基于协同过滤的
大规模数据中心性能预测方法

GROUP 1

内容目录

- ♠ 问题背景介绍
- ♠ 主要创新方案
- ♠ 具体实施方法
- ♠ 实验结果演示

内容目录

- ♠ 问题背景介绍
- ♠ 主要创新方案
- ♠ 具体实施方法
- ♠ 实验结果演示

大规模数据中心中的几个实际问题

- ♠ 服务质量相关
  - ♣ 面对日益增长的应用请求，如何保证服务力量？
- ♠ 能源开销相关
  - ♣ 如何利用动态调度，动态资源分配提高能效？
- ♠ 数据中心建造成本相关
  - ♣ 如何数据中心更新中选择能效最高的服务器类型？

大规模数据中心中的资源分配问题

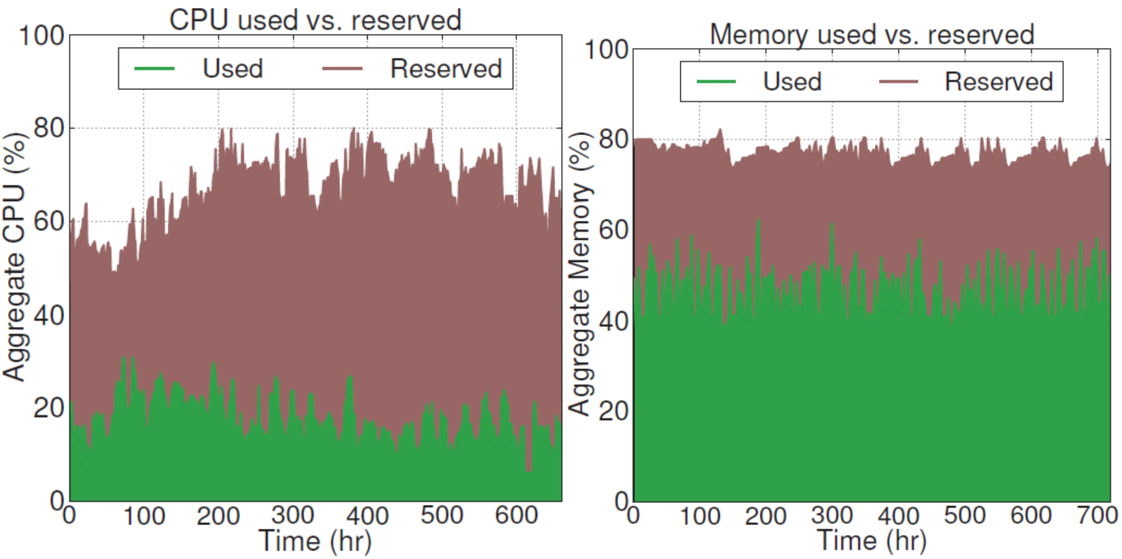


Figure 1: Resource utilization over 30 days for a cluster at **Twitter** managed with Mesos.

