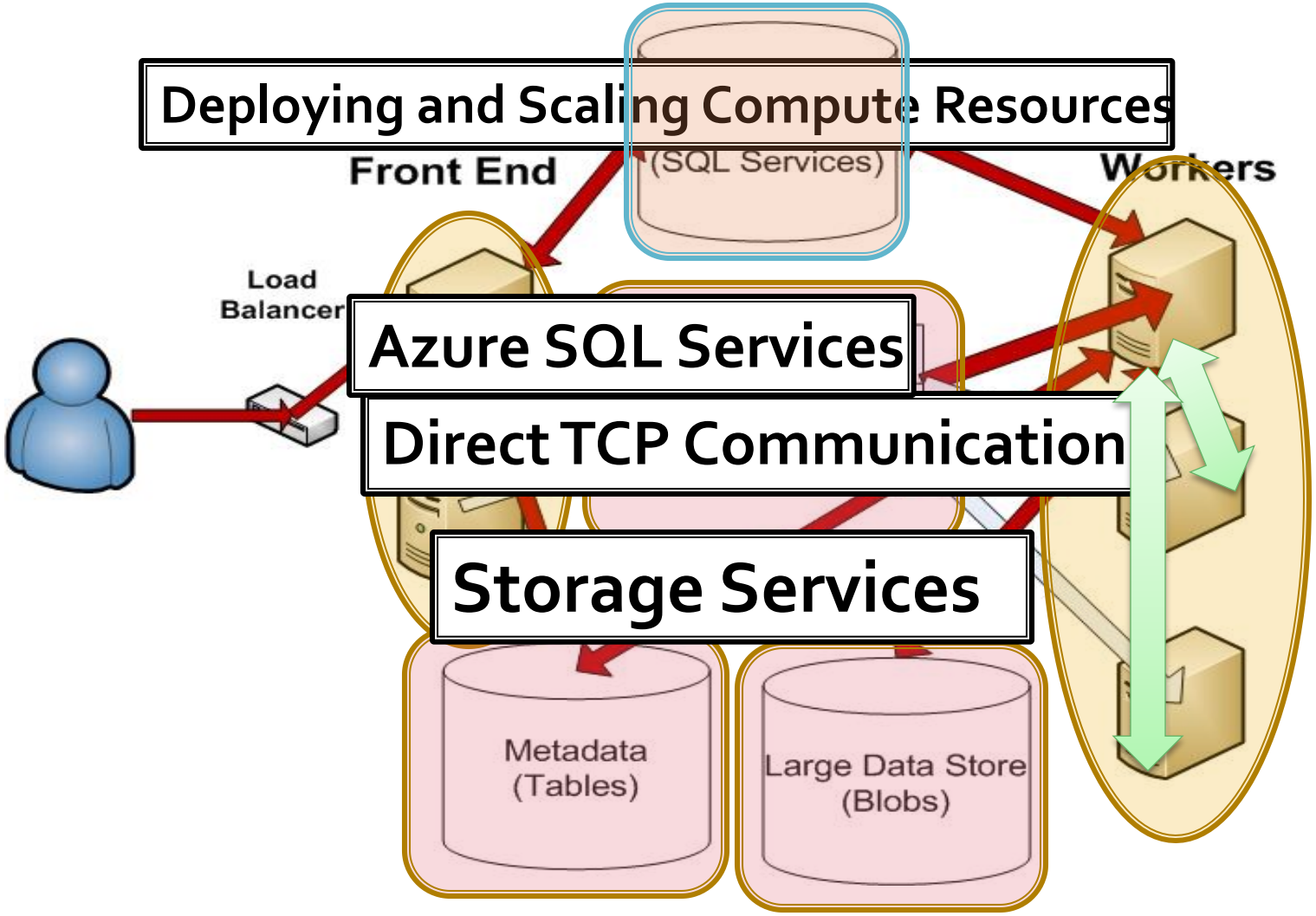
Zach Hill, Jie Li, Ming Mao, Arkaitz Ruiz-Alvarez, Marty Humphrey
Department of Computer Science, University of Virginia

Early Observations on the Performance of Windows Azure

Applications in Azure

- The question is not *can* I build my application for the cloud, it's *how* to do it well
- How will it perform?
- **Our focus**
  - **How do Azure services perform?**
- Experiments run between November, 2009 and February, 2010

