



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

SSE316：云计算技术 课程论文讲解

陈壮彬

软件工程学院

<https://zbchern.github.io/sse316.html>

课程论文目的



选择一个与云计算领域相关的研究问题，进行系统的学术界（5-10篇论文）和工业界调研，**全面了解云计算领域现有解决方案和最新技术水平。**

将调研结果整理成结构合理、格式正确的论文报告，以逻辑连贯的方式呈现研究结果，包括适当的参考和引用。

鼓励学生收集相关数据并进行实验，**得出有关问题和现有解决方案的结论，并对结果进行讨论**，以展示它们如何与研究课题和调研结果相关联。

课程论文内容



01

介绍和背景

介绍选题并说明该问题在云计算领域的重要性

方法

介绍调研开展的方法，包括调研的论文、研究方法、技术或工具、数据源等。

02

结果

呈现调研结果，包括现有解决方案的关联和比较，评估其在解决问题方面的效果，并提出可能的改进方法，提供学术界和工业界的观点。

03

总结

总结调研的关键发现

04

正确使用参考文献

正确引用研究中使用的所有资源（如论文、网站、图片），符合标准引文格式。

05

相关课题



- **资源管理 (resource management)**：如何调度云计算中的计算、存储和网络资源，以满足应用程序的需求。
- **智能运维 (AI for IT operations)**：如何通过异常检测和根因定位等手段确保云计算系统的可靠性和可用性，以保证用户数据和应用程序的完整性。
- **安全 (security)**：如何确保云计算系统的安全，以防止黑客入侵、数据泄露等。
- **隐私 (privacy)**：如何确保用户数据在云计算环境中的隐私。
- **数据管理 (data management)**：如何管理和存储云计算中的大量数据，以确保数据的完整性和可靠性。
- **效率 (efficiency)**：如何提高云计算系统的效率，以满足用户对资源的需求。
- **多云环境 (multi-cloud environments)**：如何支持多云环境下的应用程序，以提高云计算的可用性和灵活性。
- **混合云 (hybrid cloud)**：如何支持不同云服务提供商之间的数据和应用程序的互通性。
- **微服务 (microservices)**：如何构建和部署微服务架构，以提高云计算系统的可扩展性和可维护性。
- **边缘计算 (edge computing)**：如何在云计算和边缘计算之间进行数据和应用程序的协同。
- **虚拟化 (virtualization)**：如何利用虚拟化技术提高云计算系统的效率和可用性。
- **容器 (container)**：如何使用容器技术来管理云计算中的应用程序和数据。
- **计费 (cloud metering and billing)**：如何设计合理的云计算计费模型，以满足用户的需求。

例子1



- 资源管理 (Resource management)



云资源管理（Cloud Resource Management）是指对云环境中的资源进行有效的**规划、分配、监控和优化**的过程，以便为用户提供**高质量、可靠和灵活**的云服务。