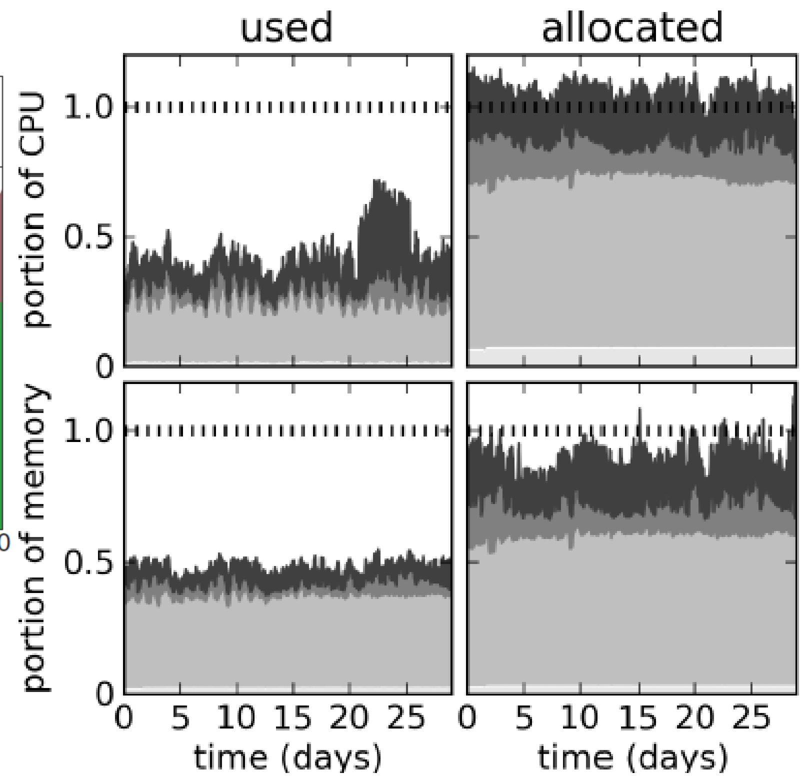
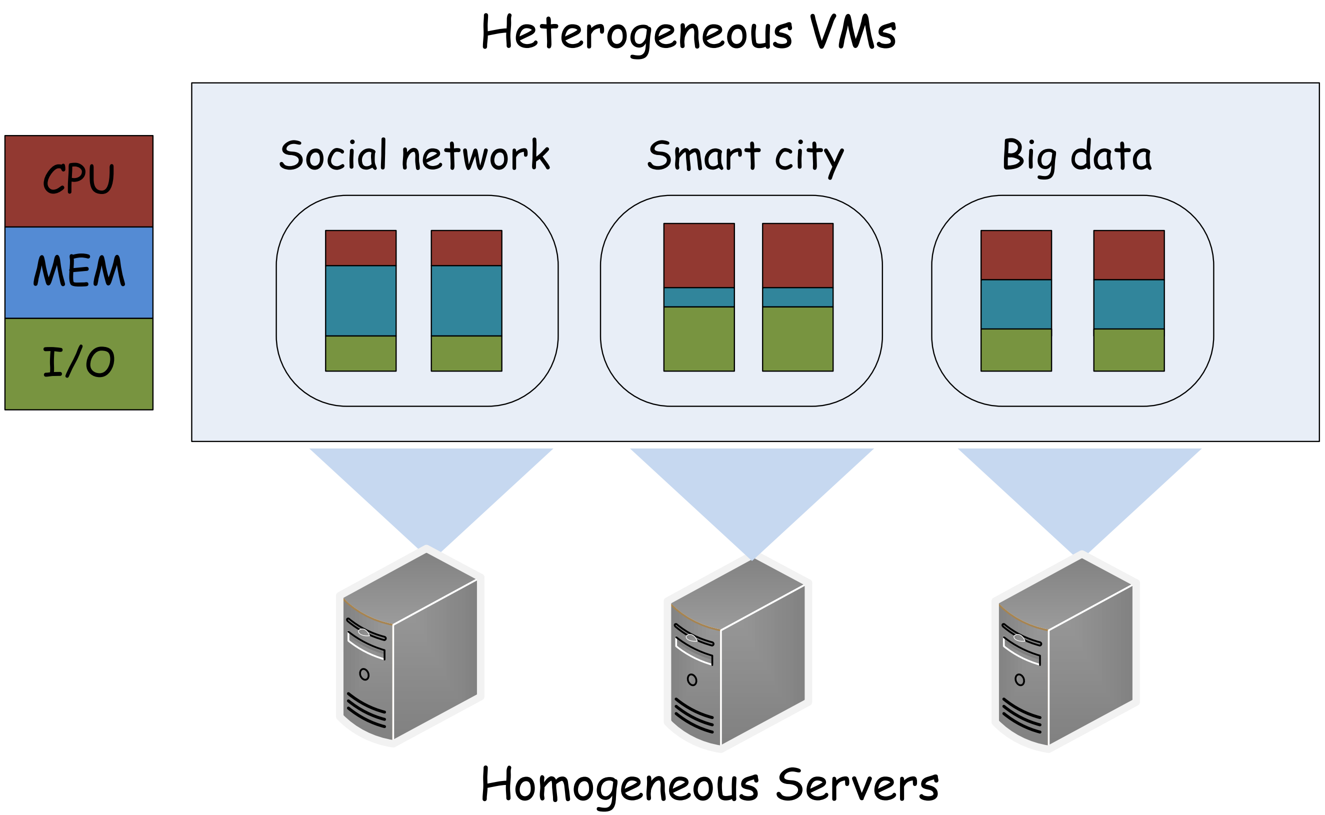


Figure 2: Moving 30 days average of CPU (top) and memory (bottom) utilization (left) and resource requests (right) for a cluster at **Google** managed with Borg.

拥有先进资源管理的互联网巨头
的数据中心资源利用率依旧不高

资源配置场景1



创建不同<CPU, MEM, I/O>配置的虚拟机来承载不同类型应用以最大化效益

资源配置场景2

Heterogeneous Servers

调度不同类型的应用至不同硬件配置的物理机以最大化效益

资源分配问题的本质

应用的需求（**demand**）是否匹配资源的分配（**allocation**）

缺少应用和资源交互（**interaction**）的知识（**knowledge**）

拟解决的核心问题（kernel）： $PERFORMANCE(Job_i, Machine_j)$

内容目录

- ♠ 问题背景介绍
- ♠ 主要创新方案
- ♠ 具体实施方法
- ♠ 实验结果演示

谷歌某数据中心的Big Data Trace

https://code.google.com/p/googleclusterdata/

Trace Characteristic	Value
Time span of trace	29 days
Jobs run	650k
Number of users (with usage)	925
Tasks submitted	25M
Scheduler events	143M
Resource usage records	1233M
Compressed size	39 GB

Number of machines	Platform	CPUs	Memory
6732	B	0.50	0.50
3863	B	0.50	0.25
1001	B	0.50	0.75
795	C	1.00	1.00
126	A	0.25	0.25
52	B	0.50	0.12
5	B	0.50	0.03
5	B	0.50	0.97
3	C	1.00	0.50
1	B	0.50	0.06

