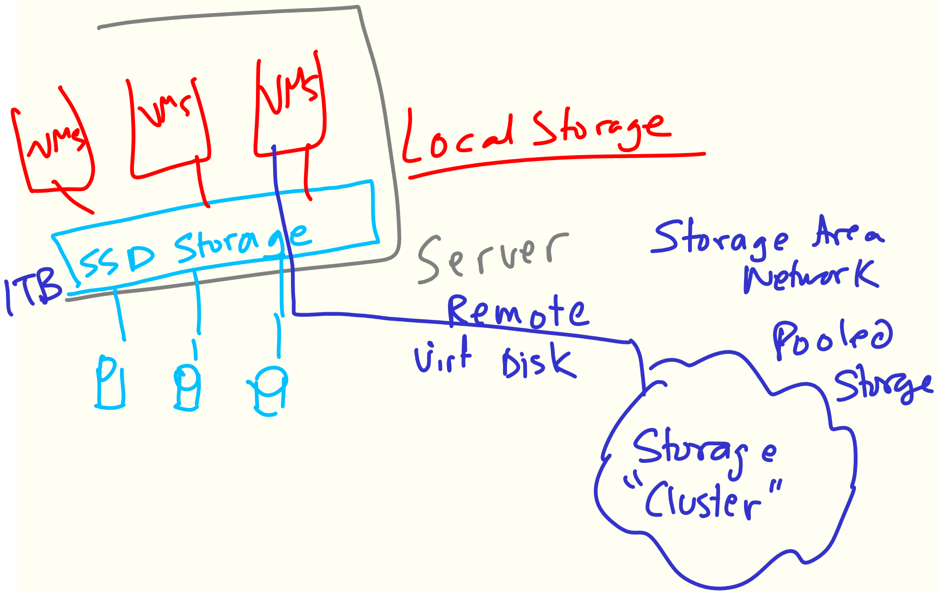
Cloud Storage & Data Center Efficiency

Data Storage Options

- Local SSD and HDD block storage (often the costliest option)
- Network block storage (more flexibility than local)
- Long-term archival (Amazon redshift, S3 Glacier, possibly tape based)
- Key-value and document databases [Amazon DynamoDB, firestore, BigTable, etc.]
- Realtime databases (Firebase)
- Managed databases as a service

Many places and abstractions for application data on the cloud.

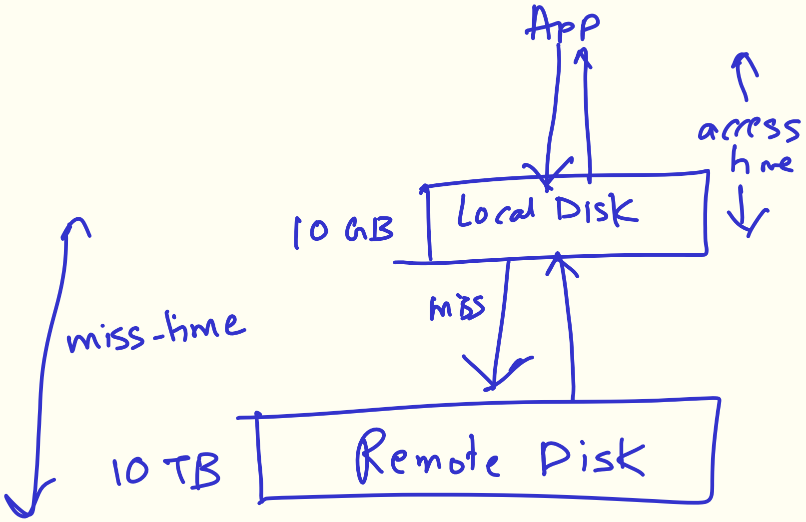


Storage Tiering

- Many options for storing data
- Price vs. latency vs. capacity tradeoffs
- Faster storage is more expensive and of lower capacity
- Pricing is a combination of: capacity and number of I/O operations

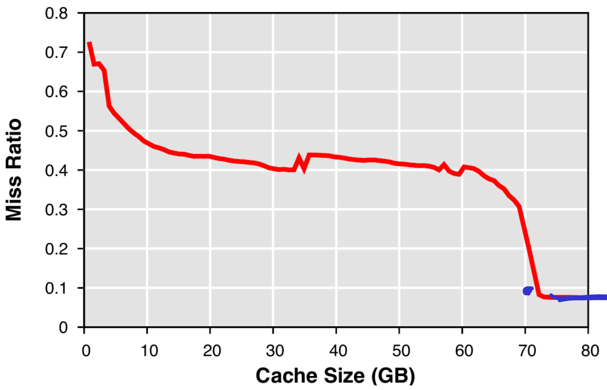
  16TB 10^6 IOPS → 1$

- Caching: Store commonly used items on faster storage
- Use spatial and temporal locality to lower average access latency
- Items are more likely to be accessed if they or nearby items have been accessed in the past
- Unused items evicted using Least Recently Used (LRU)



