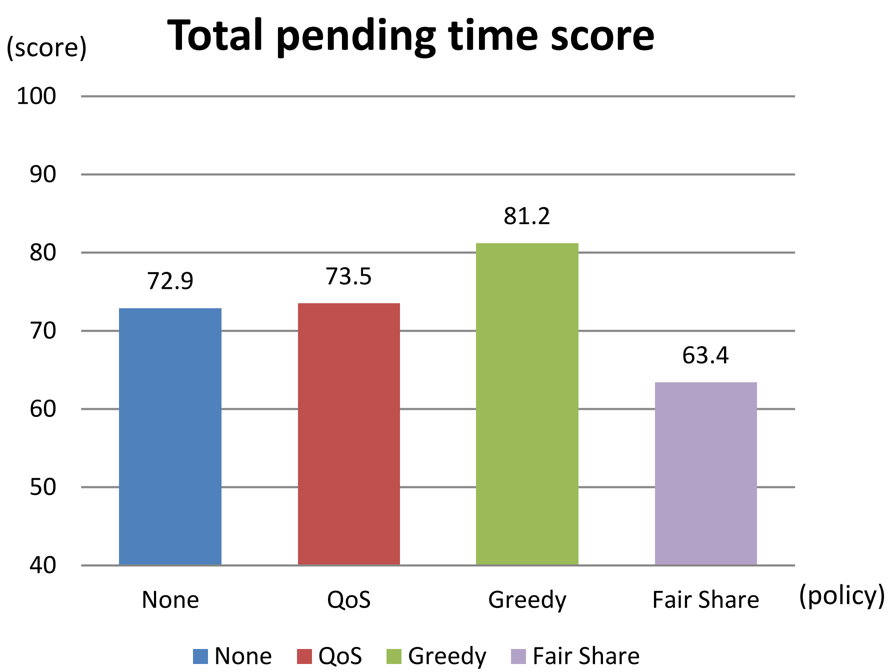
- **Fair Share → Average pending time**

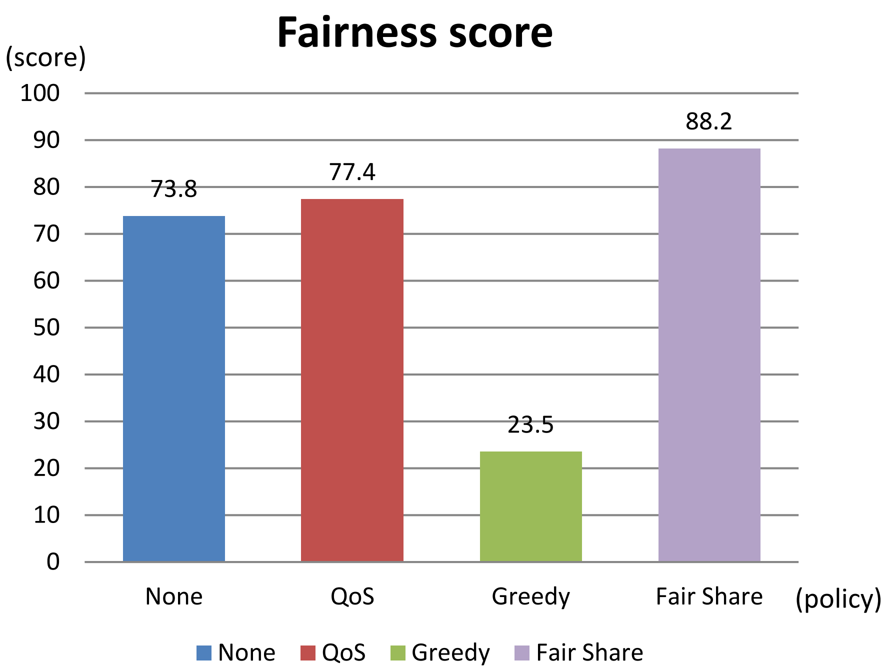
Algorithm 3 Fair share policy

```
IF big-type jobs' pending avg > small-type jobs' pending avg   THEN
    allocate to big
ELSE
    allocate to small
END IF
```