谷歌某数据中心的Big Data Trace

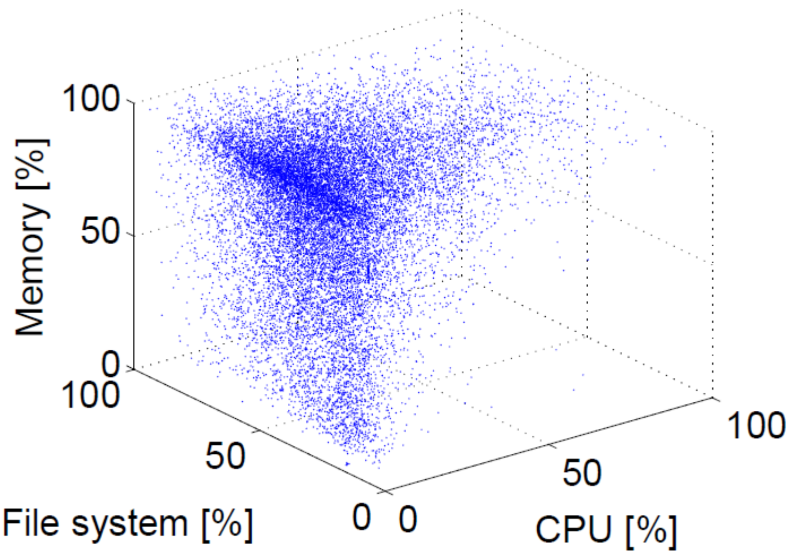


Figure 3: Diverse resource Demands on CPU, MEM and I/O, respectively.

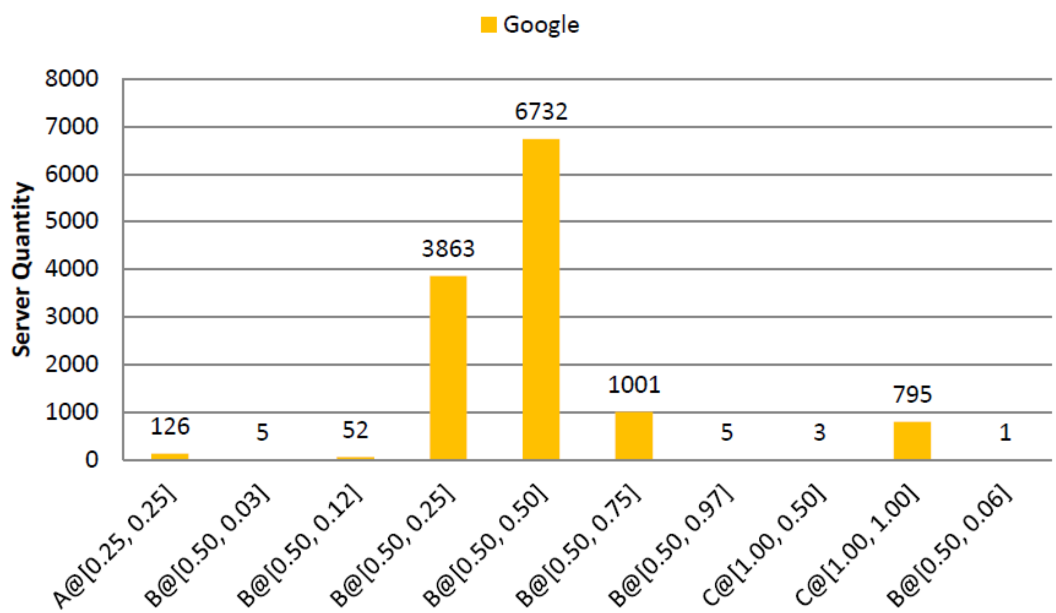


Figure 4: The quantity distributions of the ten types of machines.

谷歌某数据中心的Big Data Trace

file pattern	field number	content	format
job_events/part-?????-of-?????.csv.gz	1	time	INTEGER
job_events/part-?????-of-?????.csv.gz	2	missing info	INTEGER
job_events/part-?????-of-?????.csv.gz	3	job ID	INTEGER
job_events/part-?????-of-?????.csv.gz	4	event type	INTEGER
job_events/part-?????-of-?????.csv.gz	5	user	STRING_HASH
job_events/part-?????-of-?????.csv.gz	6	scheduling class	INTEGER
job_events/part-?????-of-?????.csv.gz	7	job name	STRING_HASH
job_events/part-?????-of-?????.csv.gz	8	logical job name	STRING_HASH

task_events/part-?????-of-?????.csv.gz	1	time	INTEGER
task_events/part-?????-of-?????.csv.gz	2	missing info	INTEGER
task_events/part-?????-of-?????.csv.gz	3	job ID	INTEGER
task_events/part-?????-of-?????.csv.gz	4	task index	INTEGER
task_events/part-?????-of-?????.csv.gz	5	machine ID	INTEGER
task_events/part-?????-of-?????.csv.gz	6	event type	INTEGER
task_events/part-?????-of-?????.csv.gz	7	user	STRING_HASH
task_events/part-?????-of-?????.csv.gz	8	scheduling class	INTEGER
task_events/part-?????-of-?????.csv.gz	9	priority	INTEGER
task_events/part-?????-of-?????.csv.gz	10	CPU request	FLOAT
task_events/part-?????-of-?????.csv.gz	11	memory request	FLOAT
task_events/part-?????-of-?????.csv.gz	12	disk space request	FLOAT
task_events/part-?????-of-?????.csv.gz	13	different machines restriction	BOOLEAN

