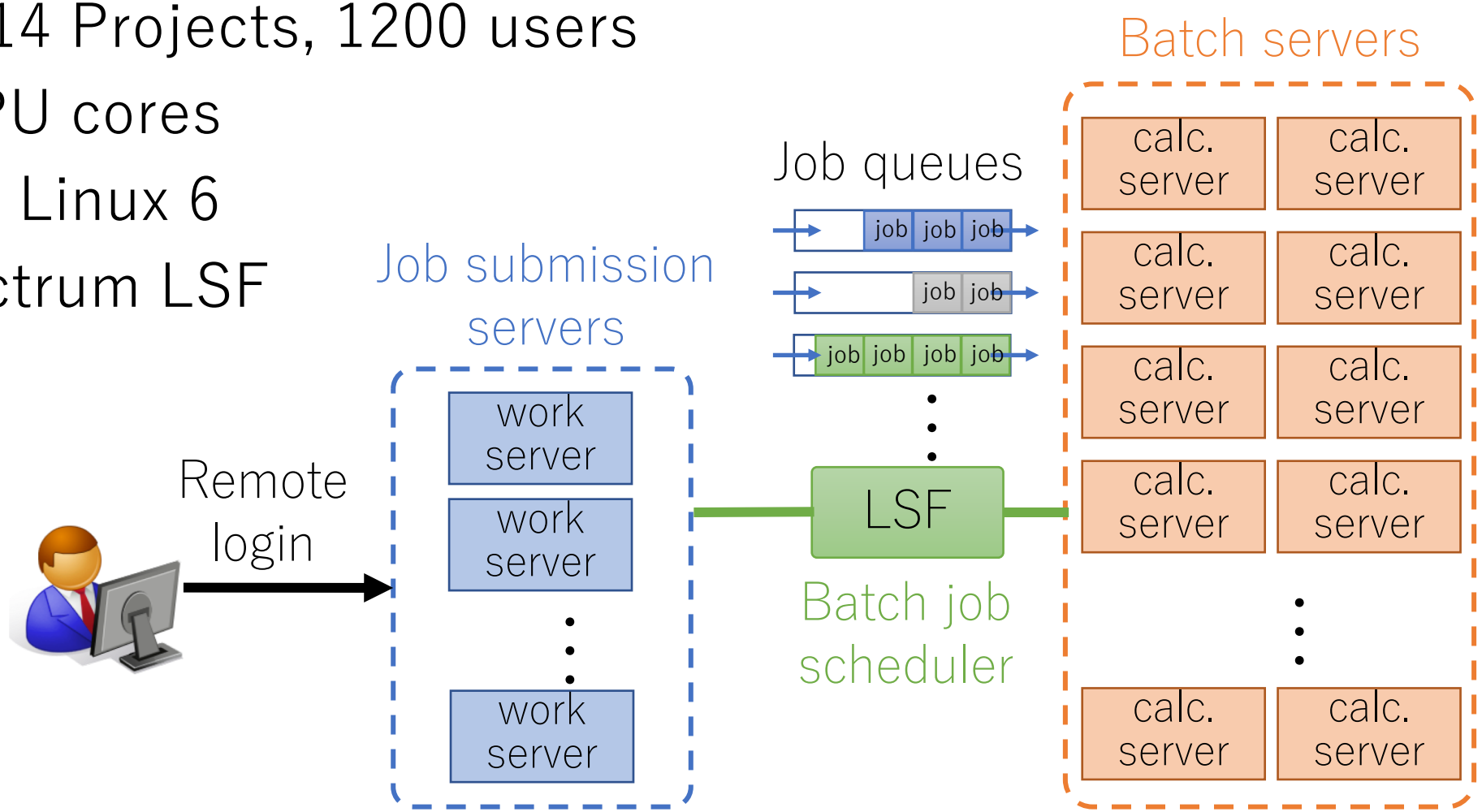
Waiting Time Estimation in Batch System by DL

