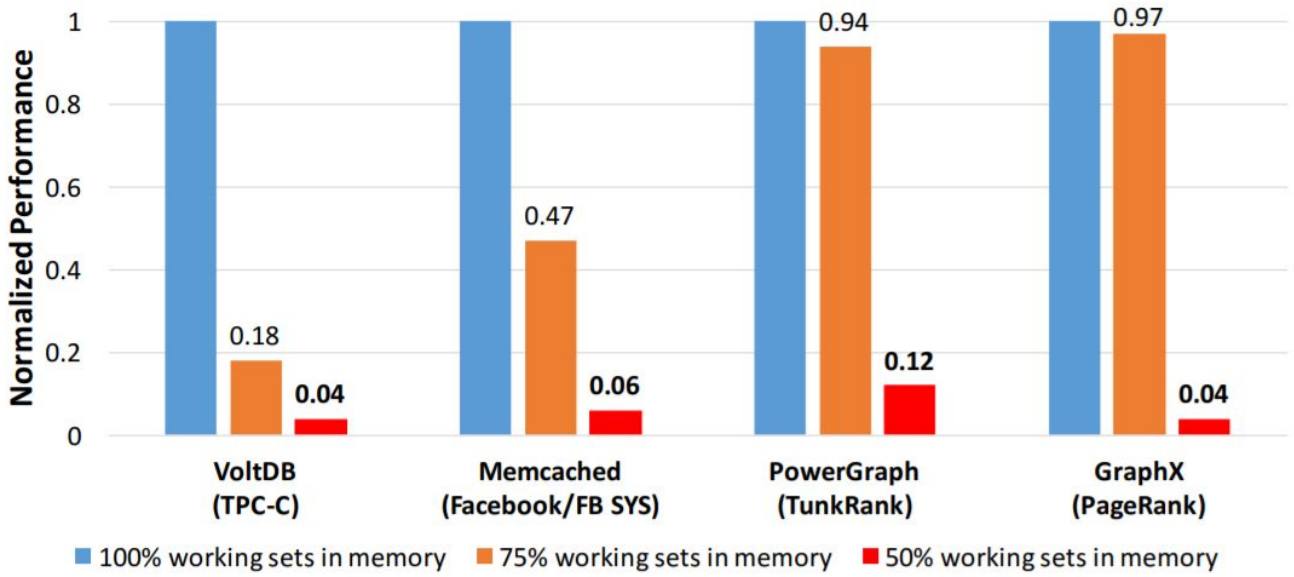
INFINISWAP

Peter Paquet, Wenting Tan

Some slides adapted from NSDI 2017 Infiniswap presentation by Jungchen Gu

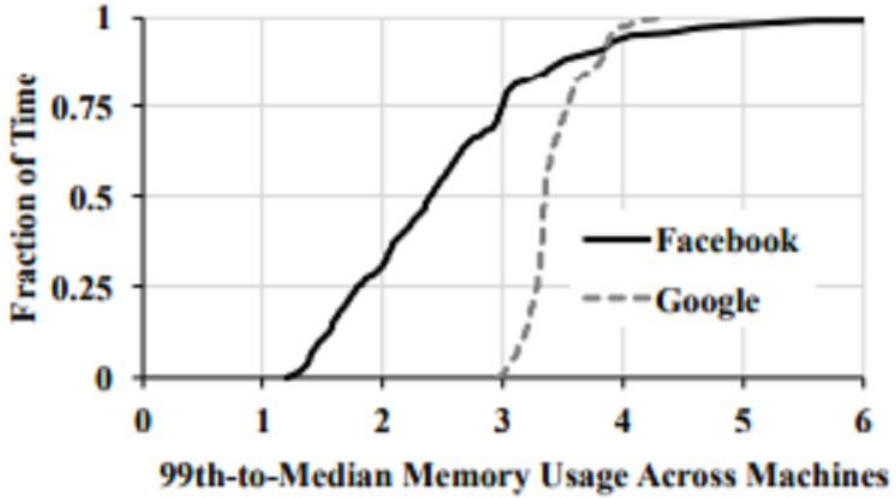
Background and Motivation

Performance Degradation



- Significant, nonlinear effect on performance
- Highlighted even more at tail latencies