谷歌某数据中心的Big Data Trace

task_usage/part-?????-of-?????.csv.gz	1	start time	INTEGER
task_usage/part-?????-of-?????.csv.gz	2	end time	INTEGER
task_usage/part-?????-of-?????.csv.gz	3	job ID	INTEGER
task_usage/part-?????-of-?????.csv.gz	4	task index	INTEGER
task_usage/part-?????-of-?????.csv.gz	5	machine ID	INTEGER
task_usage/part-?????-of-?????.csv.gz	6	CPU rate	FLOAT
task_usage/part-?????-of-?????.csv.gz	7	canonical memory usage	FLOAT
task_usage/part-?????-of-?????.csv.gz	8	assigned memory usage	FLOAT
task_usage/part-?????-of-?????.csv.gz	9	unmapped page cache	FLOAT
task_usage/part-?????-of-?????.csv.gz	10	total page cache	FLOAT
task_usage/part-?????-of-?????.csv.gz	11	maximum memory usage	FLOAT
task_usage/part-?????-of-?????.csv.gz	12	disk I/O time	FLOAT
task_usage/part-?????-of-?????.csv.gz	13	local disk space usage	FLOAT
task_usage/part-?????-of-?????.csv.gz	14	maximum CPU rate	FLOAT
task_usage/part-?????-of-?????.csv.gz	15	maximum disk IO time	FLOAT
task_usage/part-?????-of-?????.csv.gz	16	cycles per instruction	FLOAT
task_usage/part-?????-of-?????.csv.gz	17	memory accesses per instruction	FLOAT
task_usage/part-?????-of-?????.csv.gz	18	sample portion	FLOAT
task_usage/part-?????-of-?????.csv.gz	19	aggregation type	BOOLEAN

谷歌某数据中心的Big Data Trace

The performance metric uses **Time** or **Cycle per Instruction**

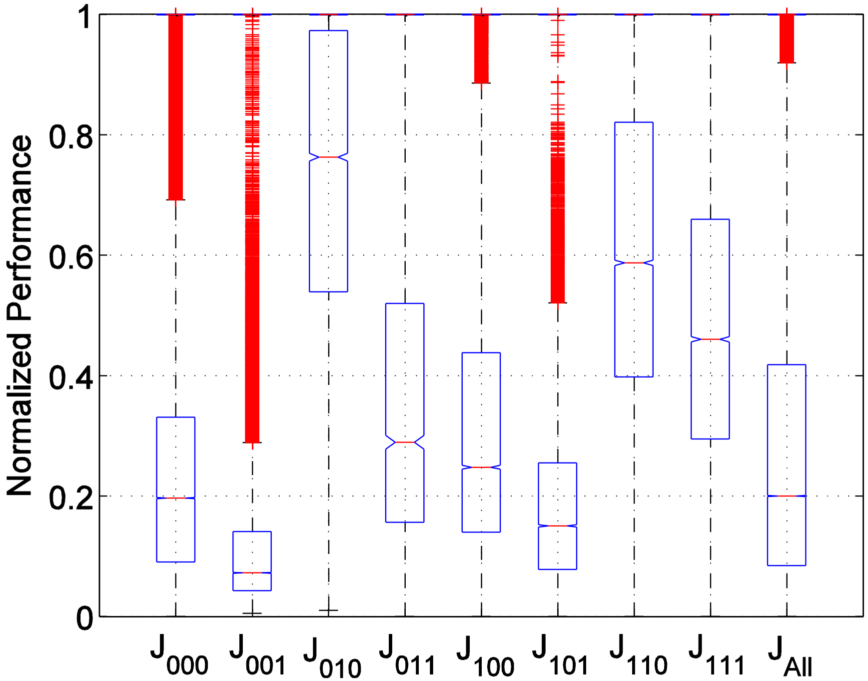


Figure 5: The performance distribution, shown by boxplot.