Wataru Takase
Computing Research Center, KEK

2nd December, 2019

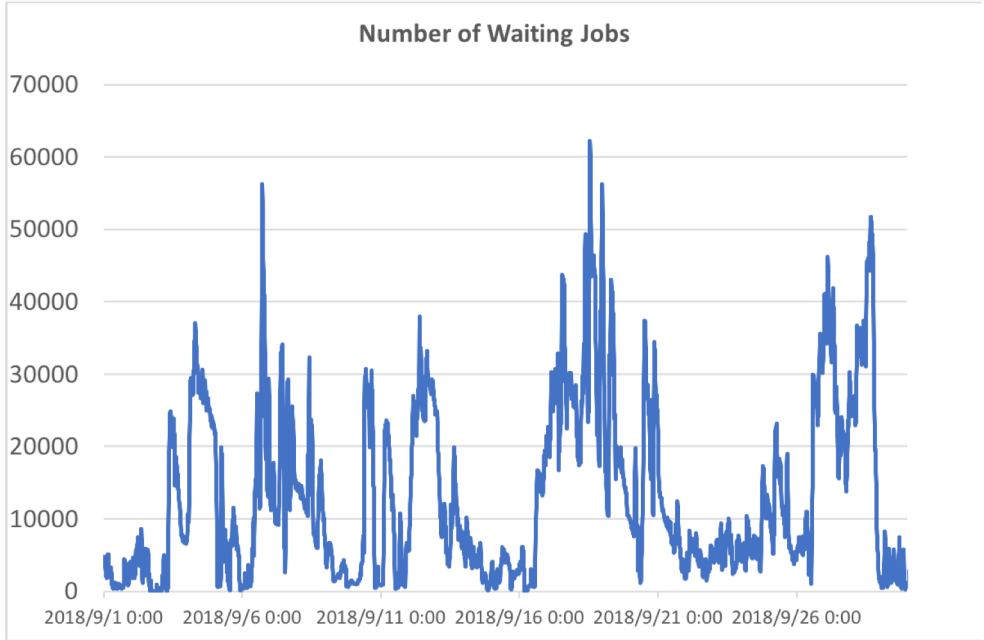
KEK Batch System

- Used by 14 Projects, 1200 users
- 10000 CPU cores
- Scientific Linux 6
- IBM Spectrum LSF

