Practical Resource Usage Prediction Method for Large Memory Jobs in HPC clusters

IBM
Spectrum
Computing

Xiuqiao Li, Nan Qi, Yuan Yuan He
{lxiuqiao,qinan, yyhe}@cn.ibm.com
IBM China Systems Laboratory

Bill.McMillan@uk.ibm.com

IBM Spectrum Computing
IBM Cognitive Systems

Agenda

Motivation

Observations from real production job traces

Problem and design purpose

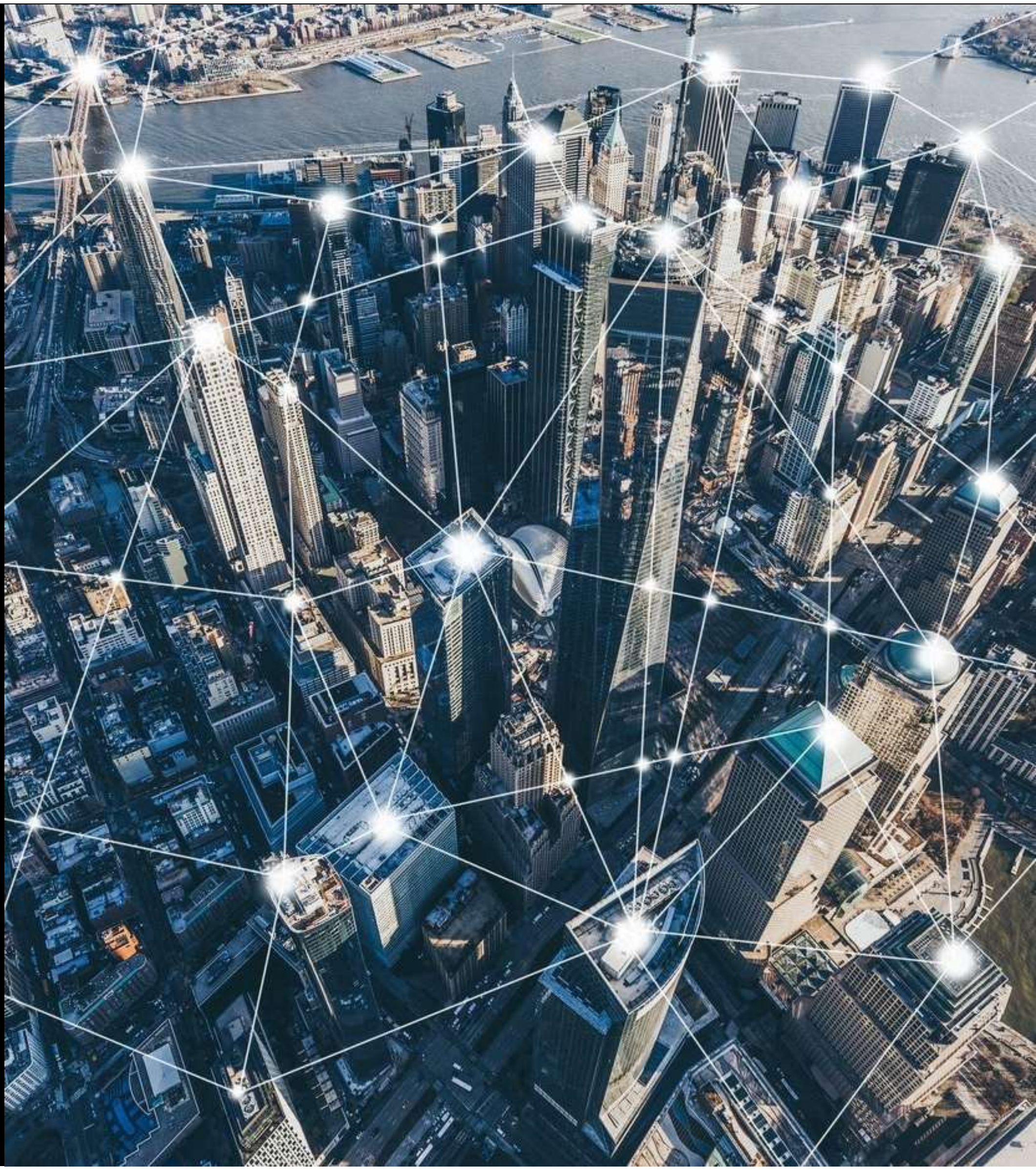
Two-stage large memory job prediction method