Dynamic Resource Management: Heuristic Policies

- Experimental Result



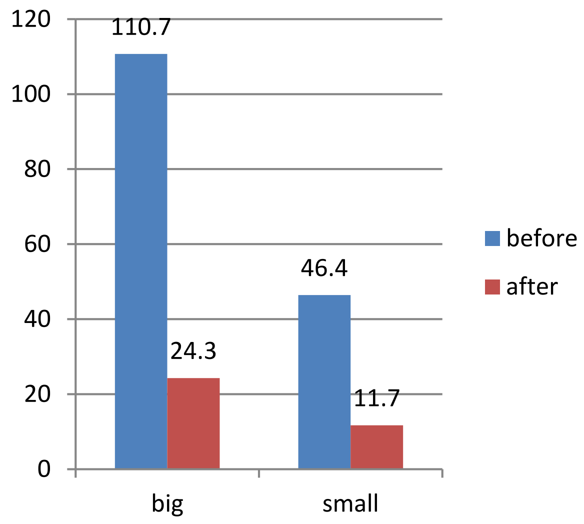


** Total pending time score = normalized value of total pending time*
*** Fairness score = normalized value of total pending time*

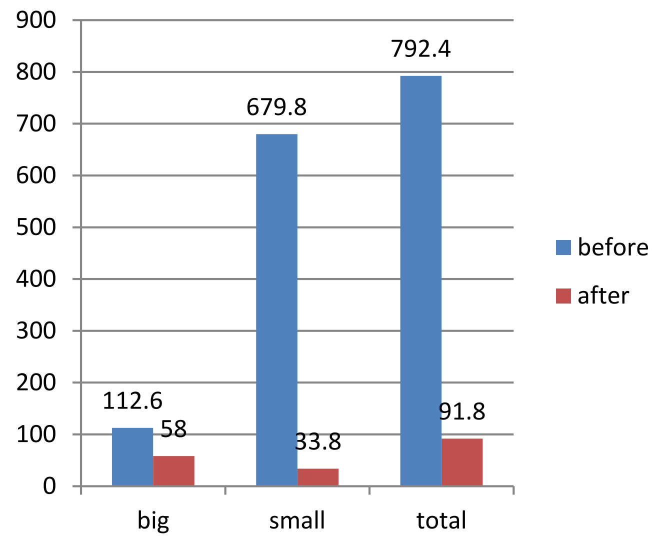
Results

- Real Environment Result

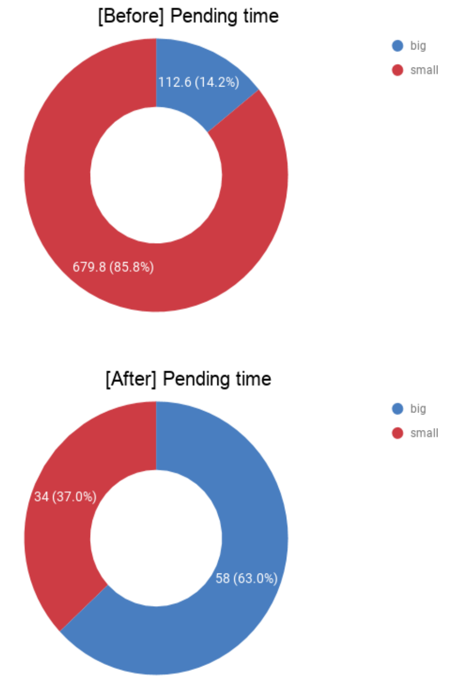
Algorithm : QoS, Greedy and Fair share combination