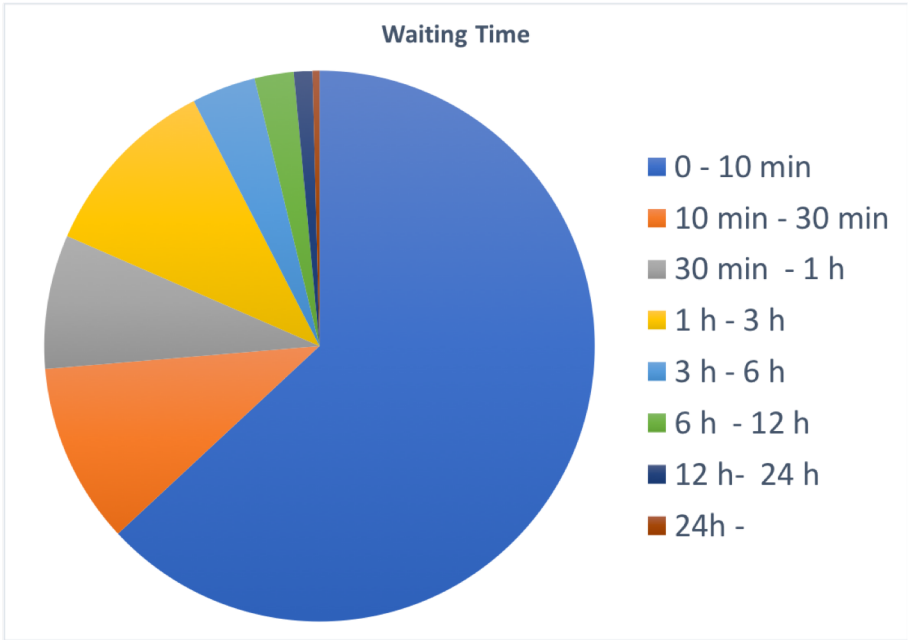


KEK Batch System: Piled up Waiting Jobs

- Available Job Slots: 10000
  - Limited by Number of CPU cores
- At the time of congestion, user jobs make a long stay in a job queue
  - Fairshare based scheduling
- Users may want to know when my jobs will start.



2018/9/1 2018/9/30



2018/9/1 – 2018/9/30

Waiting Time Estimation in Batch system by Supervised DL

- Each job history and events are save to *lsb.acct* and *lsb.events* files:
  - lsb.acct: The batch job log file of LSF.
  - lsb.events: The LSF batch event log file used to display event history.
    - Job new, job accept, job start, job move, etc…

➡ Input the job logs to a supervised DL model and try to estimate waiting time

Available information in lsb.acct and lsb.evnets

Used fields for the estimation

Field	Description
jobId	ID for the job
userId	UNIX user ID of the submitter
userName	User name of the submitter
options	Bit flags for job processing
numProcessors	Number of processors initially requested for execution
jStatus	Job status
submitTime	Job submission time
beginTime	Job start time – the job should be started at or after this time
termTime	Job termination deadline – the job should be terminated by this time
startTime	Job dispatch time – time job was dispatched for execution
endTime	Job completion time
queue	Name of the job queue to which the job was submitted
resReq	Resource requirement specified by the user
dependCond	Job dependency condition specified by the user
preExecCmd	Pre-execution command specified by the user

Field	Description
fromHost	Submission host name
cwd	Current working directory
inFile	Input file name
outFile	Output file name
errFile	Error output file name
jobFile	Job script file name
numAskedHosts	Number of host names to which job dispatching will be limited
askedHosts	List of host names to which job dispatching will be limited
numExHosts	Number of processors used for execution
execHosts	List of execution host names
cpuTime	CPU time consumed
runTime	Time in seconds that the job has been in the run state
jobName	Job name
command	Complete batch job command specified by the user
lsfRusage	resource usage information for the job
…	…

https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lsf_config_ref/lsb.acct.5.html
https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lsf_config_ref/lsb.events.5.html

0. Preparation of Input Data

1. Create CSV file from lsb.acct and lsb.events files.

submit time	pend time	user	queue	start time	end time	num processors	run time	cpu time
2017-08-10 02:37:17	7	user01	h	2017-08-10 02:37:24	2017-08-17 17:02:15	1	656691	12306
2017-08-10 02:37:18	13	user02	s	2017-08-10 02:37:31	2017-08-18 05:00:00	1	699749	24178
…	…	…	…	…	…	…	…	…