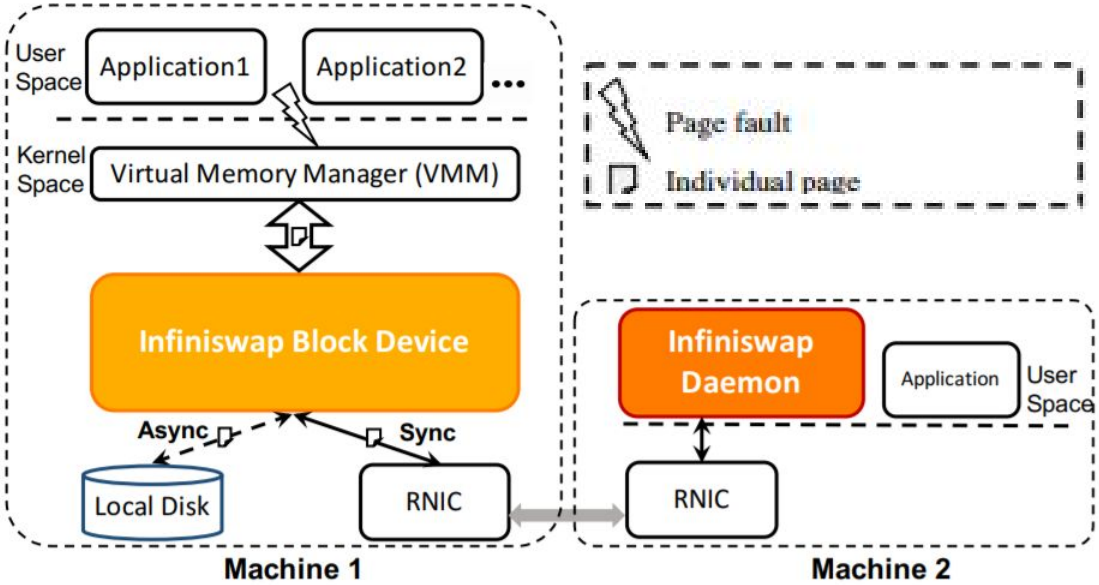
Characteristics of Memory Imbalance



- Memory usage substantially unbalanced across machines (short term)
- Memory utilization relatively stable on individual machines (short term)

Overview and Implementation

INFINISWAP System Architecture

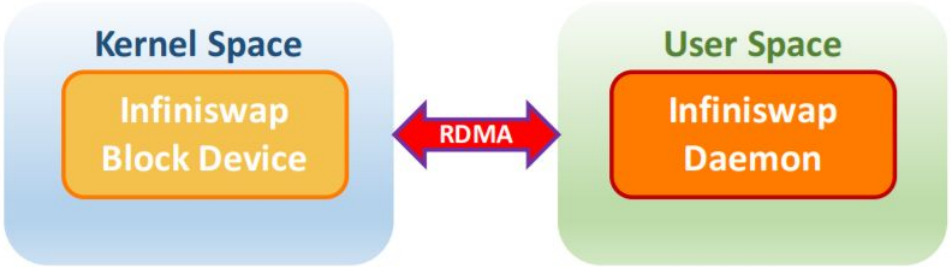


Goal: "*Efficiently expose all of a cluster's memory to user applications without any modifications to those applications or OSes of individual machines*"

Comparison to Recent Systems

	No HW design	No app modification	Fault-tolerance	Scalability
Memory Blade[ISCA'09]	✗	✓	✓	✓
HPBD[CLUSTER'05] / **NBDX**[1]	✓	✓	✗	✗
RDMA key-value service (e.g. HERD[SIGCOMM'14], FaRM[NSDI'14])	✓	✗	✓	✓
Intel Rack Scale Architecture (RSA)[2]	✗	✓	✓	✓
Infiniswap	✓	✓	✓	✓

Implementation



- **Connection Management**: One RDMA connection per active Block/Daemon pair
- **Control Plane:** SEND, RECV
- **Data Plane**: One-sided RDMA READ, WRITE