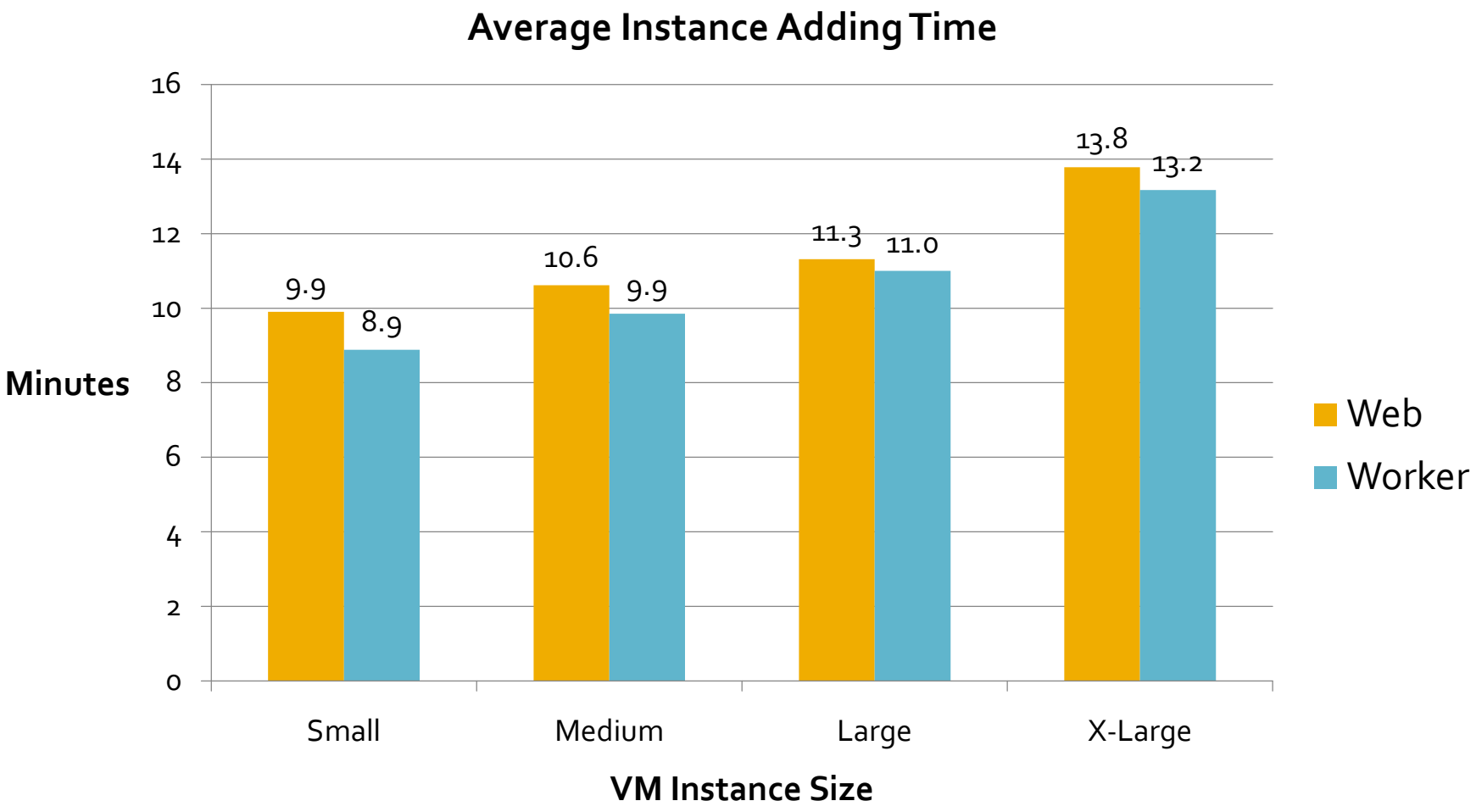
A Typical Application Architecture



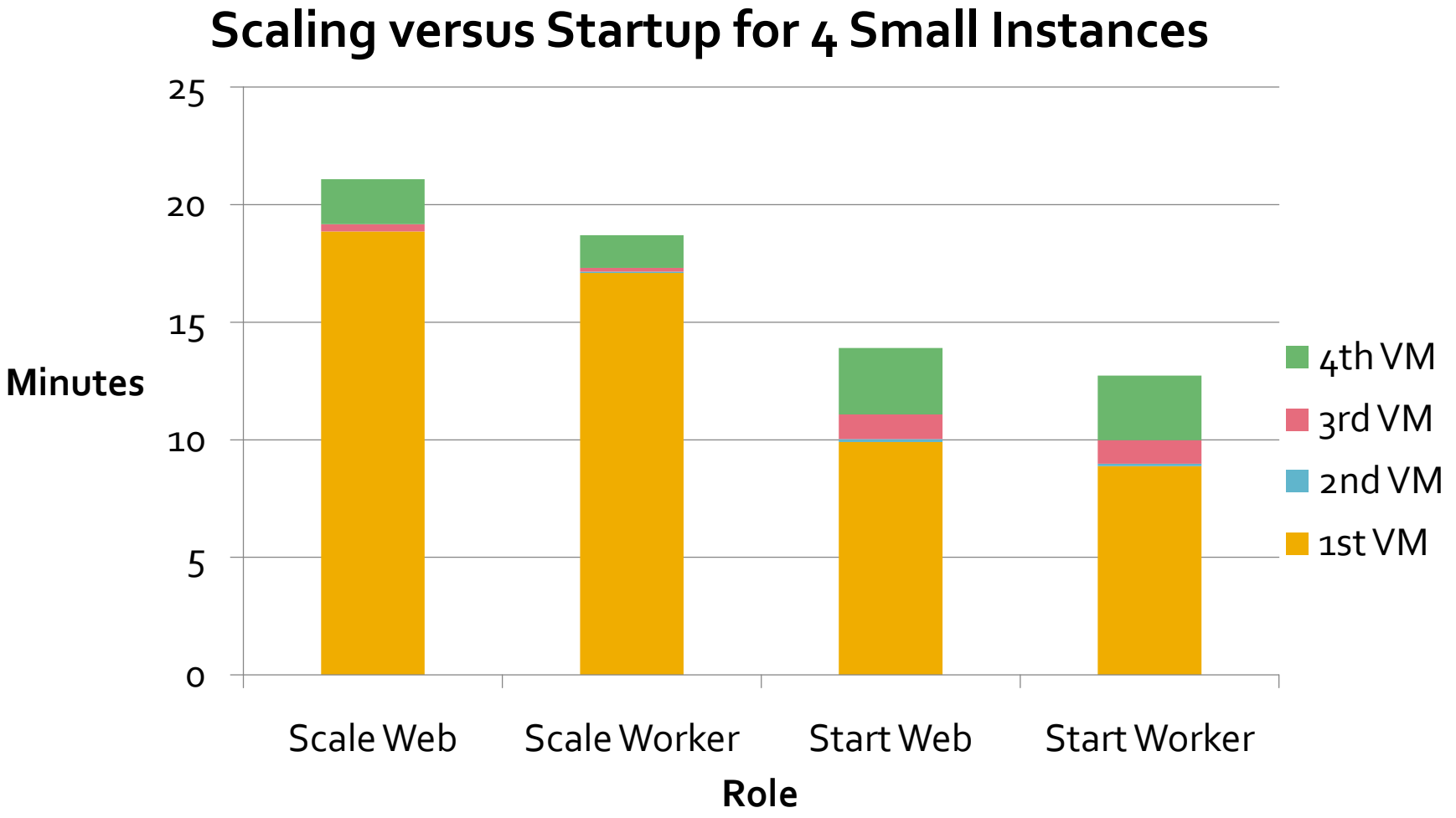
Deployment & Scaling Compute Resources

- Methodology
  - Application deployed from Azure Blob Storage
  - Deployment package <5MB
  - Measure time to start deployment (i.e. 4 small instances.)
  - Measure time to double instance count
  - Between Dec 17, 2009 and Jan 09 2010 we ran the experiment 431 times. Failure rate: 2.6%

Deploying: 1st VM Instance Startup time



Scaling: Adding Instances



Deployment & Scaling Takeaways

- Deploying a VM takes around 10 minutes—too long?
- Adding instances takes much longer than initial deployment—even worse
- Larger instance types take longer to start & web roles take longer than worker roles
- Not all instances will come online at the same time