Evaluation results and analysis

Summary

Motivation

We aren’t that good at estimating

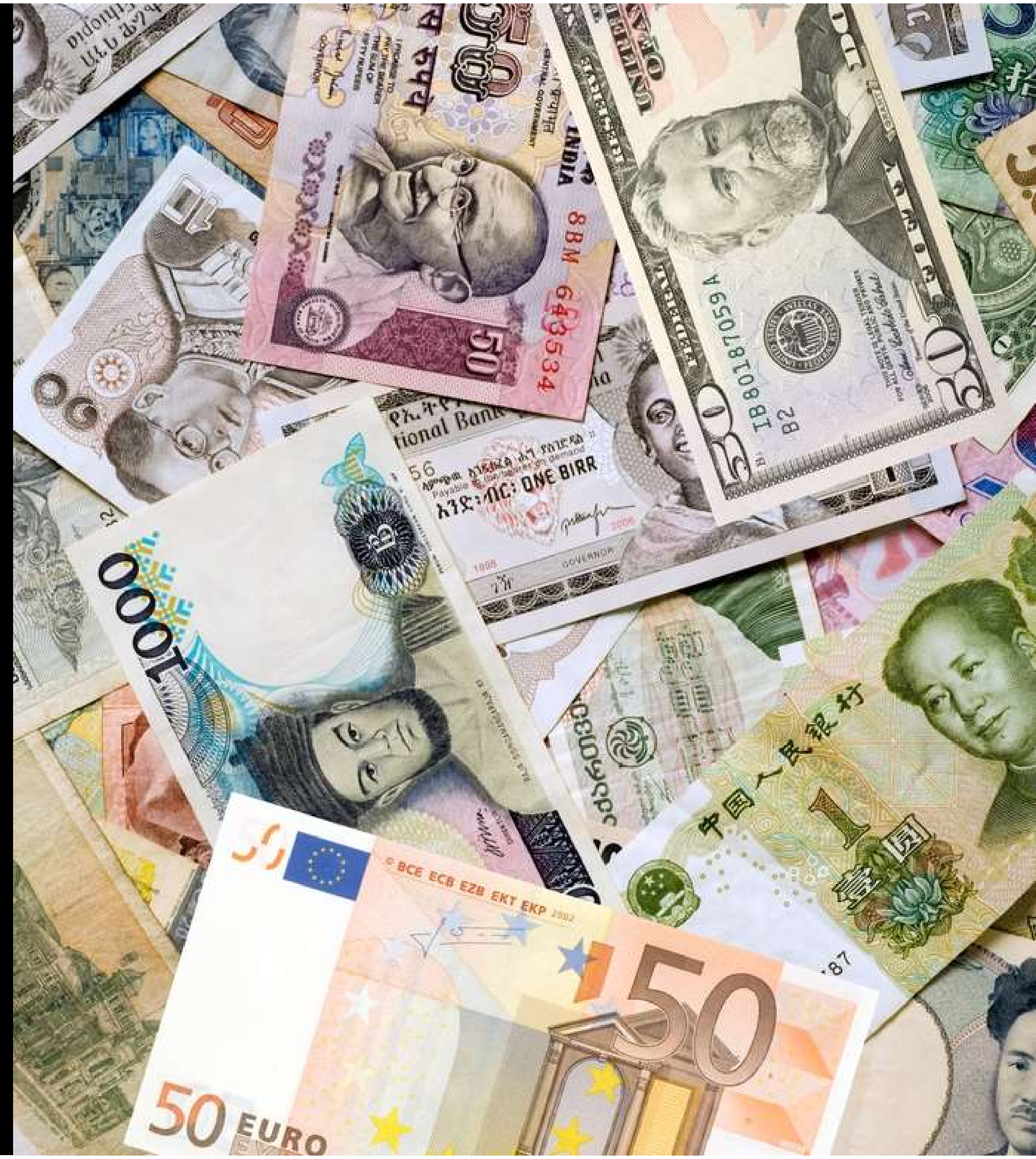


Impact

Lost Productivity

Impact

Increased Costs



The Need for Resource Prediction

Workload managers enforce scheduling policies based on job resource requirements (e.g. cores, memory size, job runtime limit, etc)

Typically, Users are not very good at estimating a jobs memory requirements or run time, and will often over estimate - e.g. asking for all the memory on a node, even if the job only really needs a small amount.

This leads to significant resource wastage, with increased turnaround times and costs (especially for single node/high throughput workloads).