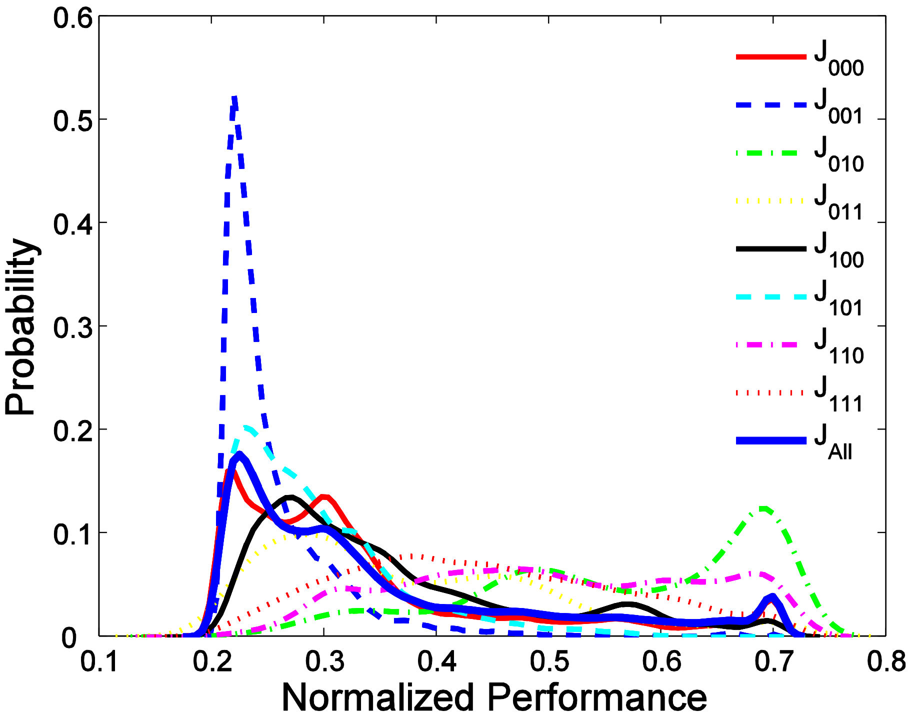


Figure 6: The performance distribution, shown by probability density function.