(A) Max pending time

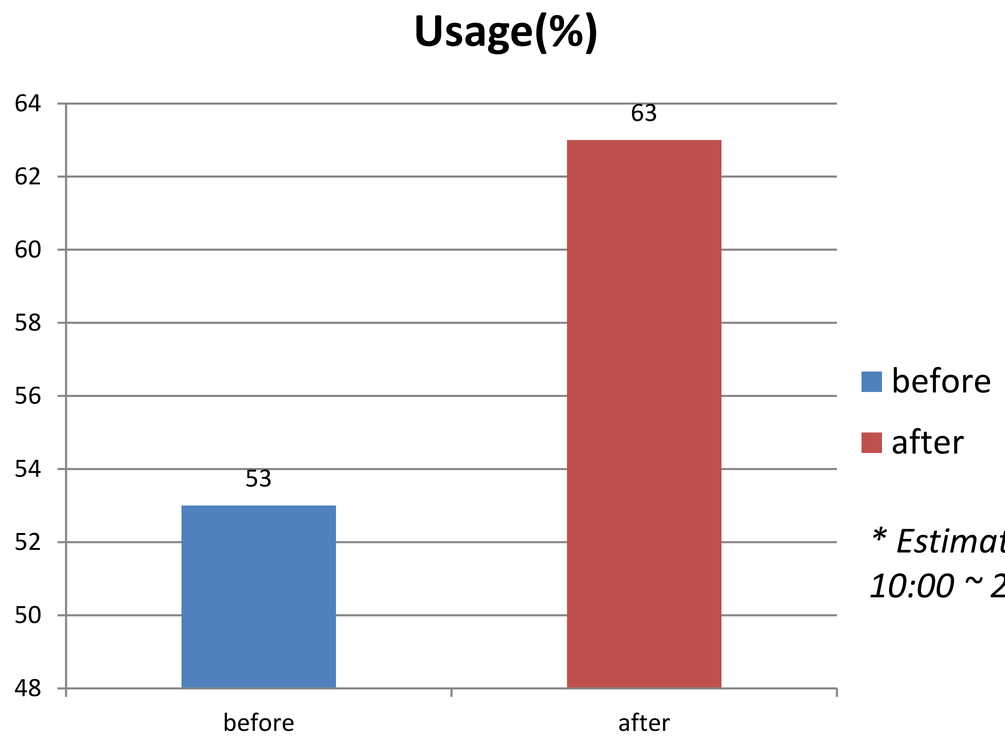


(B) Total pending time



(C) Average pending time ratio

Conclusions



** Estimated time :*
10:00 ~ 20:00 (operation time)

The increase in emulation resource utilization indicates
the **increase in the number of jobs** to run in emulation farms (**20%**)

Conclusions

- Contribution
  1. **Improve efficiency** for emulator management system
  2. First definition for **dynamic resource management** policy on emulator management system
  3. **First machine learning approach** on emulation management system

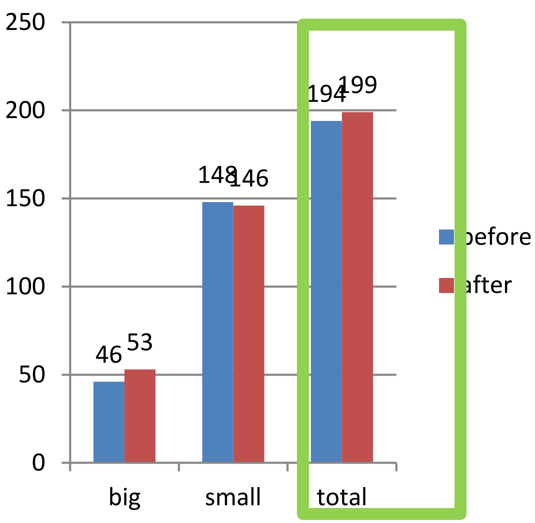
- Future works
  1. Advanced Reinforcement learning (A3C …)
  2. Common computing farm with N partitions