2. Split the CSV file for each user.

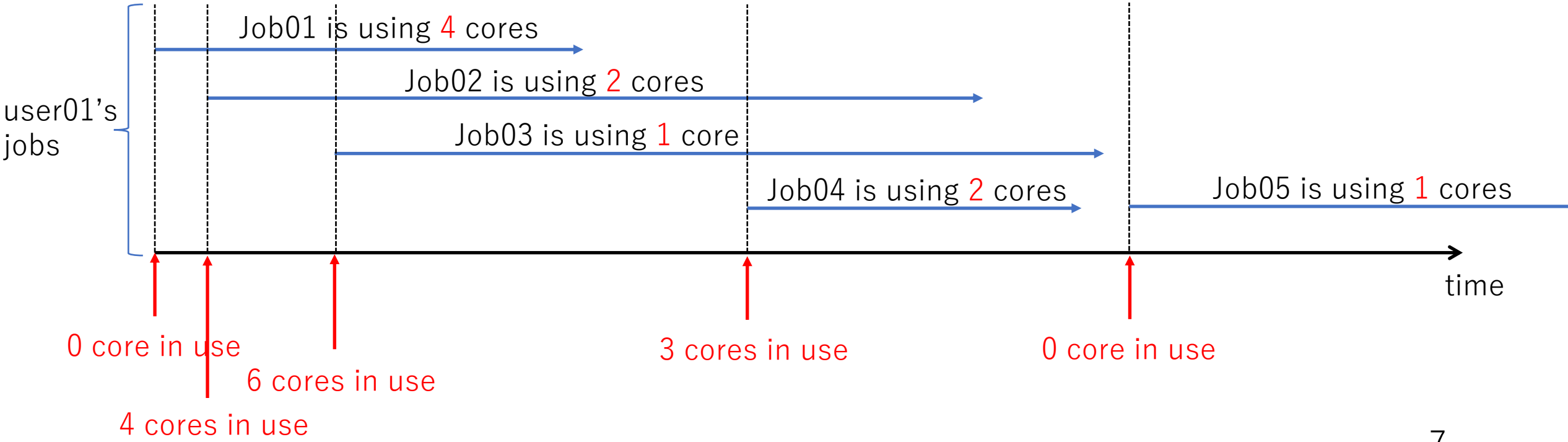
submit time	pend time	user	queue	start time	end time	num processors	run time	cpu time
2017-08-10 02:37:17	7	user01	h	2017-08-10 02:37:24	2017-08-17 17:02:15	1	656691	12306
…	…	…	…	…	…	…	…	…

submit time	pend time	user	queue	start time	end time	num processors	run time	cpu time
2017-08-10 02:37:18	13	user02	s	2017-08-10 02:37:31	2017-08-18 05:00:00	1	699749	24178
…	…	…	…	…	…	…	…	…

0. Preparation of Input Data

3. Enrich data

- Ex. Calculate "number of CPU cores currently in use" at the time of each job submission



0. Preparation of Input Data

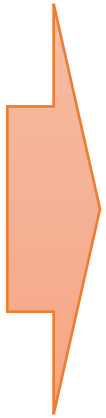
4. Classify waiting times in 6 classes

Waiting time	Class
0 to 10 mins	0
10 to 30 mins	1
30 min to 1 hour	2
1 to 3 hours	3
3 to 6 hours	4
More than 6 hours	5

0. Preparation of Input Data

5. Balance data

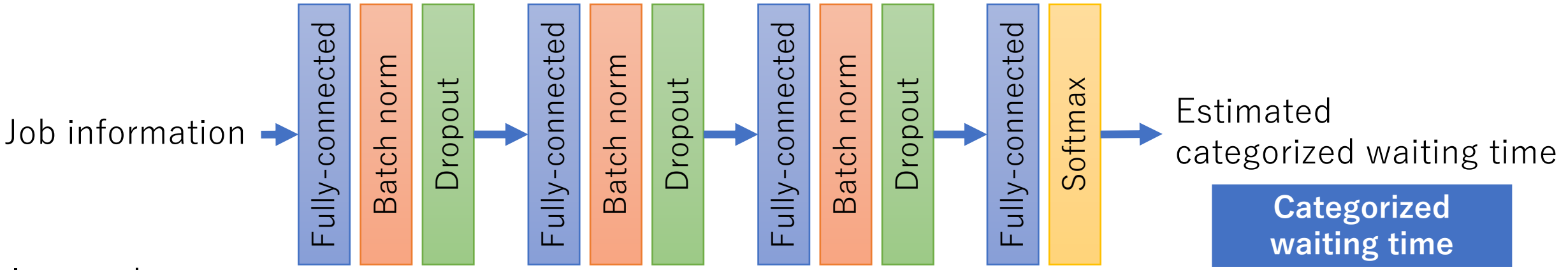
Class	Number of data
0	672908
1	97056
2	57547
3	97829
4	31945
5	40797