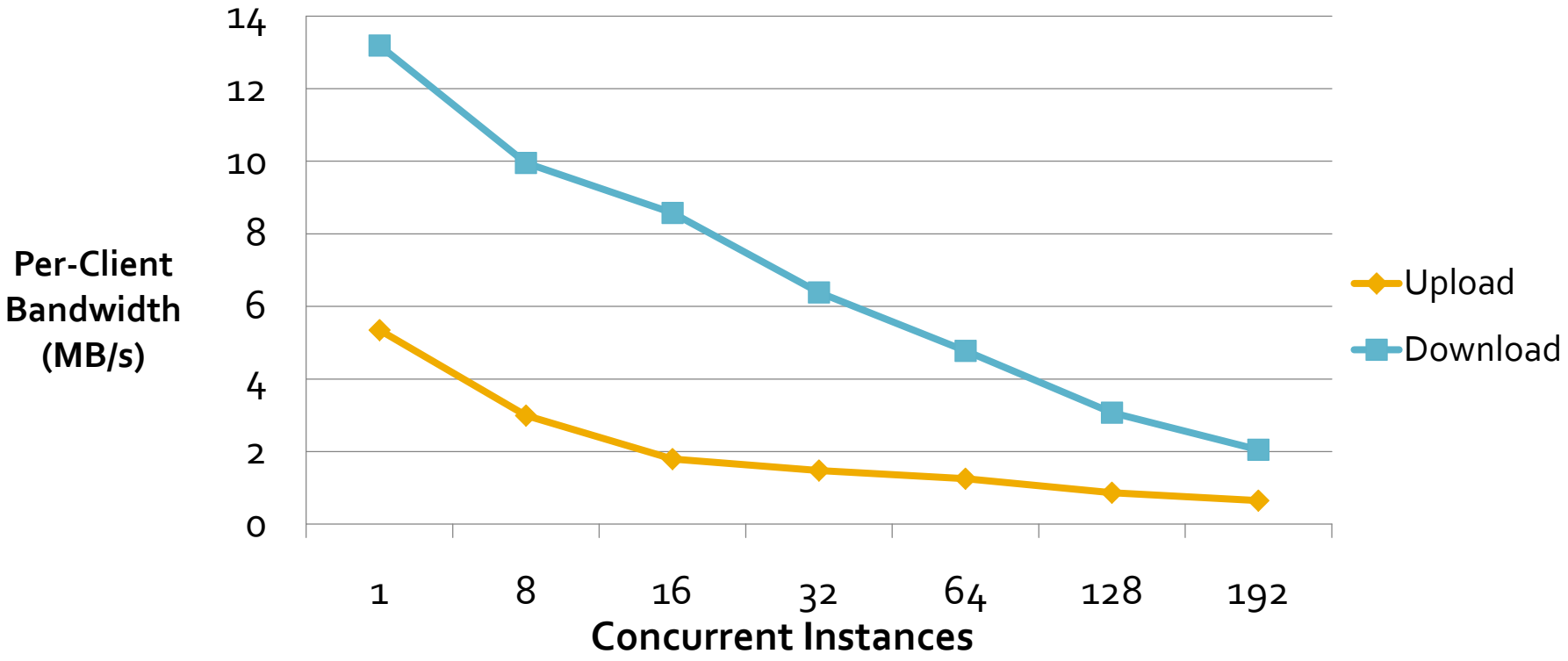
Windows Azure Storage Services

- Blobs – Large, unstructured storage
- Tables – Semi-structured data, queries, updates, inserts, deletes
- Queues – FIFO, asynchronous messaging

Windows Azure Blob Service

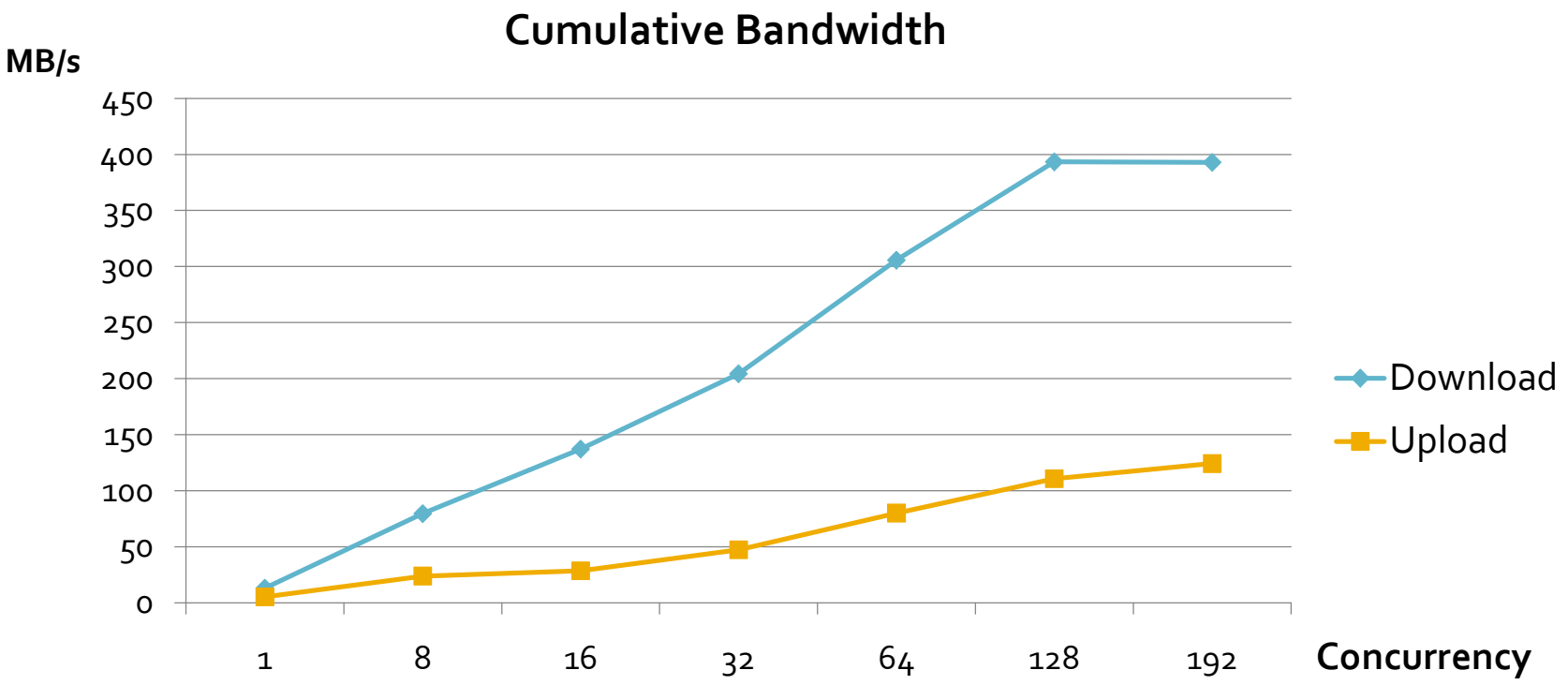
- Large object storage – 50GB or 1TB limit depending on type
- Get/Put semantics
- Performance isolated between blob containers
- Methodology:
  - Get a 1GB blob concurrently with 1 – 192 clients operating on the same blob
  - Put 1GB blobs concurrently into same container