Block Device and Daemon

INFINISWAP Block Device

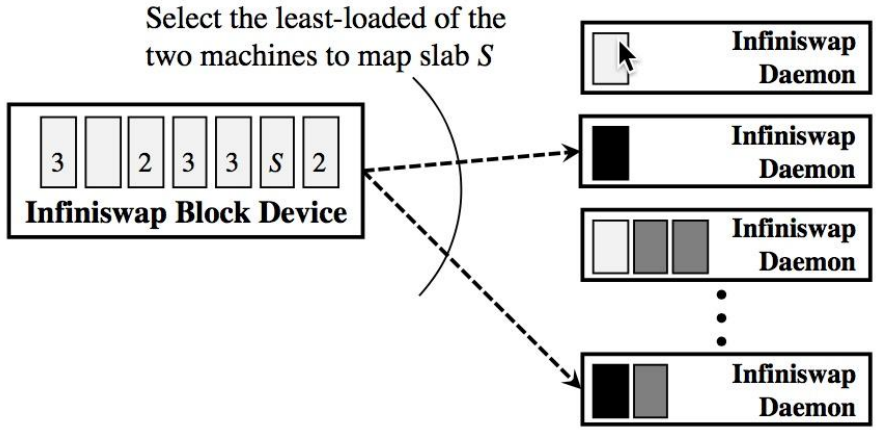
- **Efficient memory disaggregation**
- Slab management
- Remote Slab Placement
- I/O Pipelines
- Handling Slab Evictions
- Handling Remote Failures

Slab Management

- Divide address space into slabs of fixed size
- s := slab
- A(s) := total page-in and page-out activities
  - EWMA (Exponentially Weighted Moving Average)
- Above threshold, map slab to remote memory
  - RDMA WRITE
- Below threshold, remove slab from remote memory

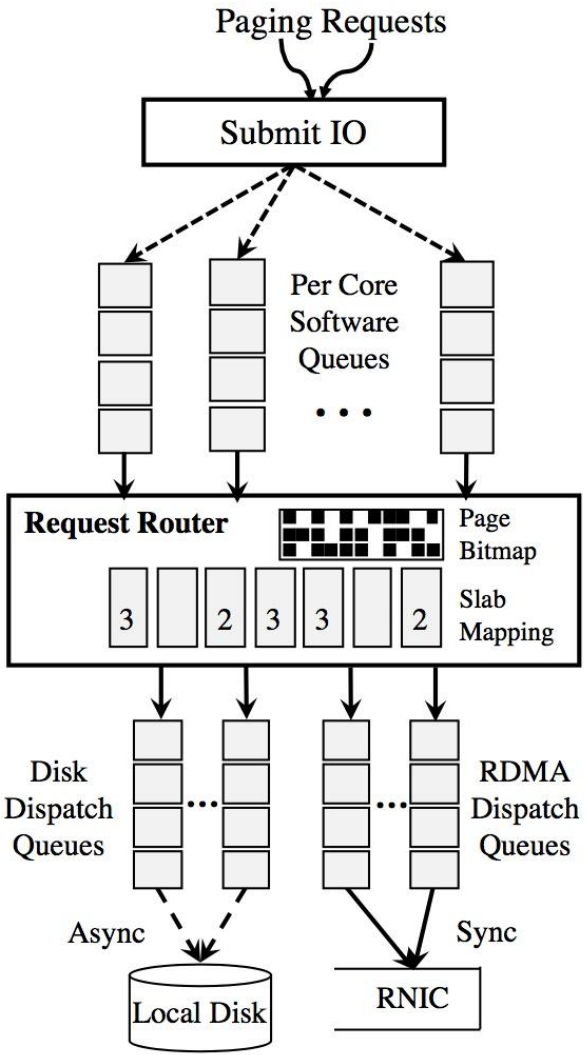
Remote Slab Placement

- Fault tolerance
- Central coordination
  - Increased mapping latency
- Uniformly random distribution
  - Unbalanced memory utilization
- Power of two choices
  - Divide remote machines into 2 sets
  - Select from 2 machines in a set