Educating Users to provide better estimates is hard.

Job resource usage generally can be predicted as applications tend to be repeatedly executed in HPC clusters.

Large Memory in High Throughput Production LSF Clusters

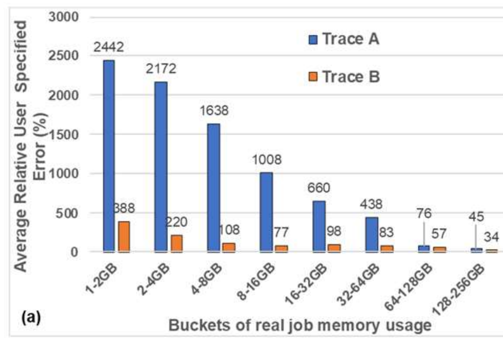
***Observation 1**: **Small** number of **large** memory jobs consume most of job memory*

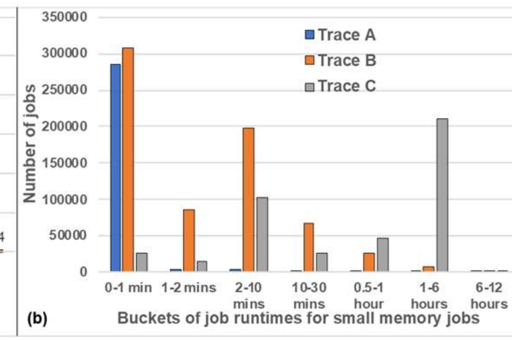
***Observation 2**: User specified memory sizes tend to be over-estimated with large errors*

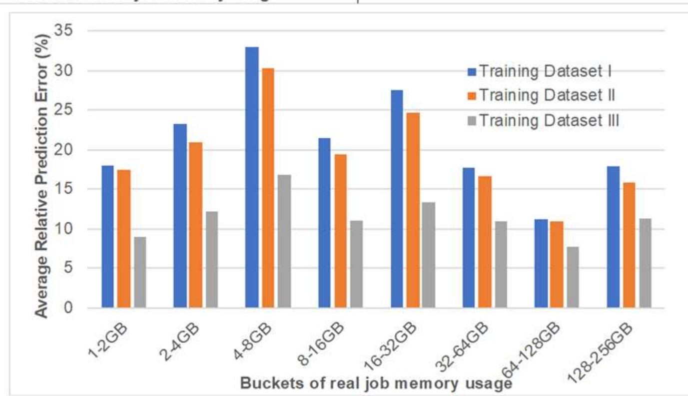
***Observation 3**: **Small** memory jobs tend to have **short** run time*

***Observation 4**: Prediction quality for **large** memory jobs reduces with more **small** memory jobs considered in training sets*