Class	Number of data
0	31945
1	31945
2	31945
3	31945
4	31945
5	31945

Random majority undersampling

1. Set up Fully-Connected Neural Network model

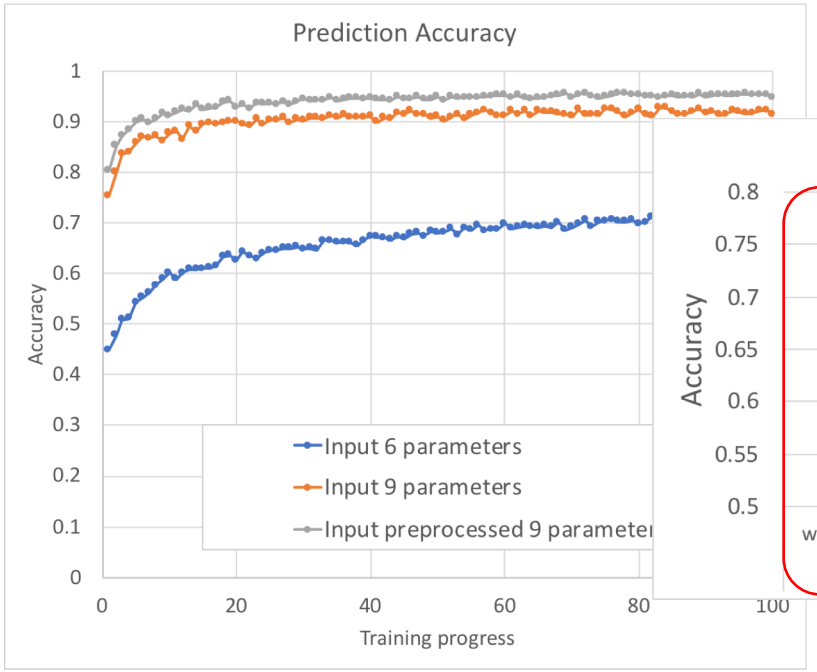


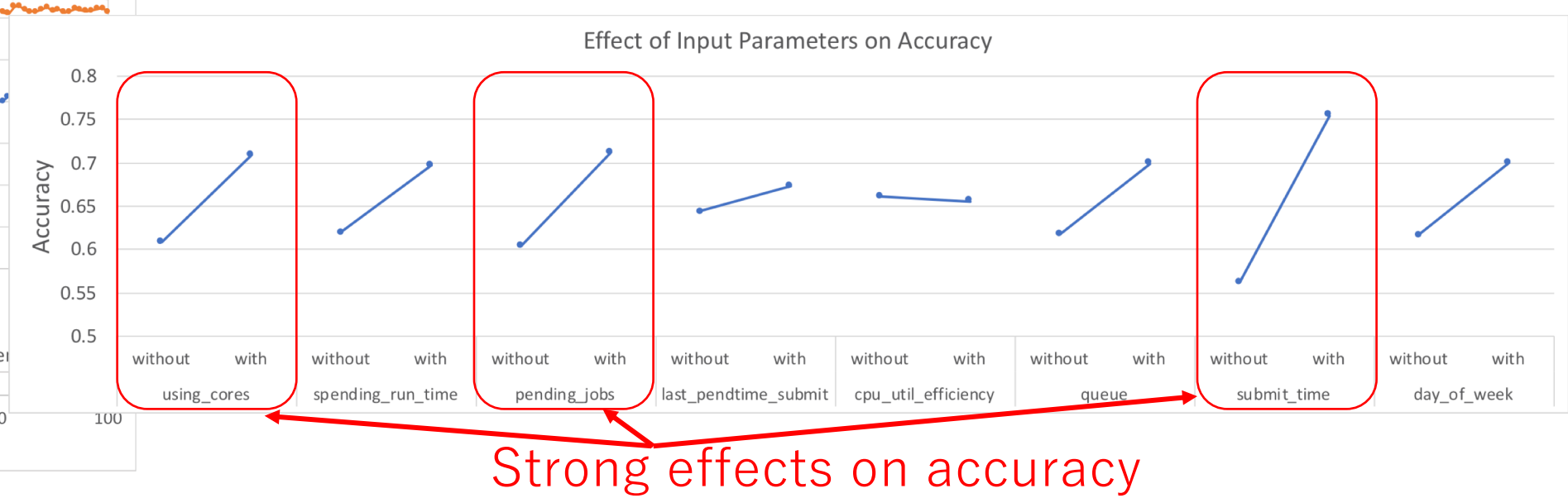
- Input data
  - Finished job logs during a certain period: 7 months