I/O Pipelines

- Multi-queue block IO
- Page Reads
- Page Writes
  - Unmapped slab
  - Mapped slab
    - Buffer
- Multi-Page Requests
  - VMM batch
  - Wait for all



INFINISWAP Block Device

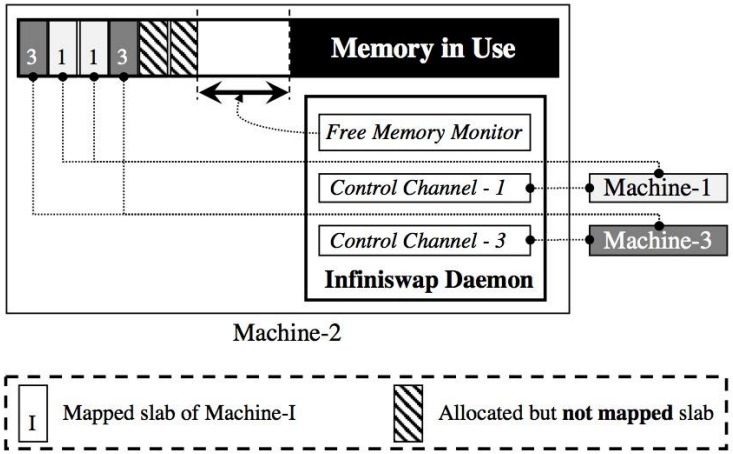
- Slab management
- Remote Slab Placement
- I/O Pipelines
- Handling Slab Evictions
  - Message from Daemon
- Handling Remote Failures
  - read-after-write

INFINISWAP Daemon

- **Claim memory on behalf of remote block device**
- **Reclaim memory on behalf of local applications**
- Memory Management
- Slab Eviction

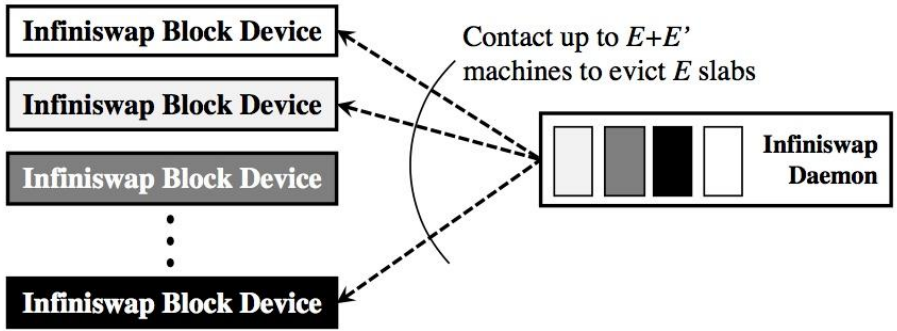
Memory Management

- U := machine memory usage
  - EWMA
- Below threshold, allocate slabs
- Above threshold, evict slabs