Windows Azure Blob Performance at Client



- Download more than 2x upload speed
- Single, small client ~100Mb/s

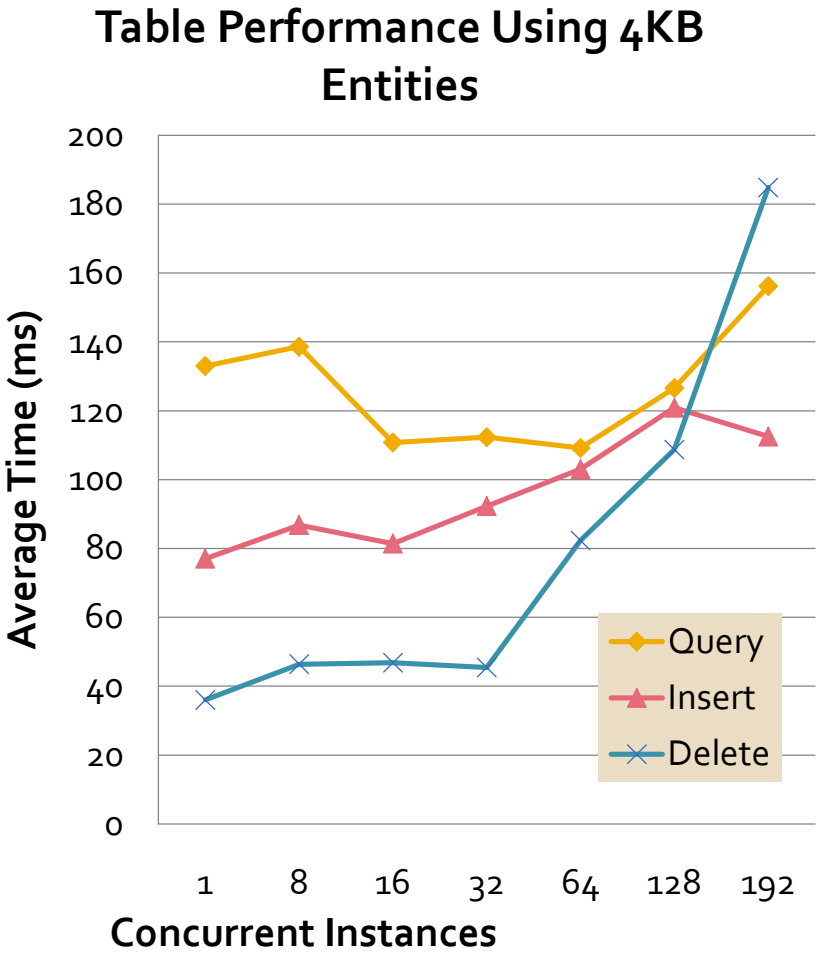
Windows Azure Blob Service Performance

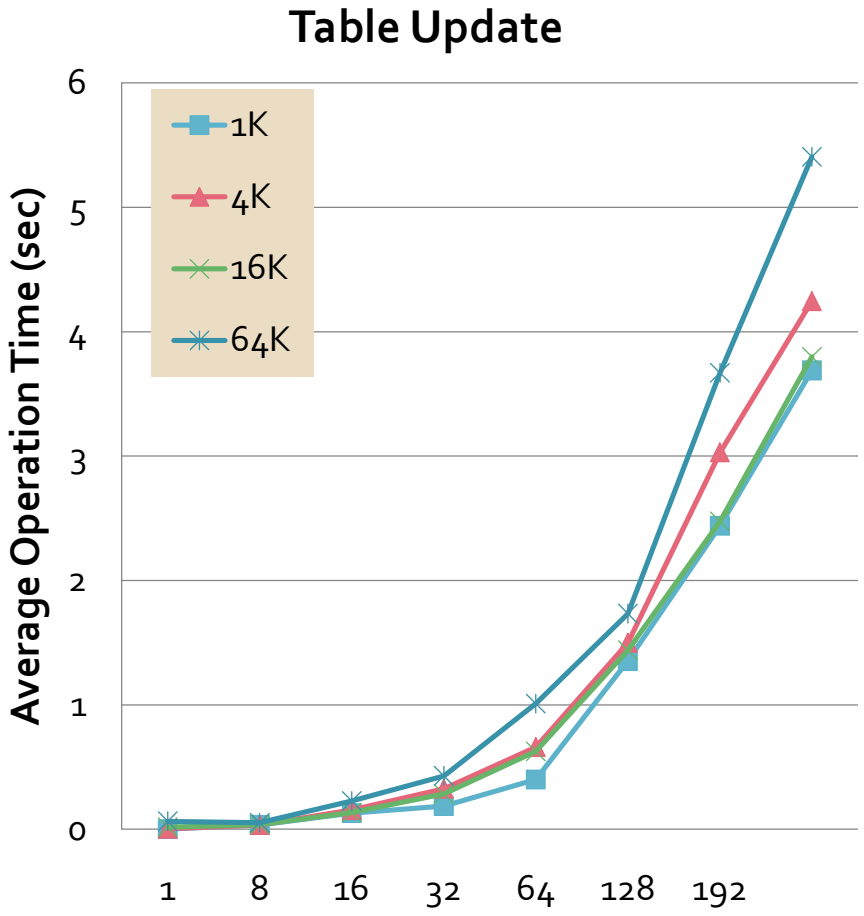


Windows Azure Table Service

- Entity, Attribute, Value model
- Semi-structured, no schema
- Methodology:
  - Perform 4 primary operations: insert, query, update, delete
  - Each client operates on unique entities (rows) within the same shared partition
  - Insert & Query & Delete: 500 ops/client
  - Update: 100 ops/client
  - ~220K entities in table for Query, Update, & Delete