云资源管理的挑战



由于**云基础设施的规模**以及云系统与大量用户之间**不可预测的交互**，云资源管理非常具有挑战性，需要复杂的决策和多目标优化决策。

外部因素

- ✓ 资源被超额订购
- ✓ 用户群体庞大，工作负载的类型和强度不可预测

内部因素

- ✓ 规模庞大，无法准确获得全局信息
- ✓ 系统软硬件的异构性
- ✓ 不同组件的故障率

为什么需要进行云资源管理？



成本优化

通过合理分配和调整资源使用，降低企业的云计算成本



实现弹性

通过动态调整资源分配以适应变化的工作负载，响应需求波动



提高可用性/可靠性

通过跨多个地域和可用区部署资源，确保在发生故障时，应用程序和数据可以继续运行



提高性能和安全性

确保工作负载运行在适当的资源上，并对资源进行监控和管理



● 资源弹性伸缩



● 资源调度算法

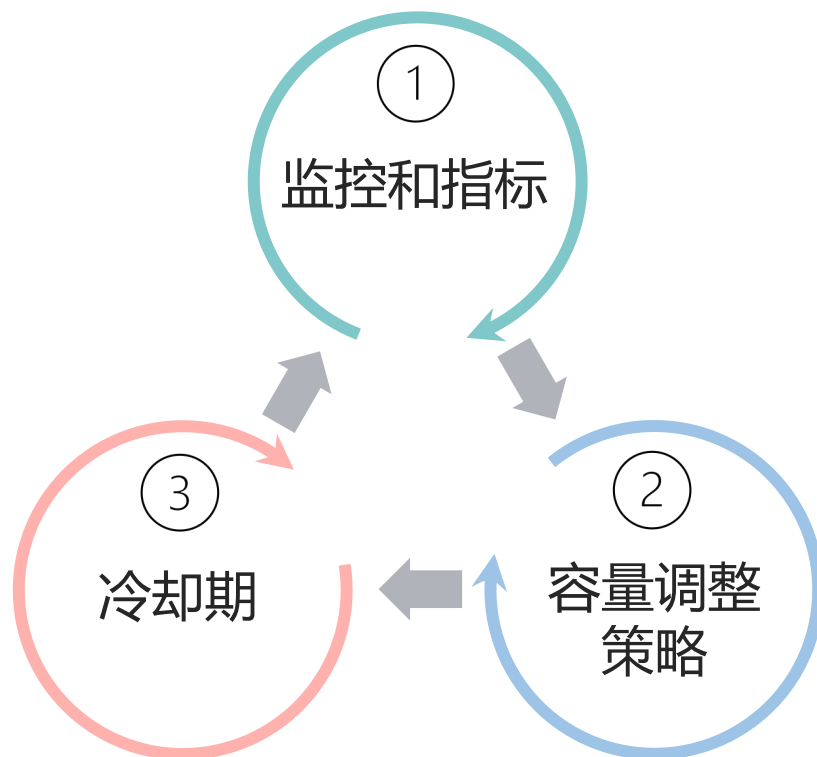
资源弹性伸缩



根据实际工作负载需求动态调整计算资源的数量。



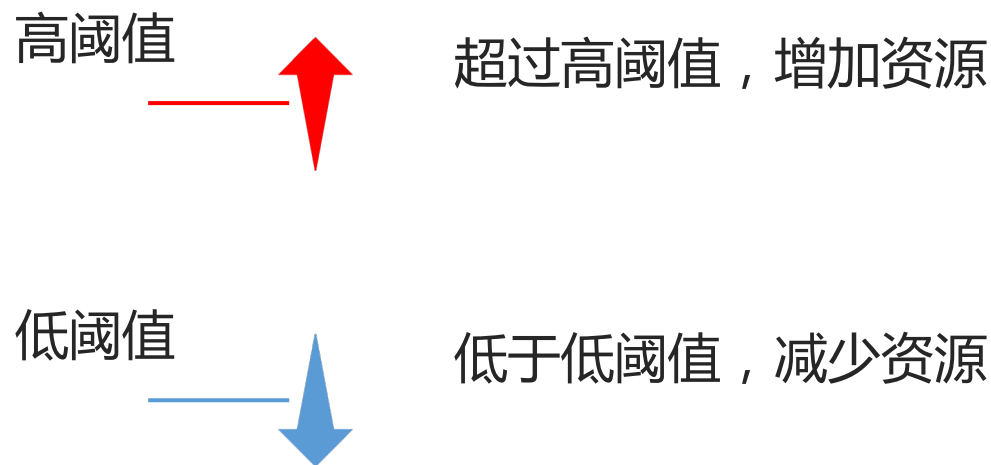
确保资源始终与需求匹配，**维持高性能同时降低资源浪费和成本。**



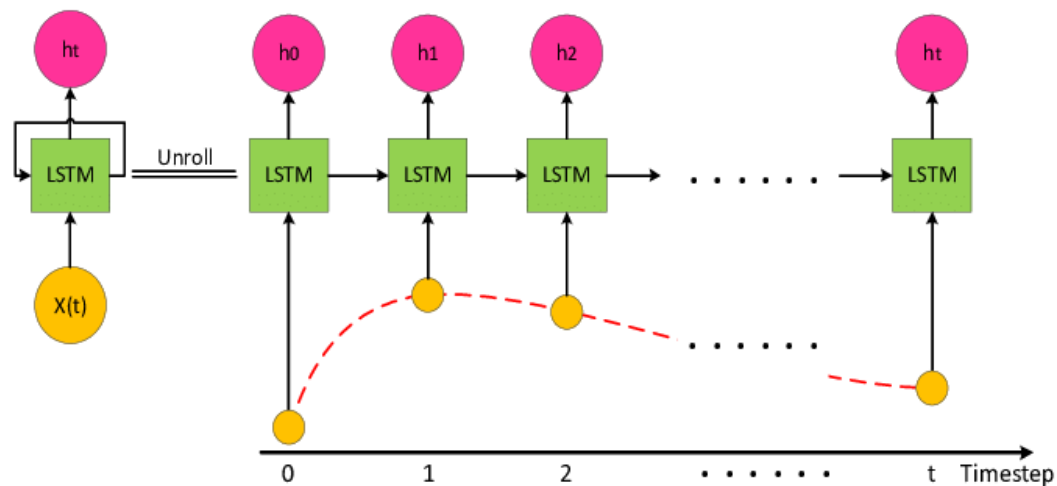
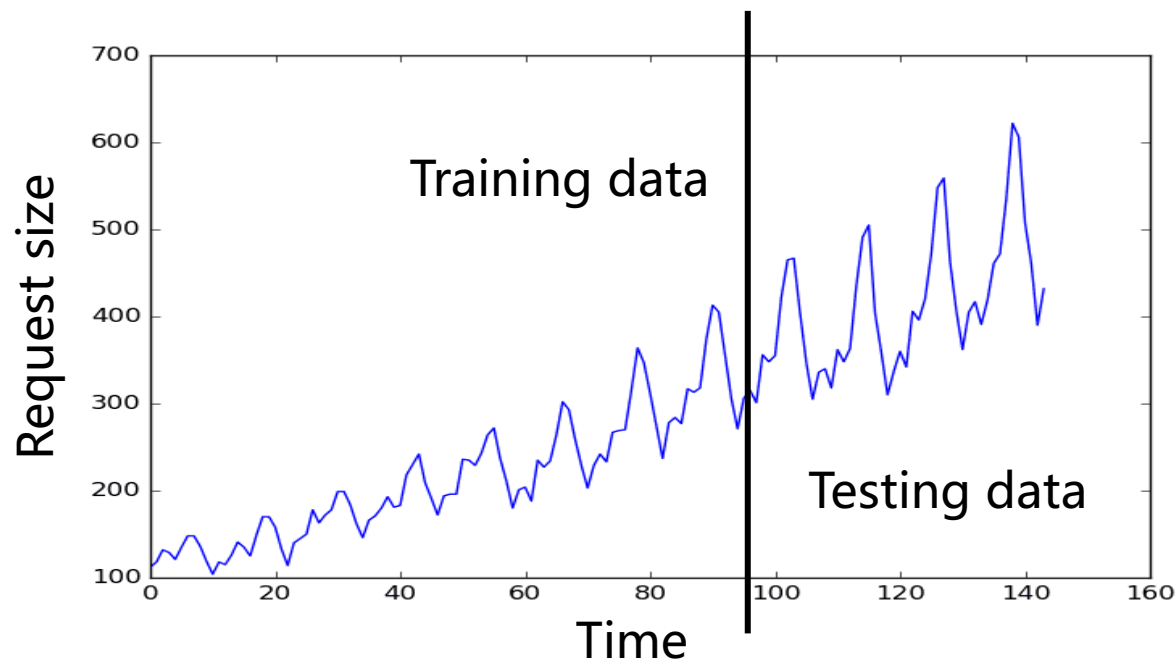
基于阈值的弹性伸缩