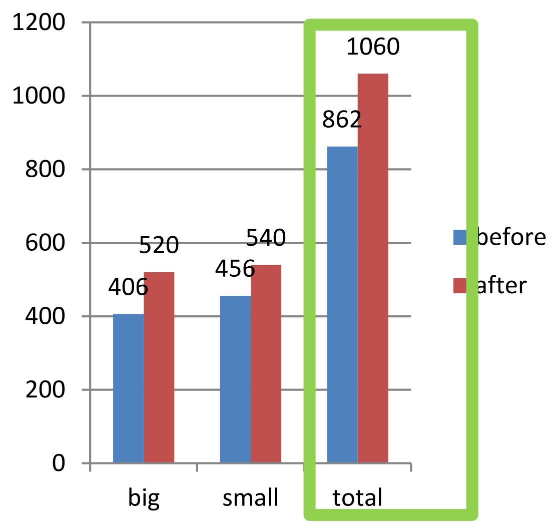
Q & A

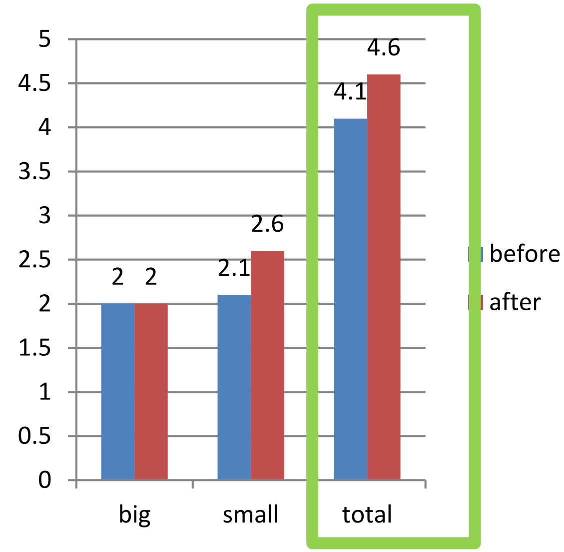
Or send an email to ***sangwoo.noh@samsung.com***

Appendix : Comparison

Before – 2019.03.13 / After – 2019. 07.02







Number of jobs			
	big	small	total
before	**46**	**148**	**194**
after	**53**	**146**	**199**
diff (rate)	**15.2%**	**-1.4%**	**2.6%**

Module size			
	big	small	total
before	**406**	**456**	**862**
after	**520**	**540**	**1060**
diff (rate)	**28.1%**	**18.4%**	**23%**

Average Run time(hours)			
	big	small	total
before	2	2.1	**4.1**
after	2	2.6	**4.6**

Implementation

Emulator status (ID:1), Team : Common

ZEBU4 | PZ1

Status matrix

Machine	UNIT	HM0	HM1	HM2	HM3	HM4	HM5	HM6	HM7
zebu4_10 (4/16)	U0	78869	78869						
zebu4_10 (4/16)	U1								
zebu4_10 (4/16)	U2	78539	78539			78814	78814	78814	78814
zebu4_10 (4/16)	U3(B)	78571	78571						
zebu4_20 (7/16)	U0(B)					78791	78791		
zebu4_20 (7/16)	U1	78751	78751	78862	78862	78841	78841	78841	78841
zebu4_20 (7/16)	U2(B)	78751	78751	78751	78751	78751	78751	78751	78751
zebu4_20 (7/16)	U3	78185	78185	78811	78811	78861	78861	78861	78861
zebu4_30 (4/11)	U0(B)	78860	78860	78860	78860	78860	78860	78860	78860
zebu4_30 (4/11)	U1(B)					78855	78855		
zebu4_30 (4/11)	U2(B)	78855	78855	78855	78855	78855	78855	78855	78855
zebu4_30 (4/11)	U3	78878	78878	78877	78877	78877	78877	78878	78878
zebu4_40 (5/10)	U0	78787	78787	78787	78787	78787	78787	78774	78774
zebu4_40 (5/10)	U1					78872	78872	78872	78872
zebu4_40 (5/10)	U2(B)	78831	78831	78831	78831	78831	78831	78831	78831
zebu4_40 (5/10)	U3	78873	78873	78805	78805	78805	78805	78805	78805
zebu4_50 (5/5)	U0(B)	78766	78766	78766	78766	78766	78766	78766	78766
zebu4_50 (5/5)	U1	78875	78875	75795	75795	75795	75795	78875	78875
zebu4_50 (5/5)	U2	78870	78870	78758	78758	78758	78758	78870	78870
zebu4_50 (5/5)	U3	78871	78871	78871	78871	78871	78871	78766	78766
zebu4_60 (8/12)	U0(B)	78750	78750	78750	78750	78750	78750	78750	78750
zebu4_60 (8/12)	U1	78465	78465	77884	77884	78465	78465	78465	78465
zebu4_60 (8/12)	U2	78820	78820	78568	78568	78798	78798	77565	77565