Windows Azure Table Service Performance

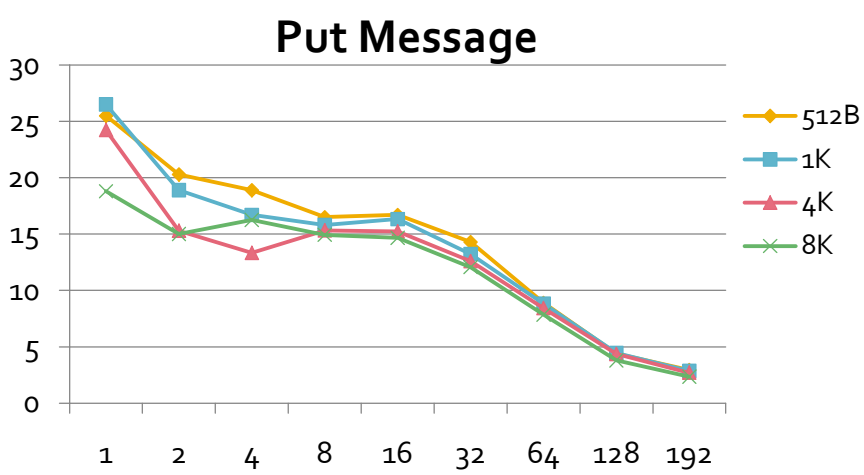


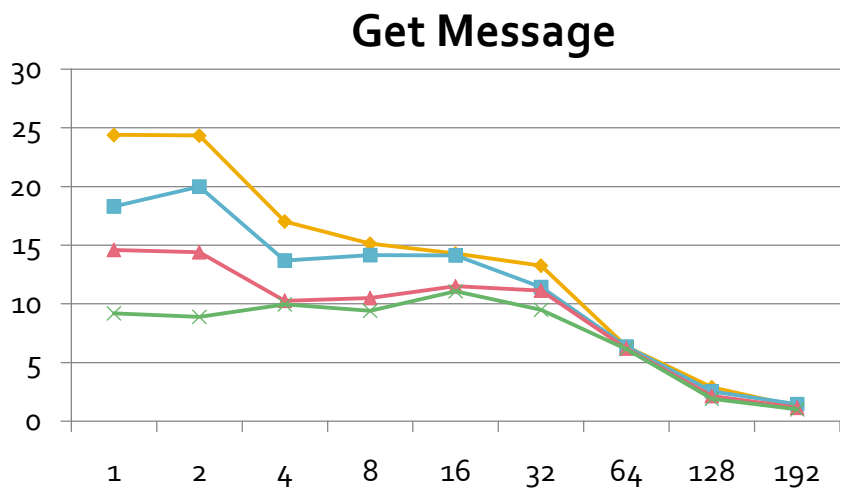


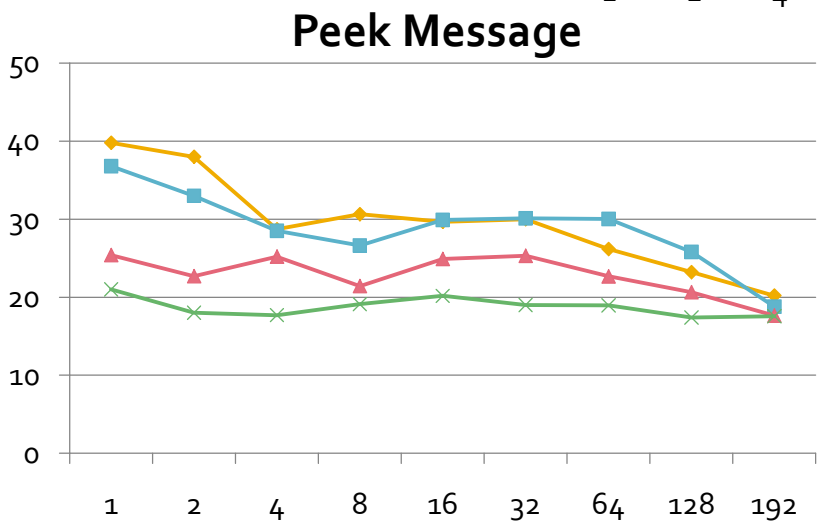
Windows Azure Queue Service

- Passing small (<=8K) messages in a FIFO style
- Get, Put, Peek operations
- Methodology: Single queue, concurrent clients get/put messages