Traces	#Jobs	Small memory jobs (%)	Small memory usage (%)	Large memory jobs (%)	Large memory usage (%)
Trace A	587k	62.7	0.51	37.3	99.49
Trace B	907k	77	3.3	23	96.7
Trace C	1m	43.4	12.1	56.6	87.9







Problem Explored in This Study

Problem

- Improve job memory usage prediction for large memory jobs
- Administrators care more about the memory usage of large memory jobs
- Coarse grained memory requirements are acceptable for small memory jobs

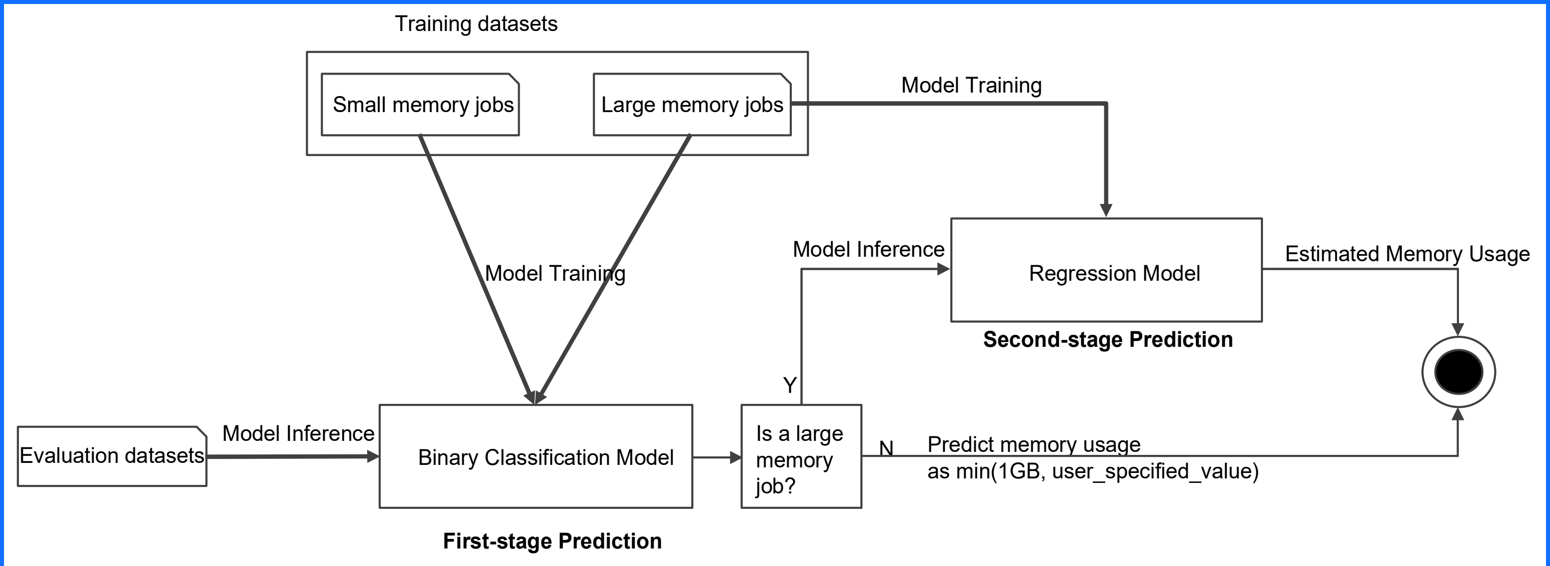
Design targets

- Improve prediction quality for large memory jobs with high coverage rates
- Reduce model training costs to enable frequent model updates
- Keep low model inference latency and reduce impacts on job submission

Two-stage memory usage prediction

Proposed method: combined two stage model to improve large memory job predictions

- A binary classification model to identify large memory jobs
- A regression model trained by only large memory jobs to predict large memory usage