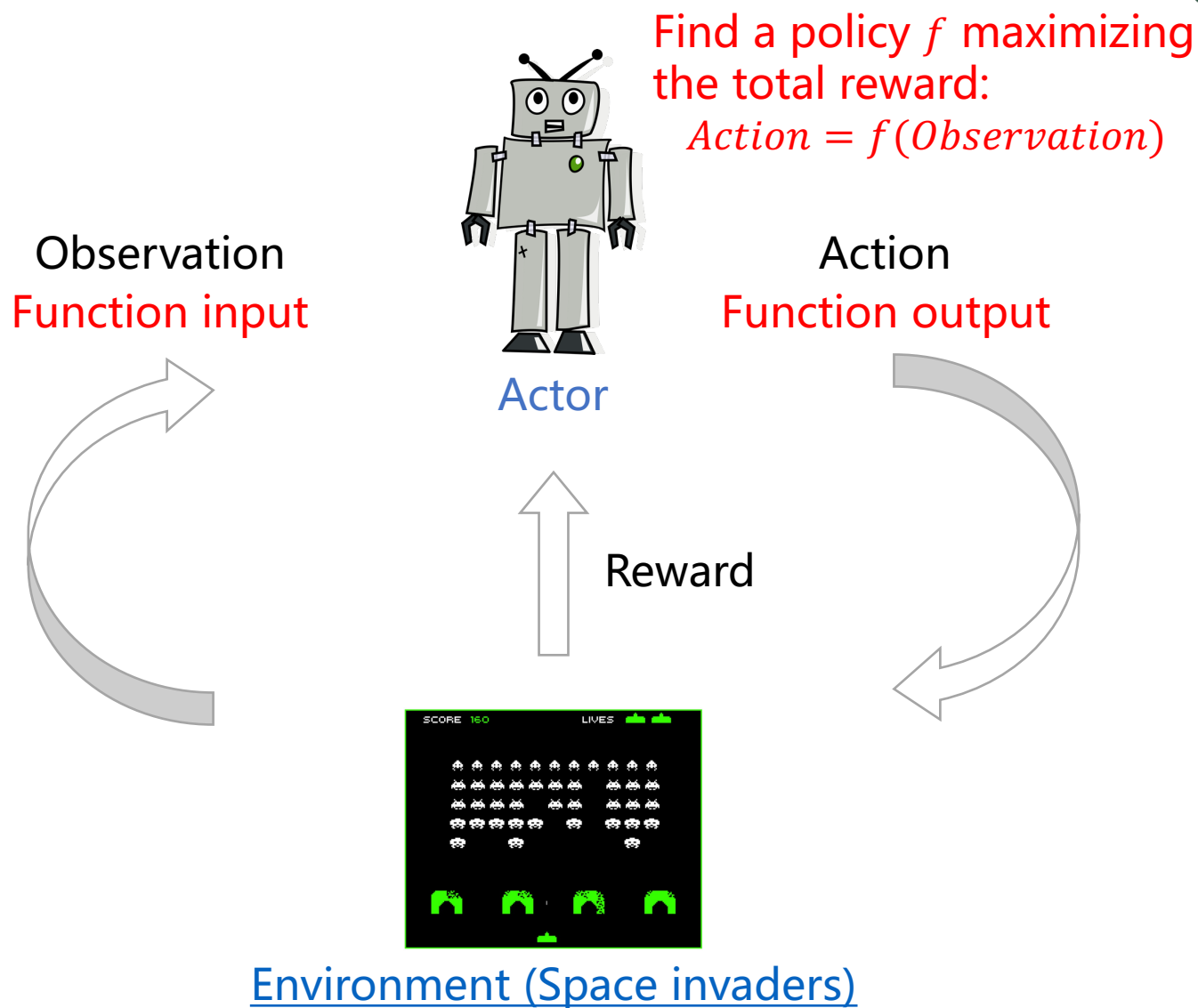
根据特定资源利用率指标（如 CPU 使用率、内存使用率、网络流量）及其相应阈值自动调整资源。



基于LSTM时序预测的弹性伸缩



基于强化学习的弹性伸缩



● 资源弹性伸缩

● 资源调度算法



传统处理器调度策略



- 先到先得 (First Come First Served)
- 最短任务优先 (Shortest Job First)
- 最短完成时间任务优先 (Shortest Time-to-Completion First)
- 时间片轮转 (Round Robin)

多租户模式



- 像MapReduce 和 Dryad 这样的集群计算系统最初是针对web索引等批处理作业进行优化的
- 然而，最近出现了另一个用例：在多个用户之间共享集群，这些用户在**公共数据集**上混合运行长批处理作业和短交互查询
- 当某个 slave 节点有空闲的 slot，master 节点会分配给某个 job 中的某个 task