Input job information from *lsb.acct* files	
The last observed user's waiting time	Run time currently spent
Submission time of the above	Day of submission
Number of cores currently in use	Submission time in 24 hours
Current CPU utilization efficiency	Queue
Number of current waiting jobs	

Categorized waiting time
0 to 10 mins
10 to 30 mins
30 min to 1 hour
1 to 3 hours
3 to 6 hours
More than 6 hours

2. Train the model with TensorFlow using single NVIDIA Tesla K20 GPU





- Achieved 95% accuracy after a few hours training
- The training time depends on number of layers, number of neurons on each layer, input dataset size, and GPU power

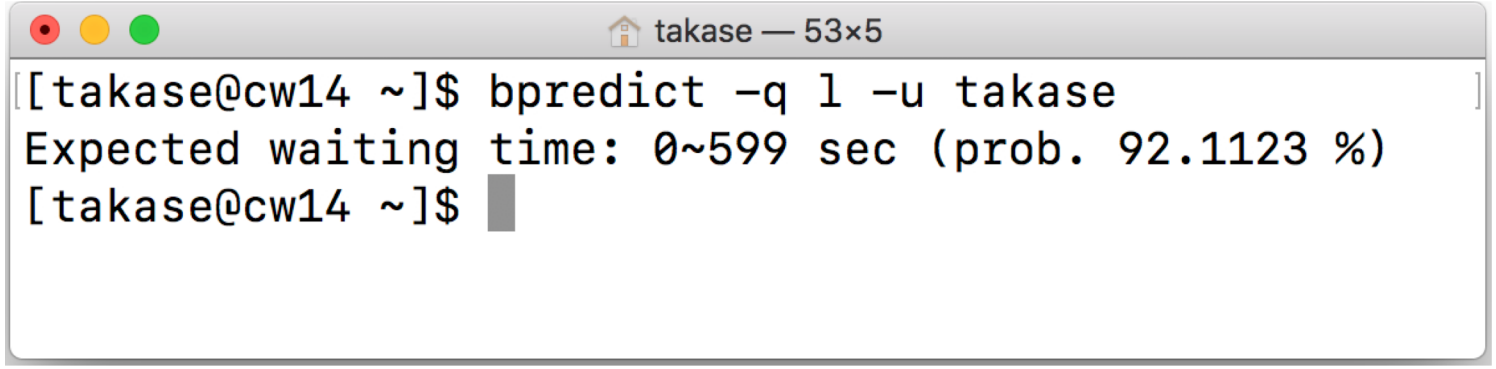
3. Estimate by the Trained Model

- Used dataset for the training
  - Job histories from January, 2018 to August, 2018.
- Accuracy against the above dataset: 95%
- Accuracy against dataset of another period (September, 2018): 87%

- Usage situation changes constantly, it's better continuing to re-train our model periodically by using fresh finished jobs information.