Storage Provisioning

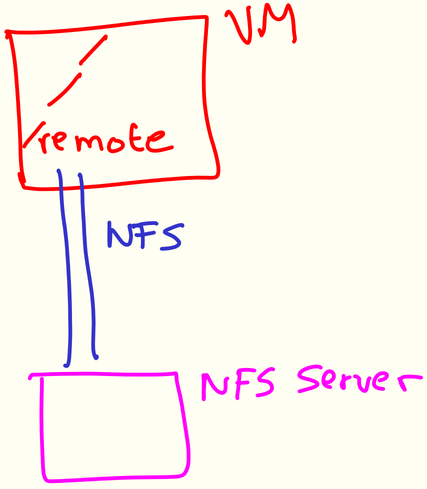
- Effective Access Time = (cache-hit-ratio)*(access-time) + (miss-ratio)*(miss-time)
- Can also be adapted to compute expected cost
- Cache provisioning using miss-ratio-curves :



$$\text{miss ratio} = \frac{\#\text{ misses}}{\#\text{ accesses} = (\#\text{ hits} + \#\text{ misses})}$$

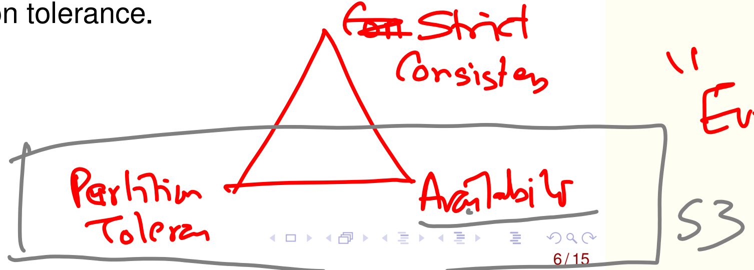
Managed NFS : Network File System

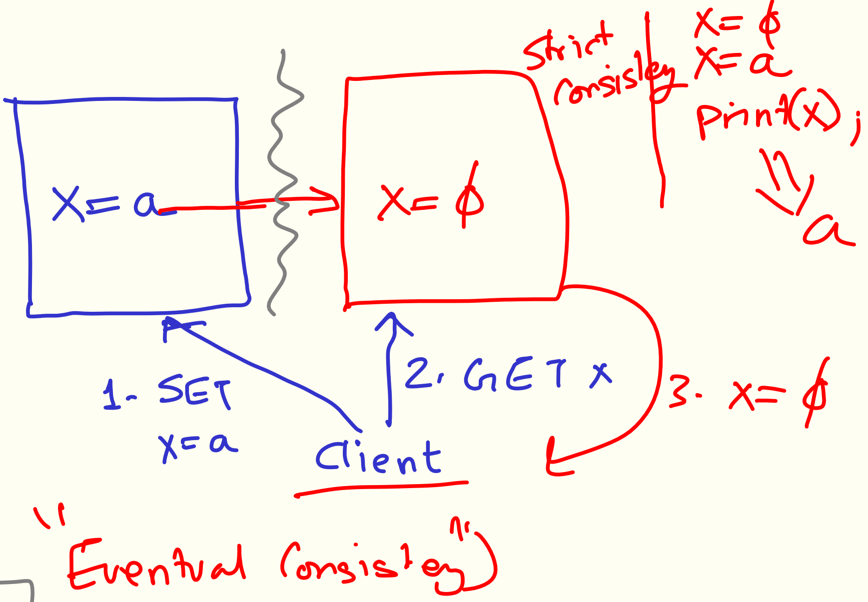
- Conventional file-server abstraction
- Block storage: Users setup their own file system
- Amazon Elastic File Server
- Google Cloud Filestore



Amazon S3

- Key-val store with buckets
- REST API for access
- All objects automatically replicated
- Possible to replicate objects across regions
- Cannot always use it as a conventional block storage: only has eventual consistency
- Replication and consistency have a fundamental tradeoff
- CAP Theorem: At a given time, can only have two of Consistency, Availability, and Partition tolerance.