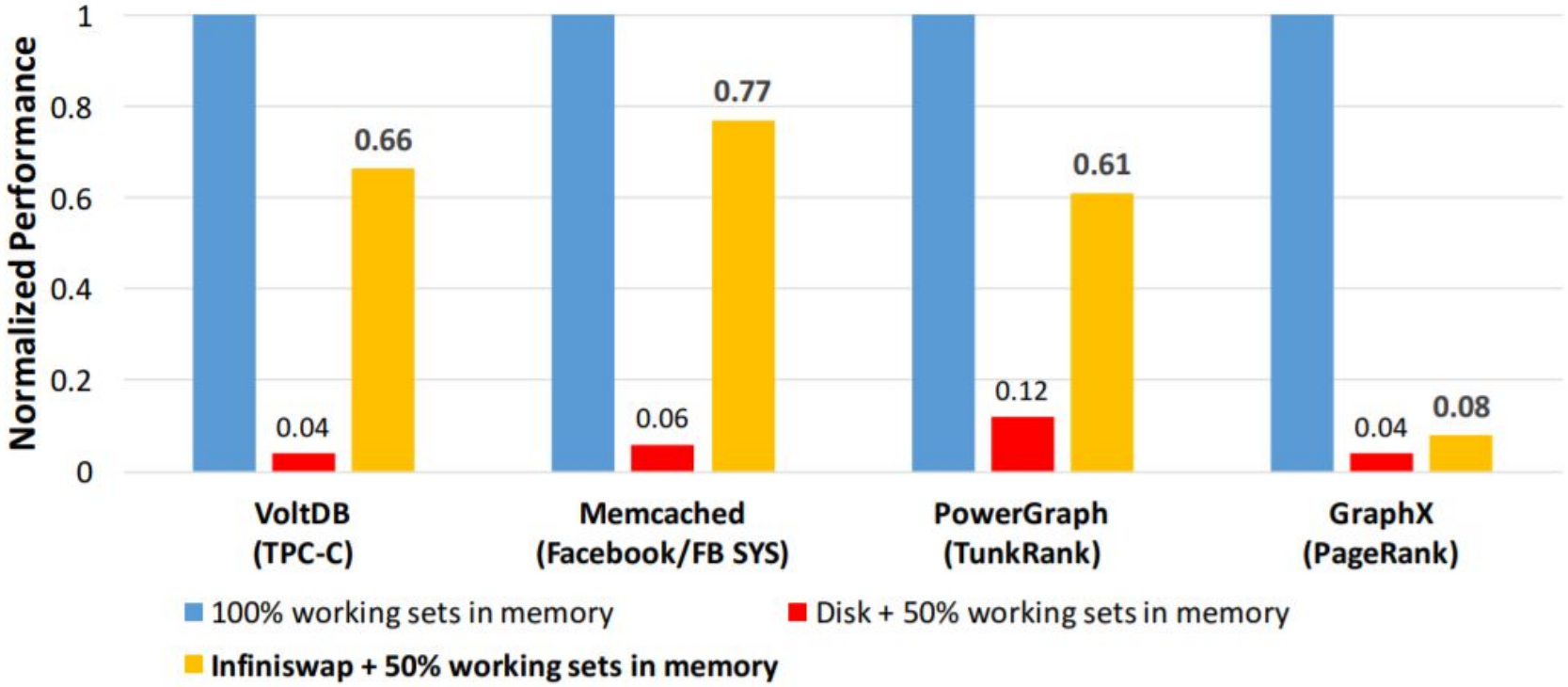
Slab Eviction

- Centralized control
  - Communication overhead
- Random choice
  - Evict busy slabs
- Power of multiple choices