如何在多租户之间调度集群
的计算资源？





Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling

Matei Zaharia

University of California, Berkeley
matei@berkeley.edu

Dhruba Borthakur

Facebook Inc
dhruba@facebook.com

Joydeep Sen Sarma

Facebook Inc
jssarma@facebook.com

Khaled Elmeleegy

Yahoo! Research
khaled@yahoo-inc.com

Scott Shenker

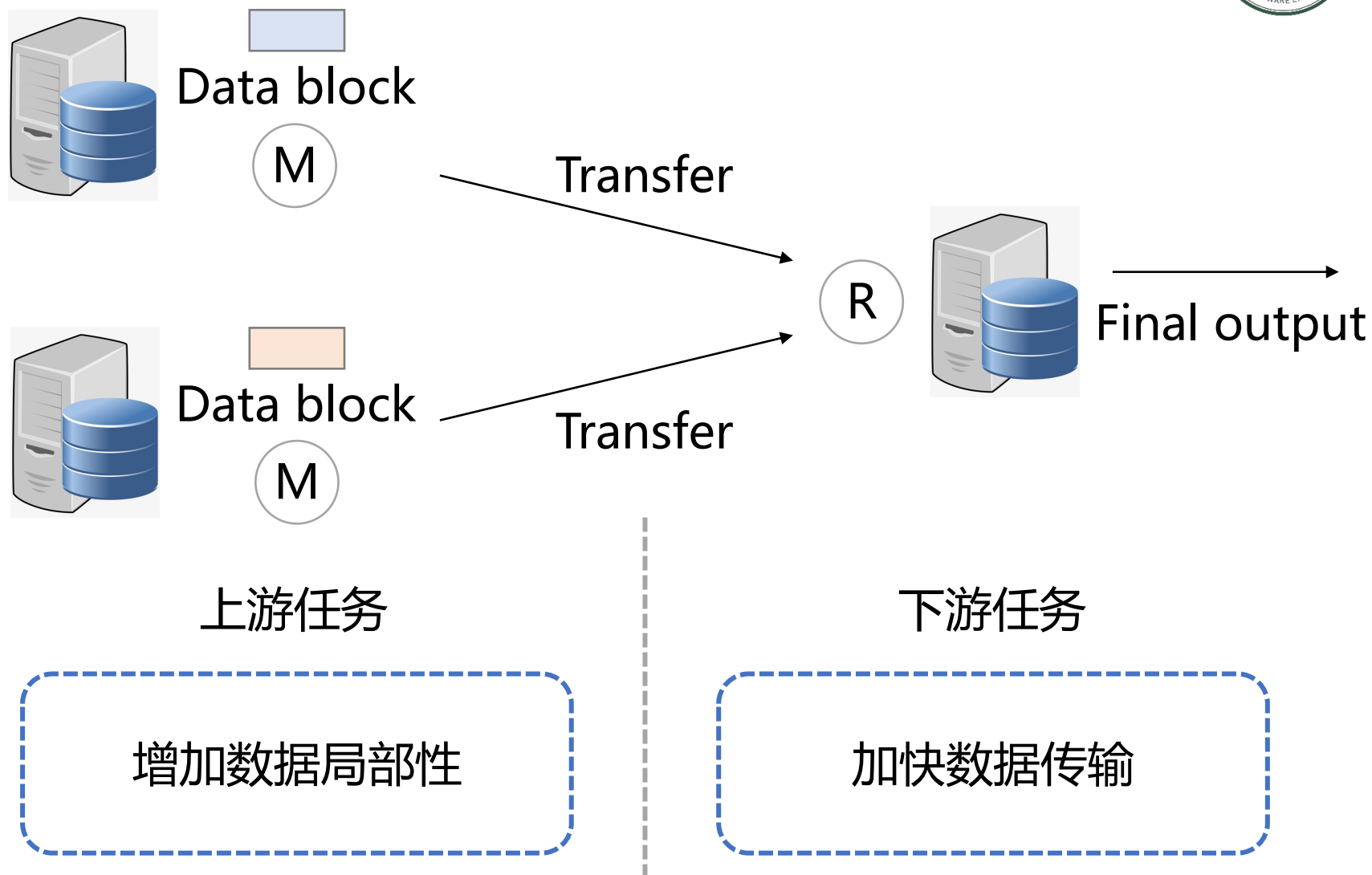
University of California, Berkeley
shenker@cs.berkeley.edu

Ion Stoica

University of California, Berkeley
istoica@cs.berkeley.edu

EuroSys 2010

如何让任务运行的更快？





The Power of Choice in Data-Aware Cluster Scheduling

Shivaram Venkataraman¹, Aurojit Panda¹, Ganesh Ananthanarayanan², Michael J. Franklin¹, Ion Stoica¹

¹*UC Berkeley* ²*Microsoft Research*

OSDI 2014

例子2



- 智能运维 (AI for operations)

智能运维的重要性



Revenue Loss

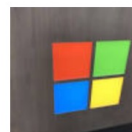


A Tiny Problem



User Dissatisfaction

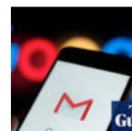
真实系统故障案例



Microsoft news recap: **Azure outage** problems explained ...