Stage I: Memory Size Classification Efficiency

Binary classification model includes all jobs in training sets

– Good estimates should have large CR and small ICR

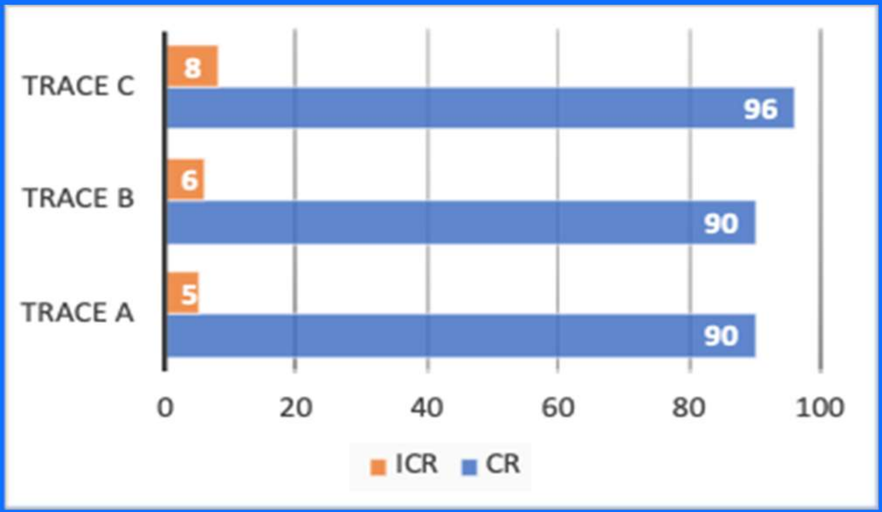
  CR=#Hit_LMEM_Jobs/#Total_LMEM_Jobs

  ICR=#Miss_SMEM_Jobs/#Total_SMEM_Jobs

– Without best hyper-parameter tuning, a binary classification model can have good estimates for testing traces

  • Binary classification complexity is lower than multi-class classification or regression

  • Classification accuracy can be further tuned with hyper-parameter settings

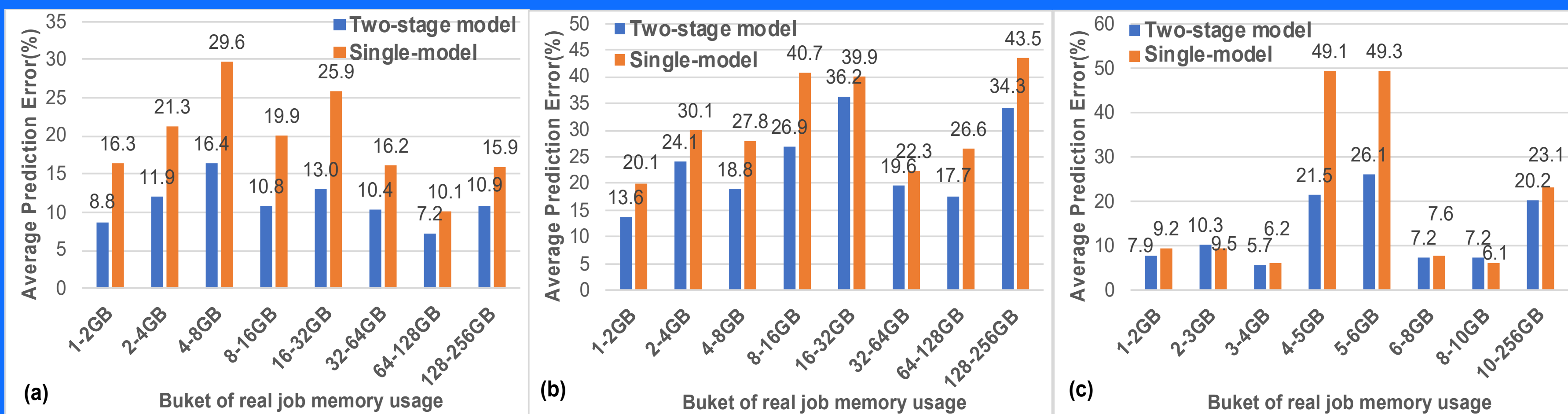


Prediction results with three production traces:
- Random forest model is used
- Hyper-parameters: n_estimators=50, n_jobs=10, max_depth=auto