真实数据中心中稀疏的性能记录

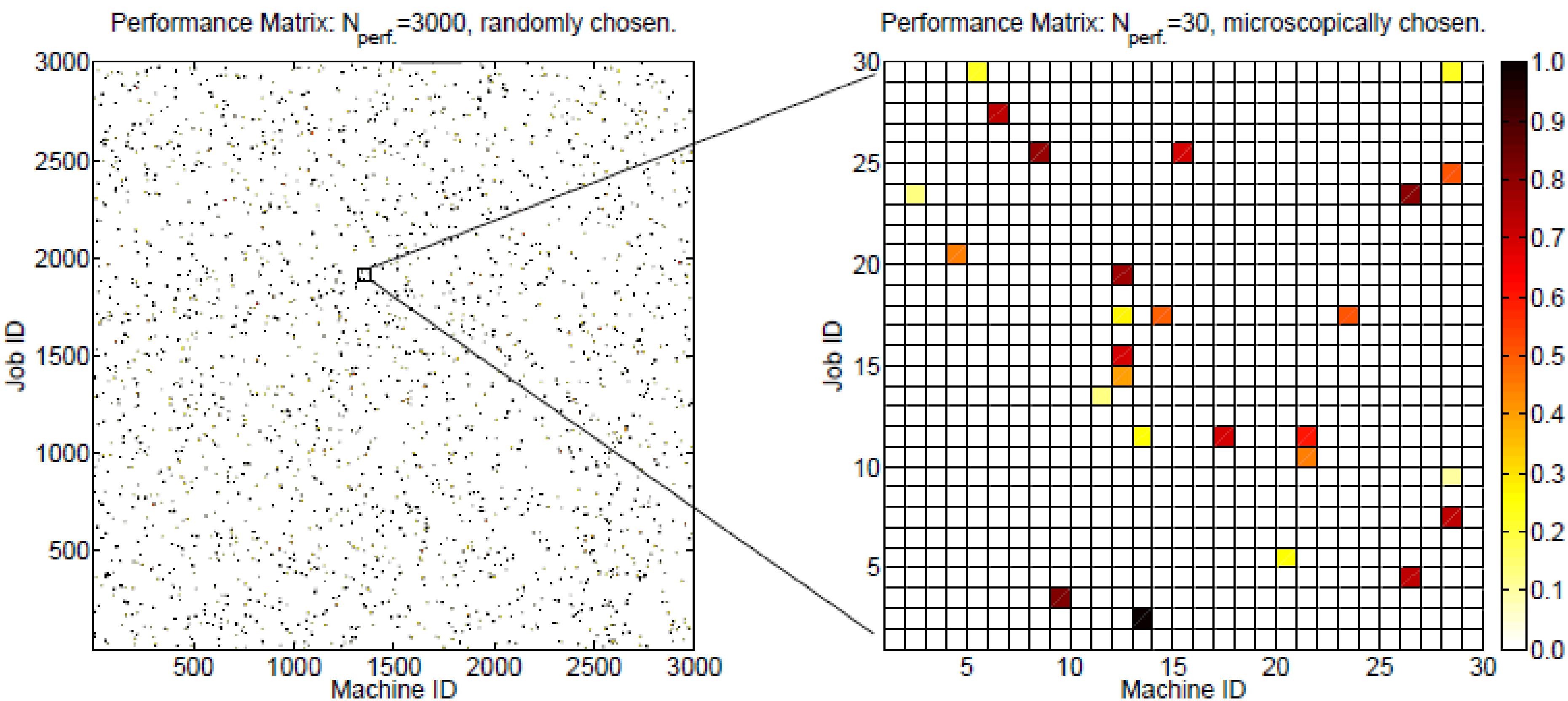
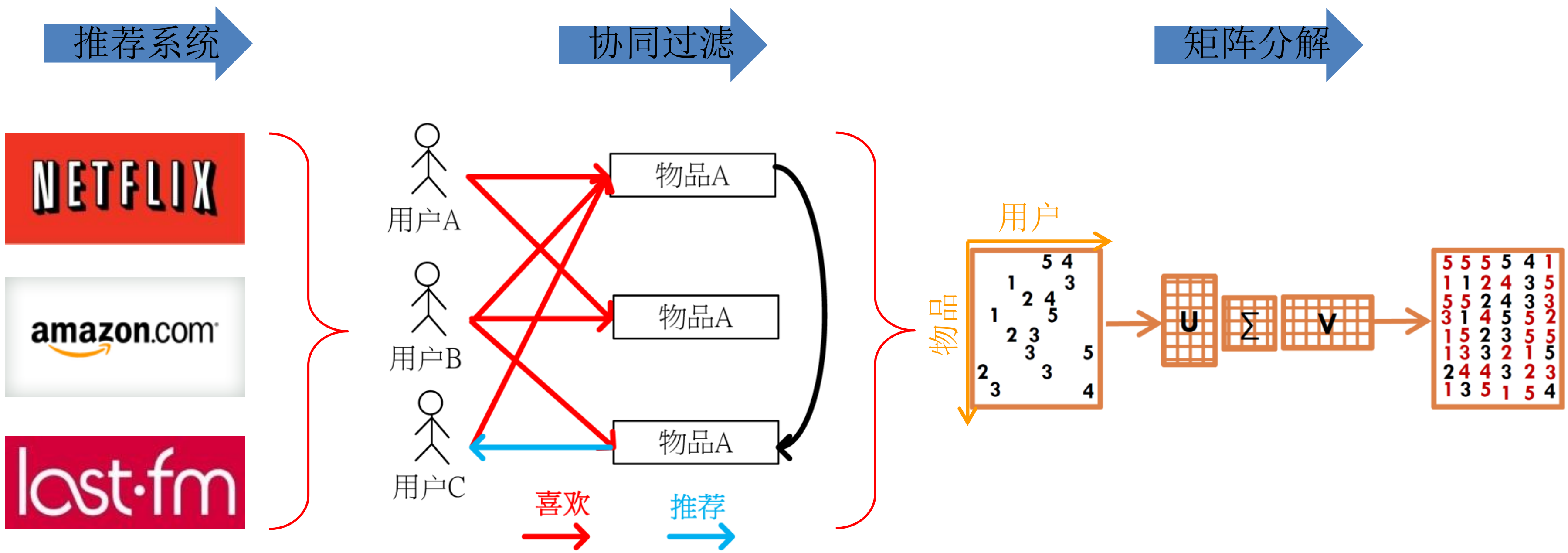


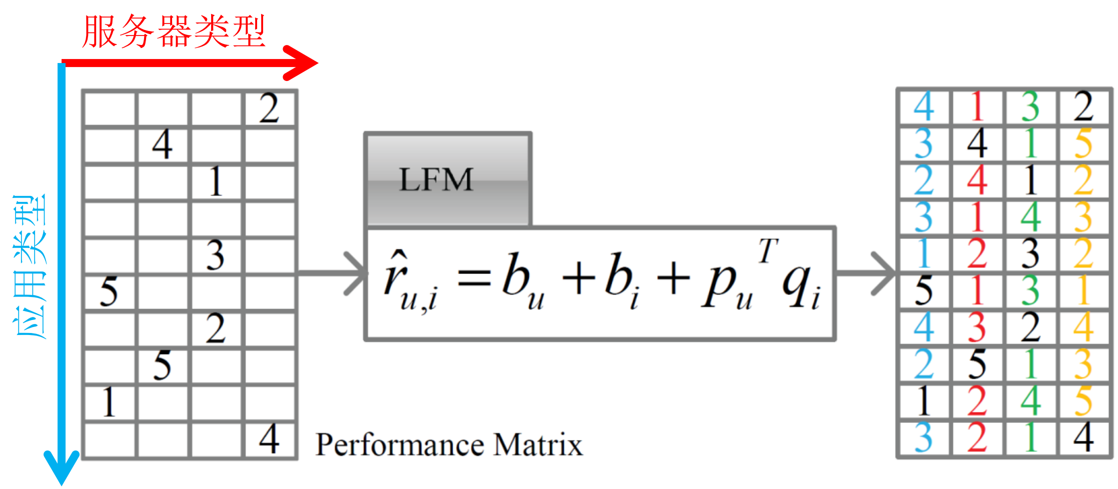
Figure 7: The non-zero elements are shown in color-filled, which denotes the performance logged in history