OnMSFT (blog) - 11 Apr 2020

Sit back, grab some coffee, and enjoy the read! Microsoft explains recent **Azure outage** problems in Europe due to "constrained capacity". Azure ...



Google **outage** hits Gmail, Snapchat and Nest

The Guardian - 8 Apr 2020

Google declared the **outage** resolved at 4:57pm BST. Big cloud providers such as Google Cloud Platform, **Amazon Web Services (AWS)** and ...



AWS cloud issues hit Sydney region

CRN Australia - 22 Jan 2020

#aws **outage** Sydney - Its been 3 hours already ... Anyone knows what's happening and recovery timeframe. Impacted ones include glue services ...

AWS suffers cloud problems in Sydney region

ITnews - 22 Jan 2020

日志在智能运维中的作用



- Development Practices

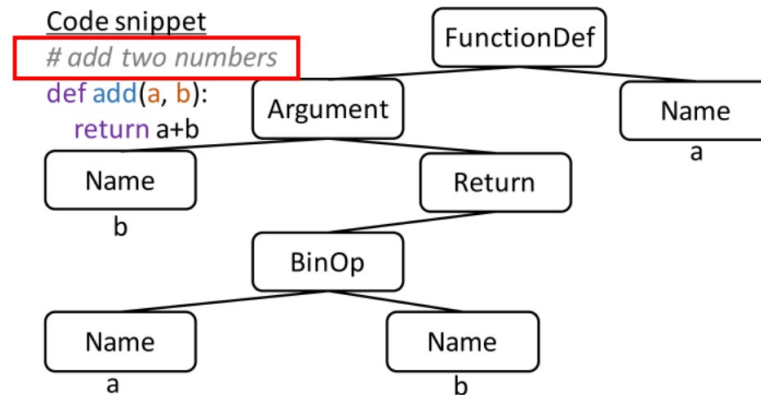
- *source code readability*, e.g., writing code comments