Emulator farm status

ZEBU Job list

JobID	Status	User	Group	Team	Host	Interactive time...	Submit time	Resource
78791	RUN	hs43.park	s5e9810...	IDT-DV-SV	zebu4_211	480min	2019-03-19 09:46:39.0	Module cnt : 2
78787	RUN	bho.lee	s5ahr80	PLATFOR...	zebu4_410	240min	2019-03-19 09:33:33.0	Module cnt : 6
78781	RUN	lab700	s5e9830	IDT-DI	zebu4_703	480min	2019-03-19 09:24:32.0	Module cnt : 4
78777	RUN	jaeho79....	artpece...	FOUNDRY	zebu4_719	480min	2019-03-19 09:23:42.0	Module cnt : 2
78773	RUN	jm5870	s5e9830	IDT-DV-PV	zebu4_618	480min	2019-03-19 09:20:55.0	Module cnt : 10
78766	RUN	sh78.song	s5e9830	IDT-DV-PV	zebu4_500	480min	2019-03-19 09:02:55.0	Module cnt : 10
78758	RUN	hwls715	fimgtrym	SOCDESI...	zebu4_503	0min	2019-03-19 08:38:55.0	Module cnt : 4
78751	RUN	jm5870	s5e9830	IDT-DV-PV	zebu4_209	480min	2019-03-19 08:04:39.0	Module cnt : 10
78750	RUN	hs43.park	s5e9810...	IDT-DV-SV	zebu4_615	480min	2019-03-19 07:59:04.0	Module cnt : 8
78571	RUN	m.murali	m.murali	DISPLAY-...	zebu4_114	90min	2019-03-18 10:52:51.0	Module cnt : 2
78568	RUN	douk.nam	douk.nam	DISPLAY-...	zebu4_605	90min	2019-03-18 10:45:59.0	Module cnt : 2
78539	RUN	buyoung...	buyoung...	DISPLAY-...	zebu4_101	90min	2019-03-18 09:04:04.0	Module cnt : 2
78465	RUN	sangjo.lee	s5ahr80	PLATFOR...	zebu4_613	240min	2019-03-16 13:31:33.0	Module cnt : 6
78414	RUN	th50.kim	th50.kim	DISPLAY-...	zebu4_707	90min	2019-03-15 18:19:09.0	Module cnt : 2
78185	RUN	th50.kim	th50.kim	DISPLAY-...	zebu4_203	90min	2019-03-14 17:59:48.0	Module cnt : 2
78125	RUN	th50.kim	th50.kim	DISPLAY-...	zebu4_711	90min	2019-03-14 15:15:59.0	Module cnt : 2
77884	RUN	drain.lee	drain.lee	DISPLAY-...	zebu4_607	90min	2019-03-13 14:37:50.0	Module cnt : 2
77647	RUN	bo.ahn	swsol	PLATFOR...	zebu4_606	300min	2019-03-12 14:29:35.0	Module cnt : 4
77565	RUN	min.byun	min.byun	DISPLAY-...	zebu4_611	90min	2019-03-12 10:05:13.0	Module cnt : 2
77413	RUN	min.byun	min.byun	DISPLAY-...	zebu4_714	90min	2019-03-11 16:10:27.0	Module cnt : 2
77407	RUN	min.byun	min.byun	DISPLAY-...	zebu4_710	90min	2019-03-11 16:01:14.0	Module cnt : 2

Job status