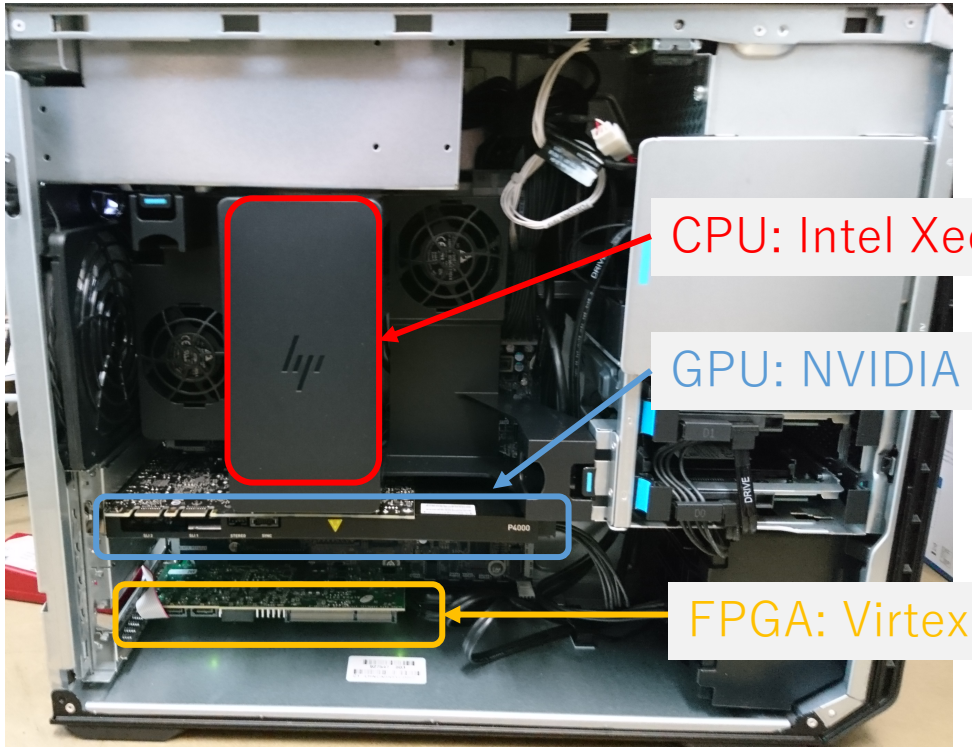
4. Implemented Waiting Time Estimator as a Command

- A user inputs 2 parameters: Queue, Username(option)
  - The other necessary information is collected automatically.
    - The last observed user's waiting time
    - Submission time of the above
    - Run time currently spent
    - Day of submission
    - Number of cores currently in use
    - Number of current waiting jobs
    - Submission time in 24 hours



Comparison of Inference Performance on CPU, GPU, FPGA

- Prepare a workstation which has CPU, GPU, and FPGA.
- Compare waiting time inference speeds.



Comparison of Inference Performance on CPU, GPU, FPGA

- GPU is the best.

Throughput comparison　　Throughput/Watt comparison

Inference Speed Comparison	Inference / sec	vs. Xeon 1 core Numpy	Accuracy	Inference/sec / Watt	vs. Xeon 1core Numpy	Price	
4-layer FC (W:32, A:32) on Intel Xeon W-2123 (3.6GHz) 1 CPU core, Numpy	8.4	1	0.956	0.095	1	€270	
6-layer FC (W:1, A:2) on FPGA Virtex 7 XC7VX690T-2	132.7	15.8	0.936	1.685	17.7	€3000	
4-layer FC (W:32, A:32) on GPU NVIDIA Quadro P4000, Cupy	1417.9	168.8	0.956	8.861	93.3	€1350	

Our Interesting Topics

- Is it applicable to the batch system at CC-IN2P3?
  - We got CC-IN2P3 batch team's consent for sharing the CC-IN2P3's data on 10th October.
- Run time estimation.
  - Based on user's command, input file, queue, …
- Make batch system resource usage efficiently by DL:
  - Ex. Estimate batch system congestion
- Anomaly detection by unsupervised DL or ML.
- Reduce training time by using many GPUs.
- Reduce inference time by using GPUs and FPGAs.