- Static Program Analysis

- *control-flow analysis*
- *data-flow analysis*
- *abstract interpretation*

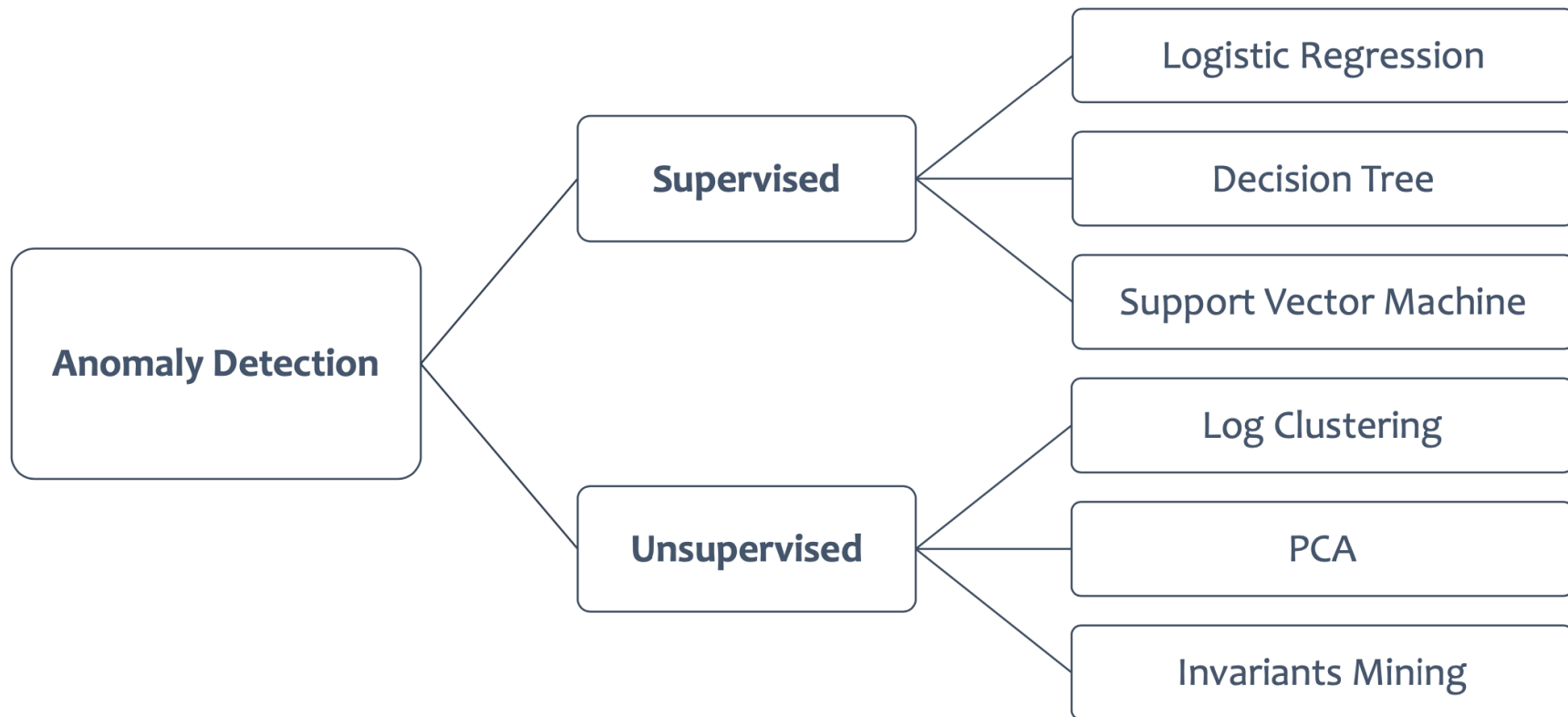
- Dynamic Program Analysis

- *testing*
- *program slicing*
- ***monitoring, e.g., logs***



```
1 | 2008-11-09 20:55:54 PacketResponder 0 for block blk_321 terminating  
2 | 2008-11-09 20:55:54 Received block blk_321 of size 67108864 from /10.251.195.70  
3 | 2008-11-09 20:55:54 PacketResponder 2 for block blk_321 terminating  
4 | 2008-11-09 20:55:54 Received block blk_321 of size 67108864 from /10.251.126.5  
5 | 2008-11-09 21:56:50 10.251.126.5:50010:Got exception while serving blk_321 to /10.251.127.243  
6 | 2008-11-10 03:58:04 Verification succeeded for blk_321  
7 | 2008-11-10 10:36:37 Deleting block blk_321 file /mnt/ hadoop/dfs/data/current/subdir1/blk_321  
8 | 2008-11-10 10:36:50 Deleting block blk_321 file /mnt/ hadoop/dfs/data/current/subdir51/blk_321
```


日志异常检测算法



- Datasets

System	#Time span	#Data size	#Log messages	#Anomalies
BGL	7 months	708 M	4,747,963	348,460
HDFS	38.7 hours	1.55 G	11,175,629	16,838



Time-stamp



Task-identifier

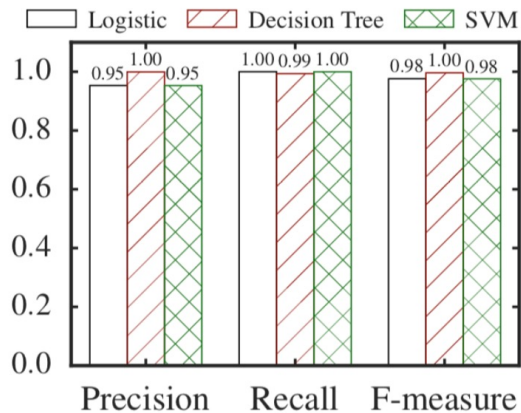
- Evaluation metric:

Precision / Recall / F1-Score

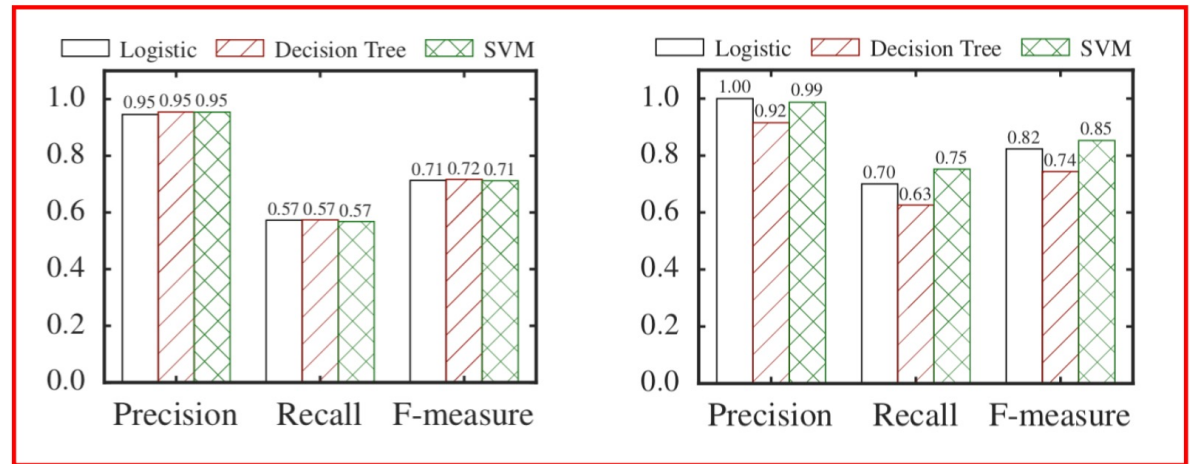
实验结果 (1)



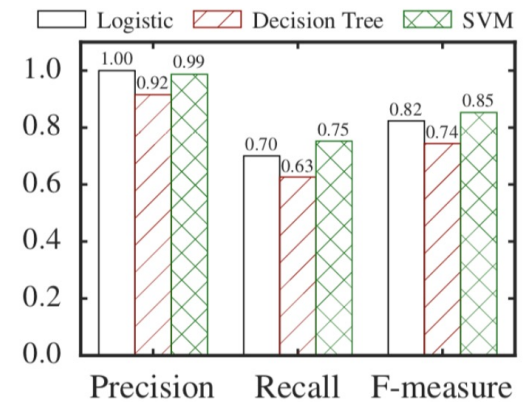
- Accuracy of supervised methods



HDFS



BGL (Fixed window)



BGL (Sliding window)

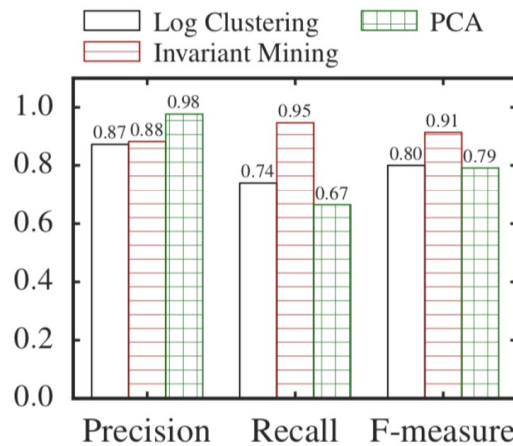
Finding 1: Supervised anomaly detection achieves **high precision**, while **recall varies**.

Finding 2: **Sliding windows** achieve higher accuracy than **fixed windows**.

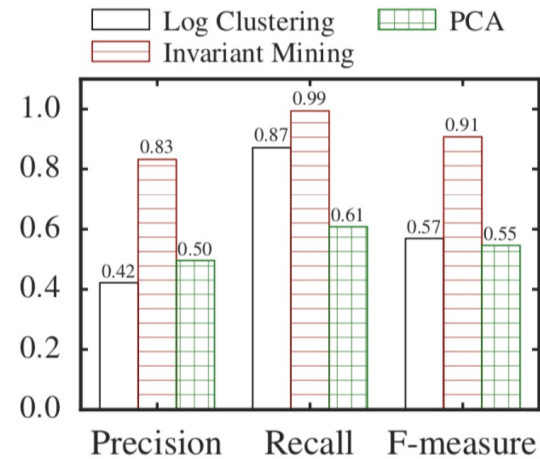
实验结果 (2)



- Accuracy of unsupervised methods



HDFS



BGL (Sliding window)

Finding 3: Unsupervised methods generally achieve inferior performance against supervised methods.



中山大學

SUN YAT-SEN UNIVERSITY

软件工程学院

SCHOOL OF SOFTWARE ENGINEERING

谢谢

陈壮彬

软件工程学院

<https://zbchern.github.io/sse316.html>