Stage II: Regression Quality for Large Memory Jobs

The second stage adopts regression model which trained with only large memory jobs

- Average prediction errors can be reduced by 40.7, 24.3 and 14.5 percent compared with the single model approach
- Remove noise of small memory jobs achieve better prediction accuracy

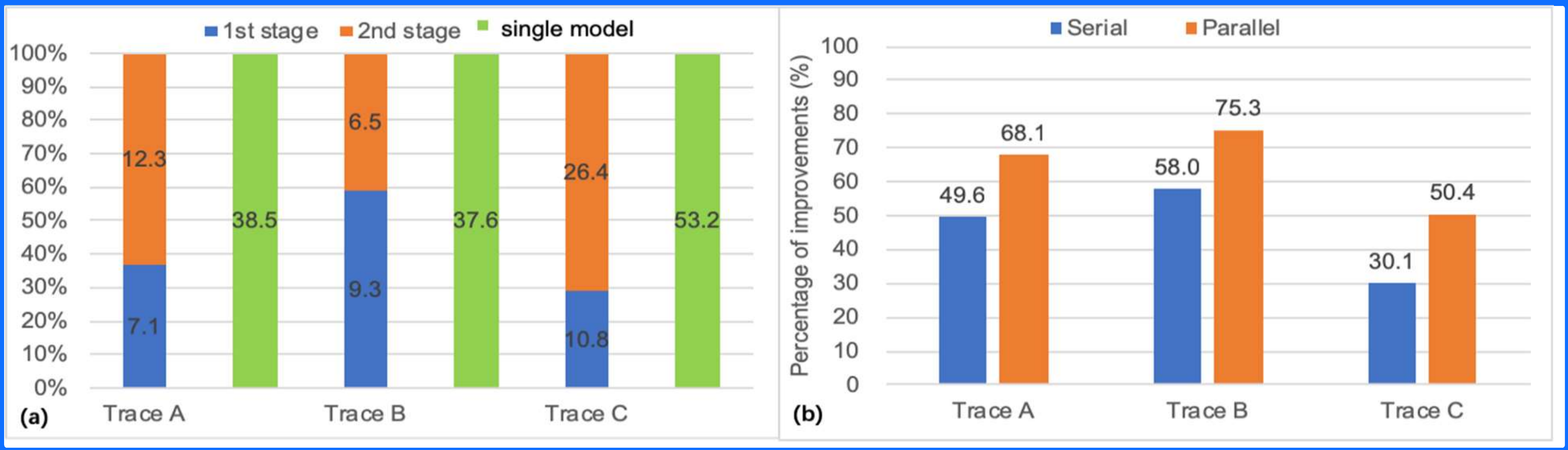


Comparisons of prediction errors for three production traces:
Random forest regression model with n_estimators=100 and n_jobs=10 is used for the tests

Benefits of Two Mode Training Costs

Binary classification cost is small due to its low training complexity

- The cost of the 2-stage regression model is significantly reduced due to removing large amounts of small memory jobs (noise)
- Running two models can be run in parallel due to no data dependency between two stage models



Comparisons of a) training cost in seconds and b) model training savings by training two-stage models sequentially or in parallel

Impacts on Model Inference Costs

The proposed two stage model prediction will add one more inference steps for each job:

- Results show that inference overhead is very small, and in most cases could be ignored when compared to normal job submission latency (especially when submission filters are used)
- Model inference delay can be further hidden by running two models in parallel with additional computing resources

