Performance Isolation and Fairness for Multi-Tenant Cloud Storage

David Shue, Michael J. Freedman, and Anees Shaikh

Mehmood Ahmad

Introduction

- Isolation
  - the separation of cloud service on per-tenant basis
  - making sure that each tenant can access the service at any time
  - billing is based on their usage of the service

Introduction

- Fairness
  - cloud tenants have even access to the cloud service
  - the resources in the cloud storage are allocated fairly across all tenants

Shared Storage in the Cloud

- Multiple tenants share same physical server and network infrastructure
- Use common platform services
- Examples: key-value stores, block storage volumes, SQL databases, message queues, notification services, etc

Shared Storage in the Cloud

- Two key issues:
  - Multi-tenant interference and unfairness
  - Variable and unpredictable performance

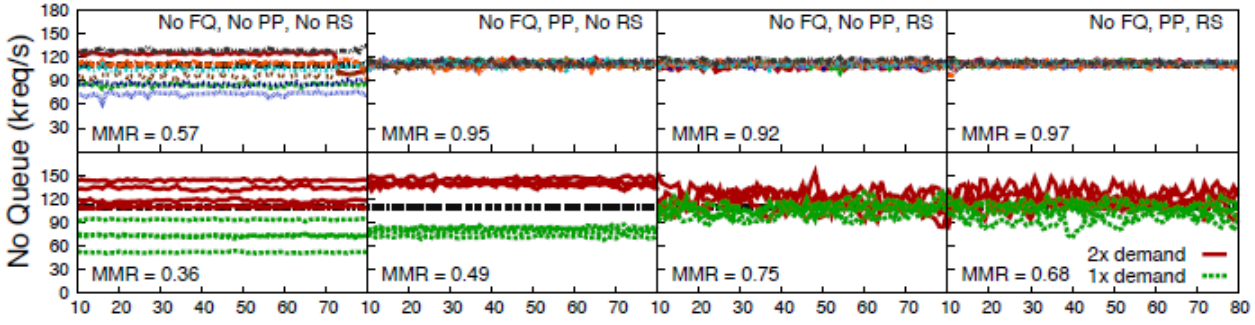
PISCES

- Partition Placement (PP)
  - involves the allocation of partitions to tenants in making sure that the load on each node does not exceed the rate capacity of that node
  - no node can shift its load to another, which may belong to another tenant
- Weight Allocation (WA)
  - making sure that the resources allocated to each tenant match their demands

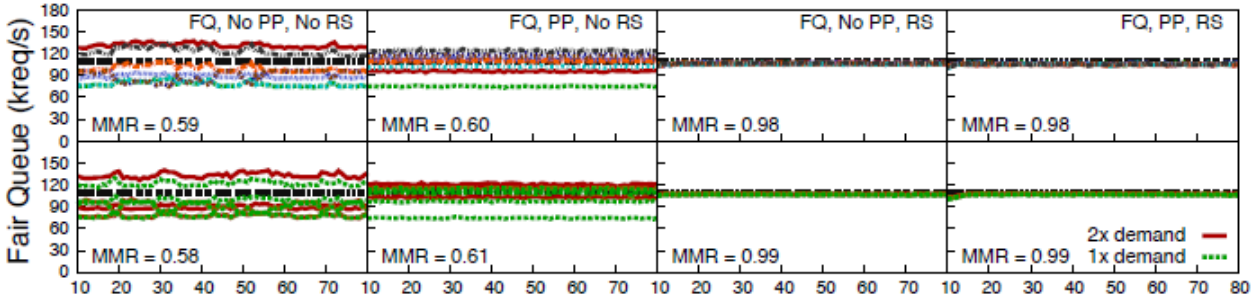
PISCES

- Replica Selection (RS)
  - using implicit feedbacks
  - saves both cost associated with storage and bandwidth
- Fair Queuing (FQ)
  - use of an algorithm known as deficit (weighted) round robin (DWRR)
  - to mediate the contention of resources between individual tenants and individual storage nodes belonging to the cloud service provider

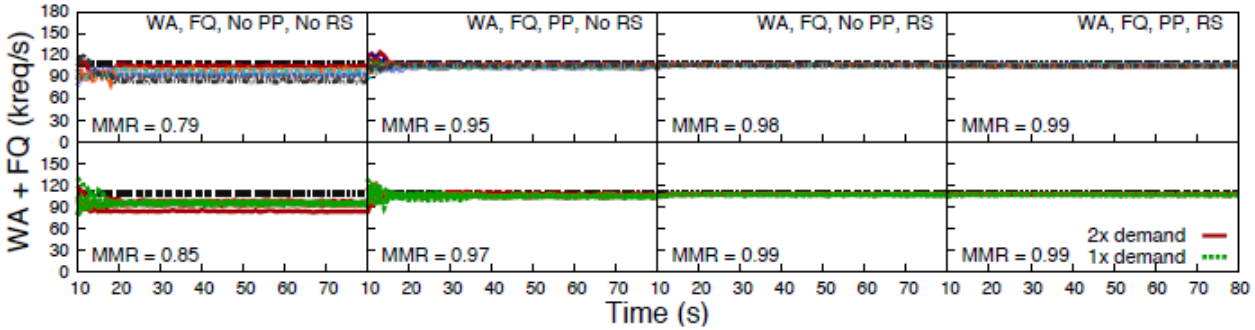
Evaluation/Results



(a) Membase (no queuing): Unfair unless partition placement or replica selection are enabled and provides no isolation.



(b) Fair Queuing (equal local weights): Provides strong isolation, but struggles with fairness unless replica selection is enabled.



(c) Weight Allocation + Fair Queuing: Achieves the fairness under all conditions, even under infeasible partition mappings (no PP).

Figure 5: System-wide fairness and isolation under a combination of Pisces mechanisms.

Strong Points

- ability of the system to achieve isolation in a multi-tenancy cloud environment
- provision of fair service to multiple users at the same time
- data is not only availed consistently but also safe and secure

Weak Points

- overall throughput is not exhaustively addressed
- capitalized on only one algorithm
- overheads have not been discussed extensively as a parameter of the overall performance

Conclusion

- Per-tenant max-min fair shares of system-wide resources
- min guarantees, high utilization
- *Arbitrary object popularity*
- *Different resource bottlenecks*

DynamoDB vs. PISCES

- Per-*tenant provisioned rates*
- *rate limited*
- *non-work conserving*
- *Uniform object popularity*

Questions/Discussion

- How important are the concepts of isolation and fairness in the modern world?
- DynamoDB or PISCES?
- Do you agree with the authors that cloud service is the future of computing and storage?
- How do you feel about privacy issues in cloud service?