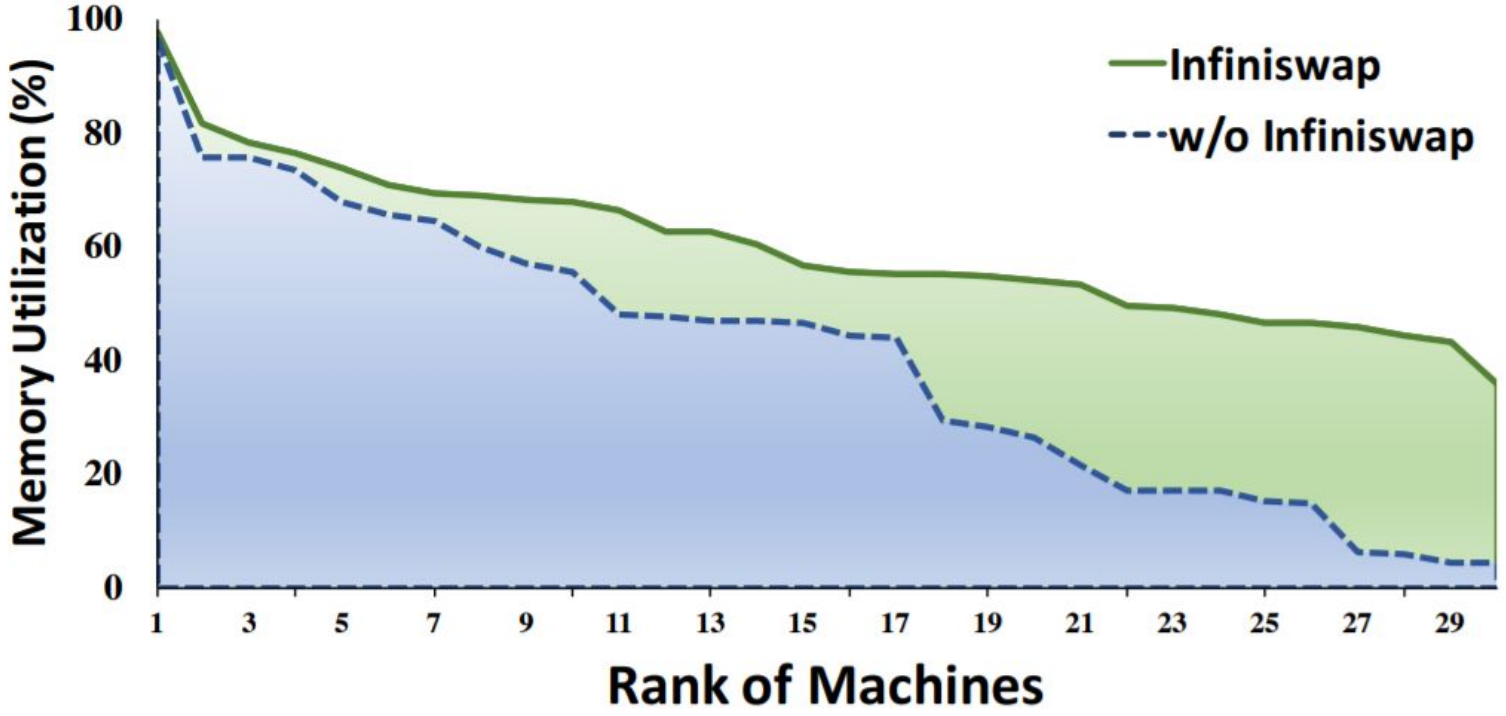


Evaluation

Performance Evaluation



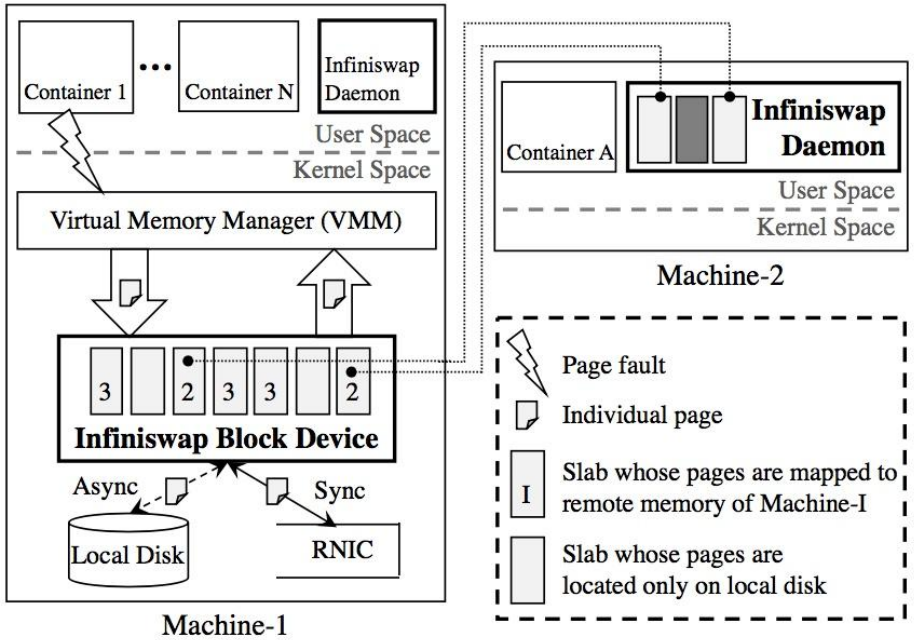
Cluster Memory Utilization



Conclusion

Limitations and Future Work

- Application-Aware Design
  - Memory pattern
  - Isolation/Differentiation
- OS-Aware Design
  - Swap overhead
- Fault-tolerance
  - Local disk bottleneck
  - Remote replicas
- Slab-size choice
- Network Contention/Bottleneck
- Spark Compatibility



Thanks for a Great Semester!