Windows Azure Queue Service Performance







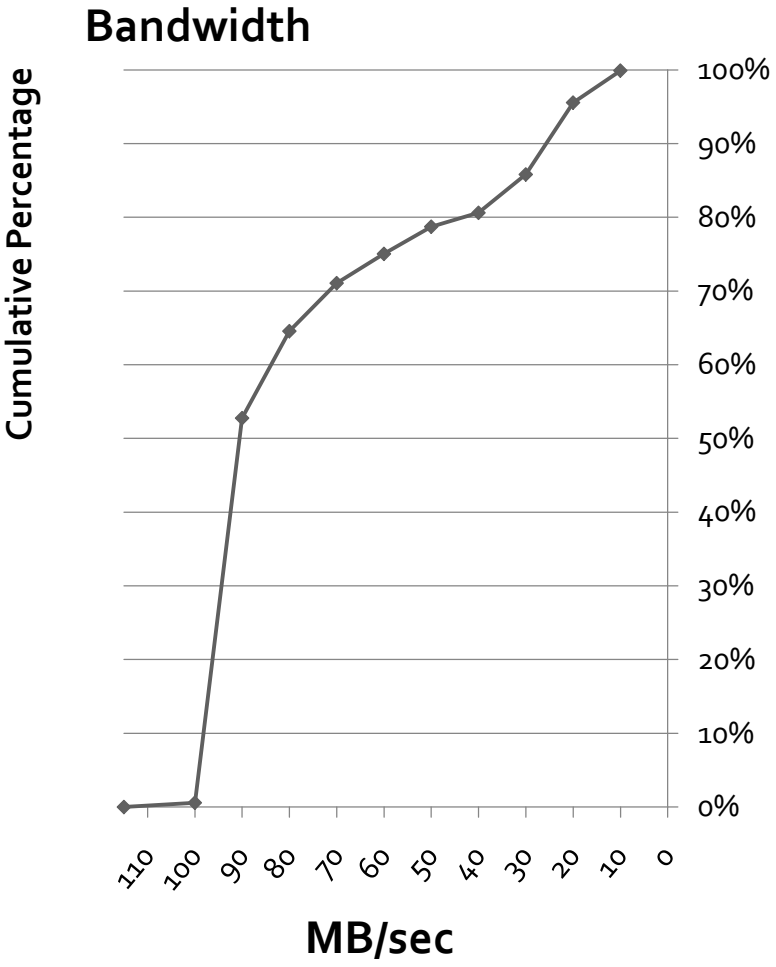
Vertical Axis: Msg/sec

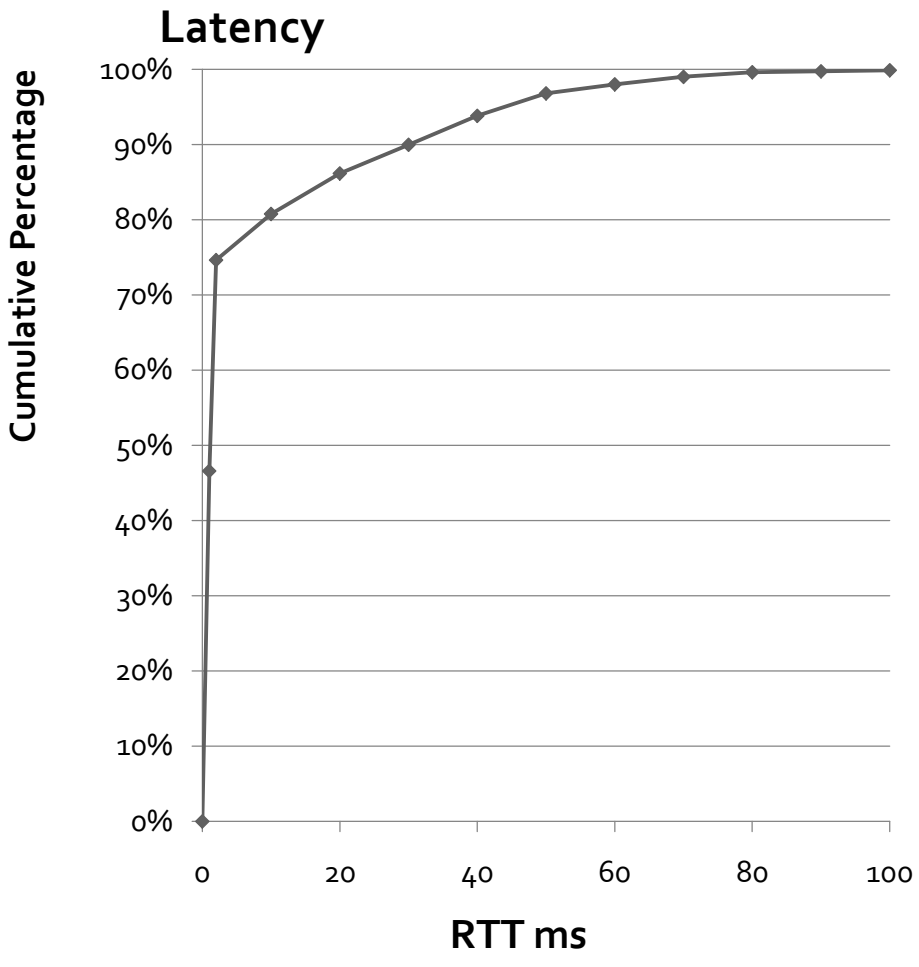
Horizontal Axis: Concurrent Instances

Direct TCP Communication

- TCP Endpoints allow Worker-to-Worker Role communication directly
- Offers a lower latency communication mechanism than message queues
  - No intermediary required