Trace Name	Avg. model inference latency (microseconds)		
	1st stage	2nd stage	Single model
Trace A	2.38	7.28	4.88
Trace B	1.57	7.31	2.87
Trace C	1.76	4.49	3.04

Per-job average model inference latency

Trace Name	Total time of model inference (milliseconds)			Inference delay (per-cent)
	1st stage	2nd stage	Single model	
Trace A	279.6	318.2	597.8	4.38
Trace B	325.3	343.7	595.9	12.3
Trace C	442.8	618.2	765.7	38.6

Total cost of model inference and overhead compared with single model approach

Summary

Conclusions

- A small number of large memory jobs dominate the memory usage in these clusters
- The two-stage model approach can remove the noise of small memory jobs to get better prediction quality for large memory jobs
- The model achieves high prediction accuracy with little inference overhead.
- The model training costs can also be substantially reduced to enable possibility of frequent model updates

Future directions

- Further model tuning to minimize miss prediction for classifying large memory jobs
- Explore the application for predicting other job resource metrics. e.g. long running jobs

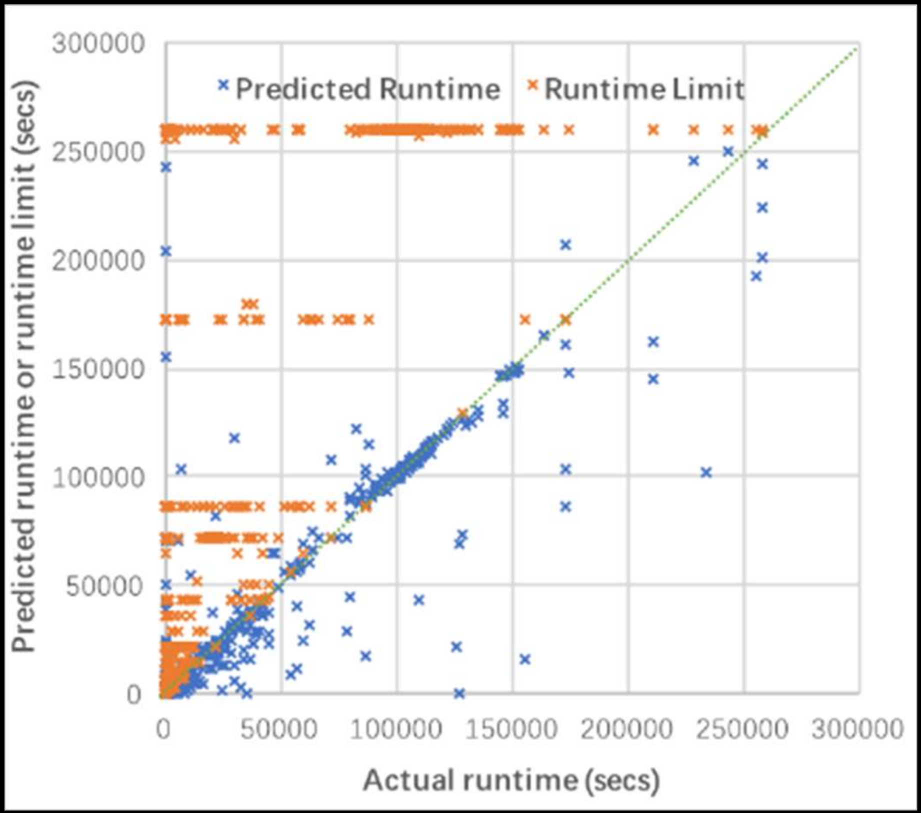
Runtime Prediction

Under-specifying runtime leads to:

- Jobs being killed - loss of productivity.
- Delayed execution for other users.
- Many organisations do not specify run limits as killing production workloads is unacceptable.

Over-specifying runtime leads to:

- Lower utilization - loss in productivity
- Poor backfill scheduling.
- Poor multi-cluster and hybrid cloud forwarding decisions



Notices and disclaimers

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."
Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

.