稀疏的性能矩阵，既历史性能的记录很有限，
线下Profiling的方法开销太大

利用矩阵分解预测性能



推荐系统
协同过滤
矩阵分解
NETFLIX
amazon.com
last.fm
用户A
用户B
用户C
物品A
物品A
物品A
喜欢
推荐
用户
物品
U
Σ
V



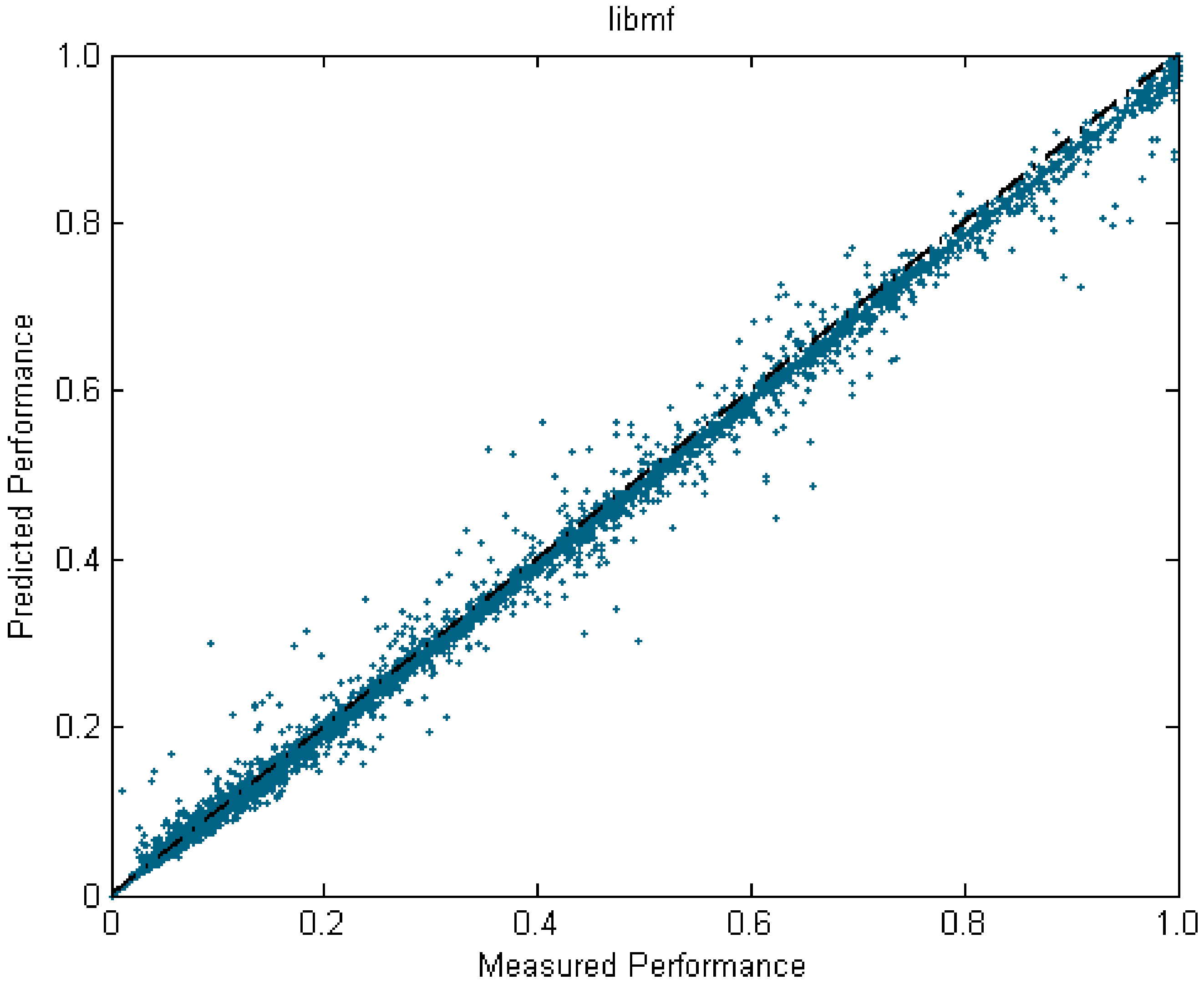
服务器类型
应用类型
LFM
$\hat{r}_{u,i} = b_u + b_i + p_u{}^T q_i$
Performance Matrix