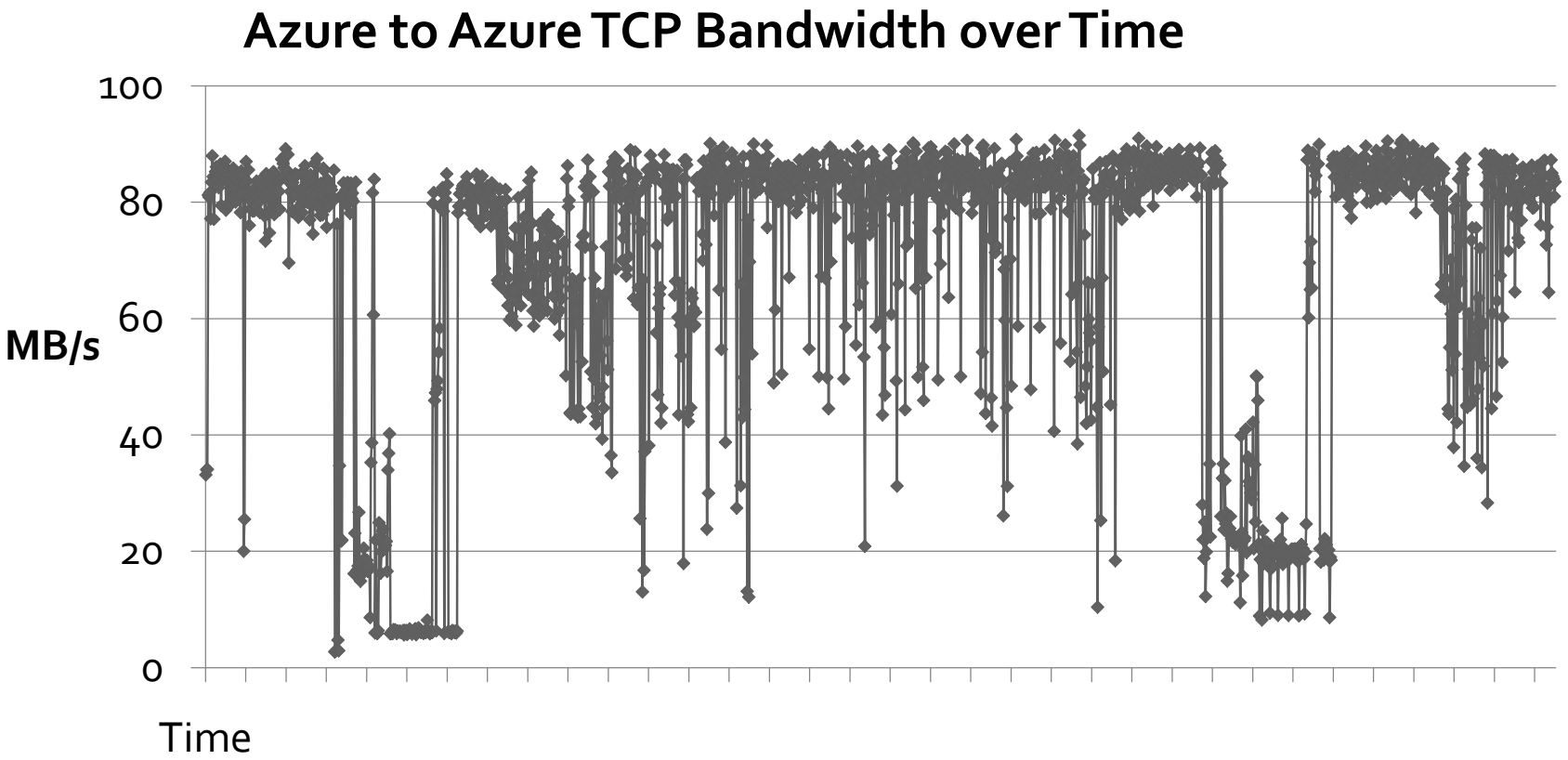
Worker Role TCP-Endpoints





TCP Bandwidth Variance Over Time

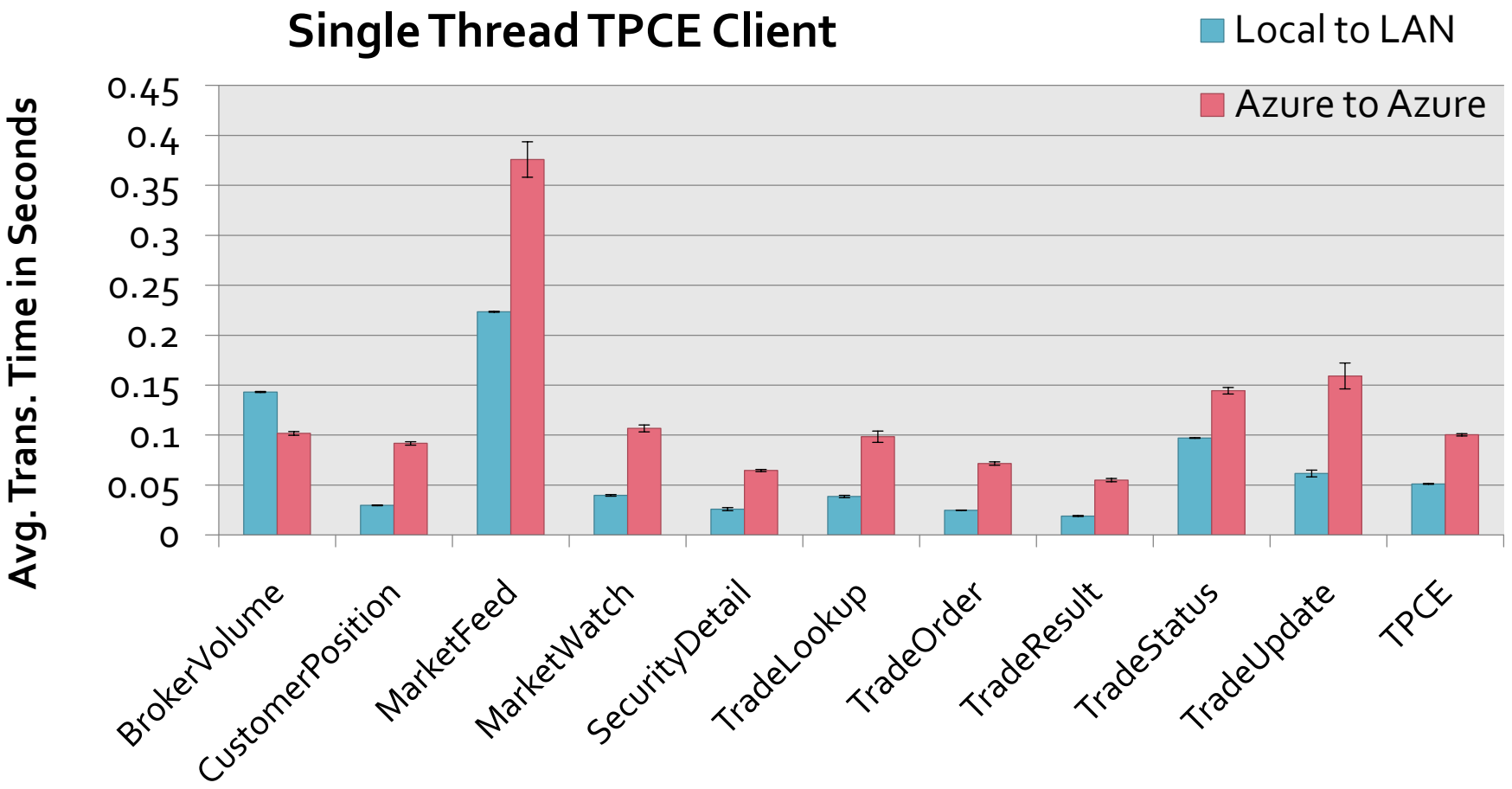
- TCP performance can change dramatically, why?



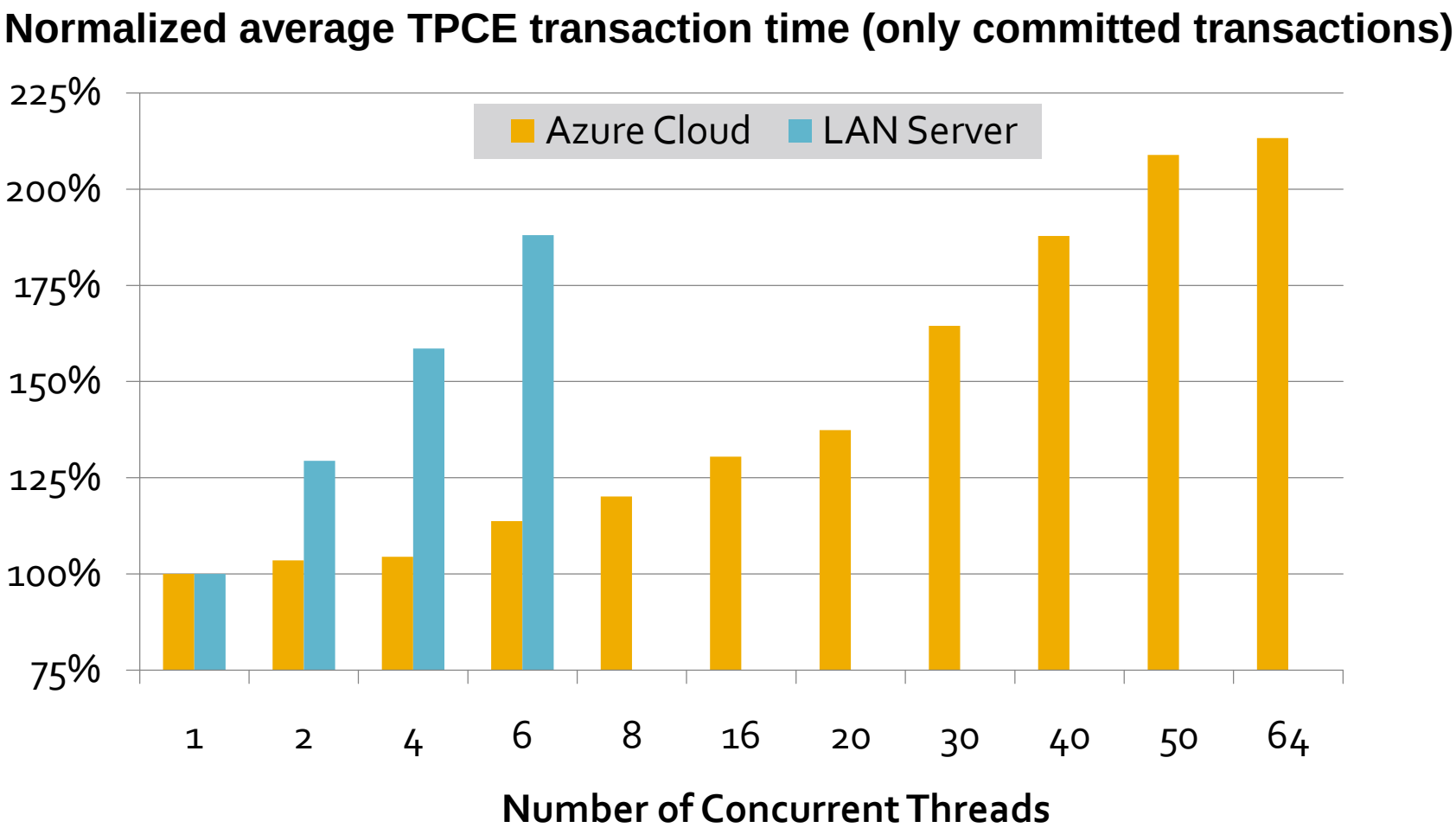
SQL Azure

- Normal SQL Server capabilities (RDBMS)
- Size limited to <50GB per database
- Tested with TPC-E benchmark for OLTP workload
  - Our .NET implementation of the benchmark
  - Simulates a brokerage house
  - Testing DB is 3GB in size

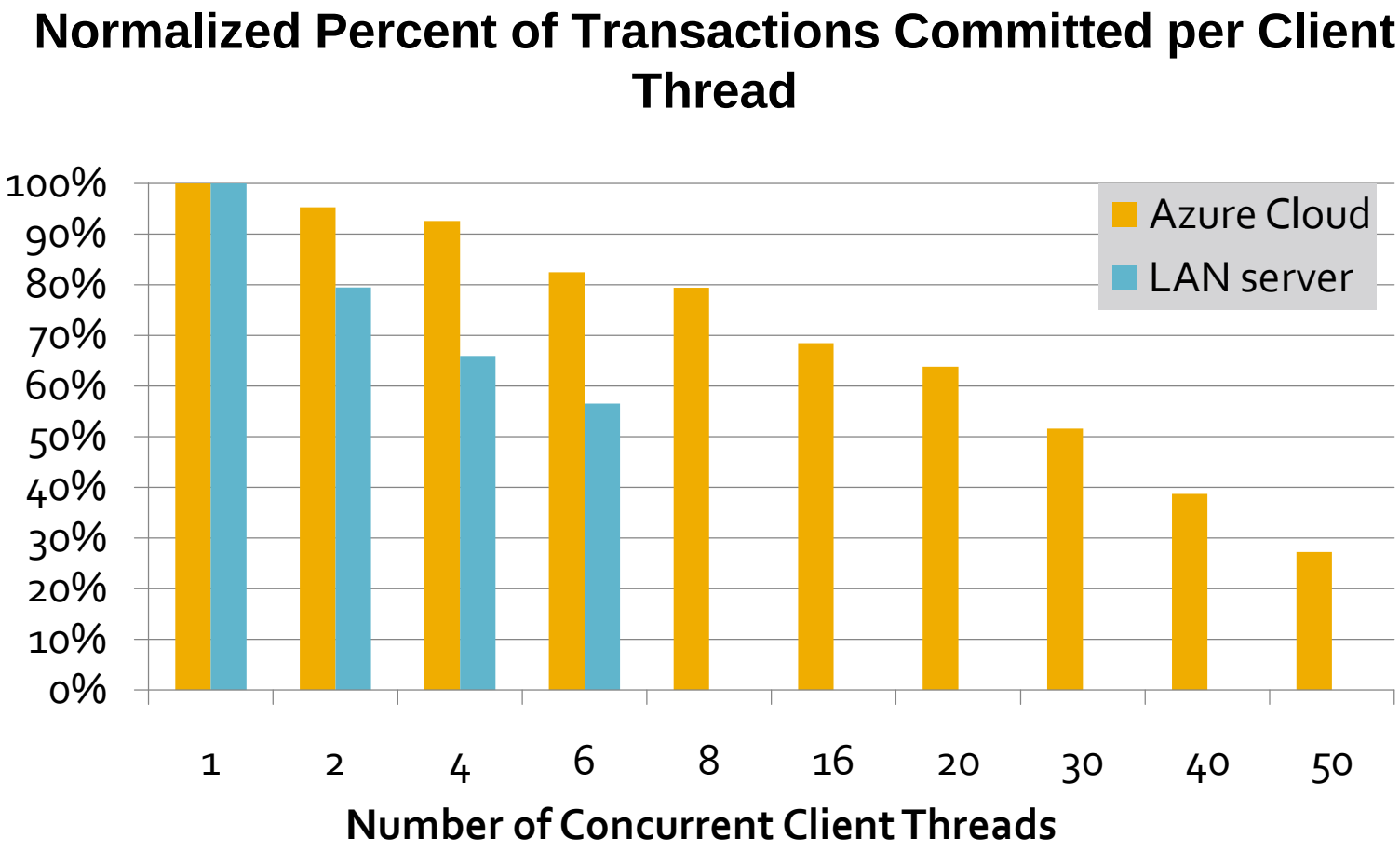
SQL Azure Performance



SQL Azure Scaling (I)



SQL Azure Scaling (II)



General Recommendations & Conclusions

- Deployment size → expected client slowdown and service throughput
- Deployment scaling is slower than initial deployment, web roles slower than worker roles, large VMs slower than small VMs
- VM deployment can take a long time depending on how many are requested
- Distribute blob accesses across as many containers as possible to achieve performance at scale

General Recommendations & Conclusions (II)

- Access tables by partition and row key. Property filters are slow
- Tables scale well for query and insert, but watch out for delete and update – this is expected
- Expect SQL Azure 2x slowdown
- SQL Azure scales reasonably well, especially under 30 or less concurrent clients
- SQL Azure performance over time: low variability