内容目录

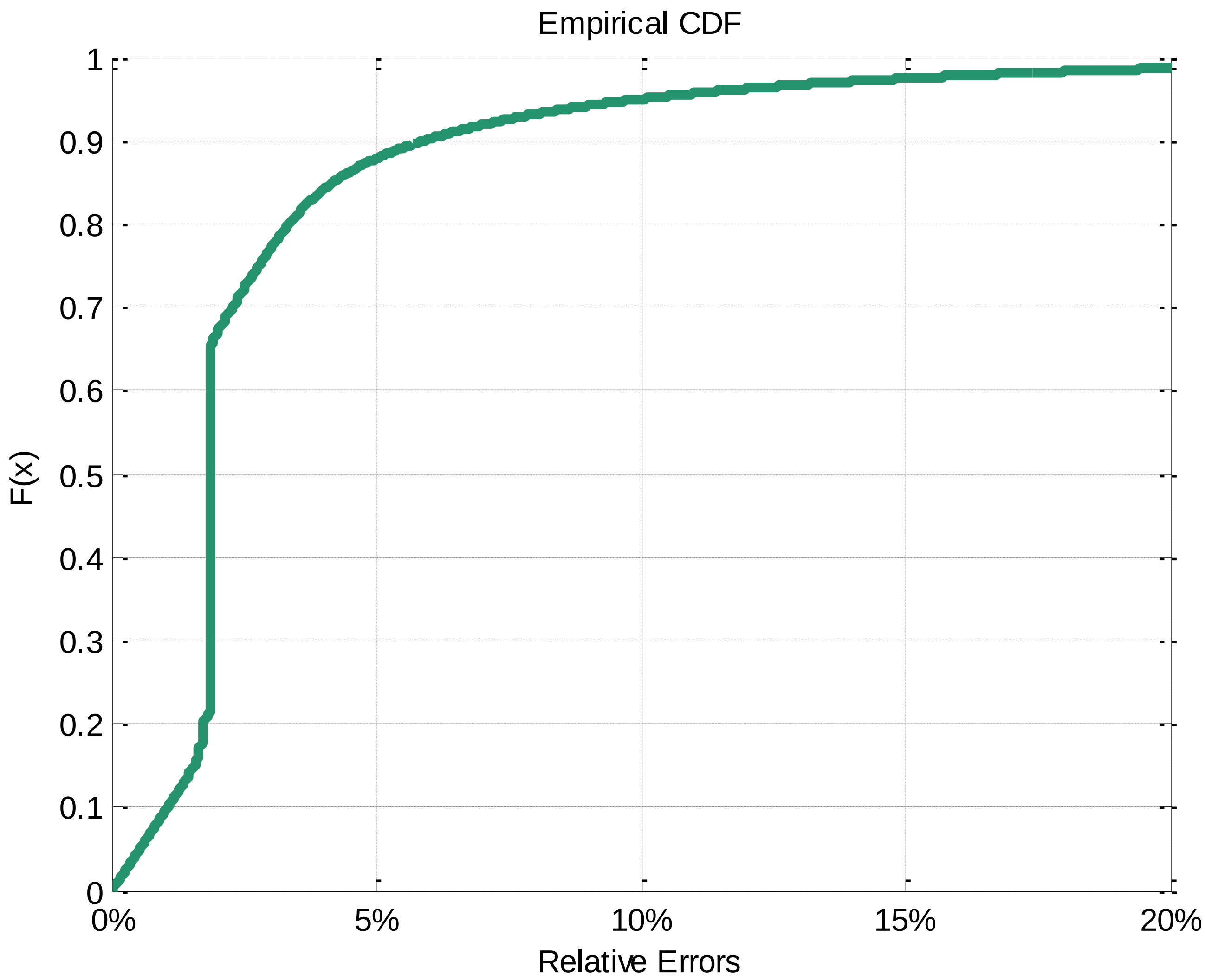
- 问题背景介绍
- 主要创新方案
- 具体实施方法
- 实验结果演示

内容目录

实验结果


libmf
Predicted Performance
Measured Performance
0
0.2
0.4
0.6
0.8
1.0

实验结果



Empirical CDF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
F(x)
0%
5%
10%
15%
20%
Relative Errors

实验结果

Rank	Iteration	Lambda	RMSE by Time
10	20	0.01	0.06096503939734827
10	20	0.005	0.06096646200447656
10	10	0.01	0.05909167674248318
10	10	0.005	0.06512720399301436
50	20	0.01	0.06230503201959108
50	20	0.005	0.07041228095049425
50	10	0.01	0.06360154839730821
50	10	0.005	0.08316303698435945

实验结果

Rank	Iteration	Lambda	RMSE by CPI
10	20	0.01	0.0014458431658637112
10	20	0.005	0.0014458431658637111
10	10	0.01	0.0014458431658637112
10	10	0.005	0.0014458431658637111
50	20	0.01	0.0014458431658637112
50	20	0.005	0.0014458431658637114
50	10	0.01	0.0014458431658637113
50	10	0.005	0.0014458431658637112

总结

- ♠ 线上挖掘大规模数据中心中Job和Machine所蕴含的决定性能的隐式特征，并以此预测性能
- ♠ 将挖掘人类对物品偏好的推荐系统模型映射到Job对Machine偏好的问题中
- ♠ 使用google trace，spark，scala验证我们的方法可以非常方便地应用于大规模数据中心资源管理领域，这蕴含着巨大的经济效益

Thanks for BigDataU & IBM Analytics