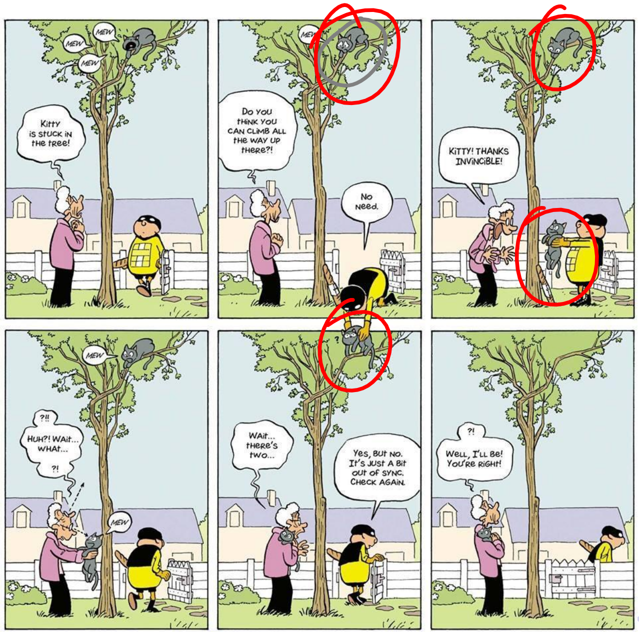
Eventual Consistency

Eventual Consistency

If no updates take place for a long time, all replicas *eventually* become consistent (have the same data stored)

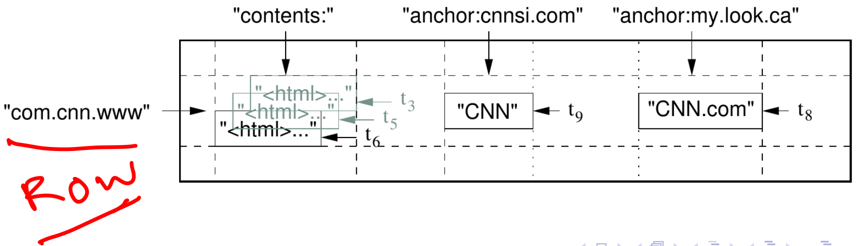
- Easy to implement
- In practice, write-write conflicts handled through some form of leader election
- Inconsistency windows often small (<500 ms)

Eventual Consistency

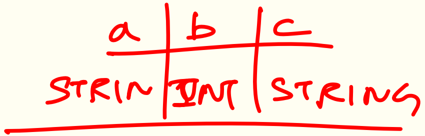


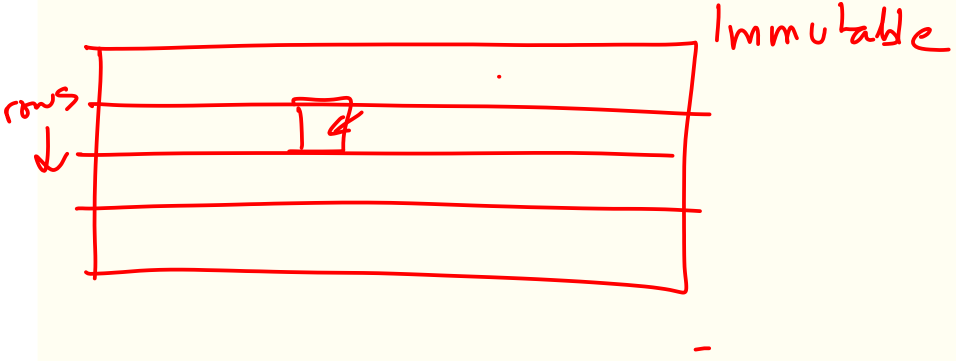
Google BigTable ~ 2006

- Storage engine for semi-structured data (aka NoSQL)
- Items may have different fields, unlike traditional relational DBs → strict schema
- Can view as a "Document database" or specialized key-value store
- Data model: (row, column, time) → array of bytes
- Timestamps help in versioned data management
- Building block: Sorted-strings table (chunks of data and index)
- Strong consistency: row operations are atomic



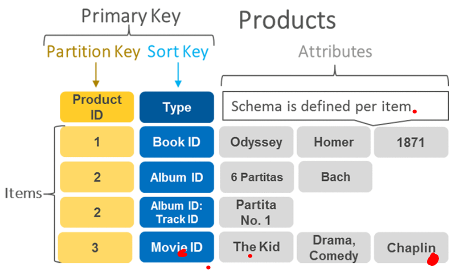
Higher level of storage abstraction can yield better performance and consistency properties.





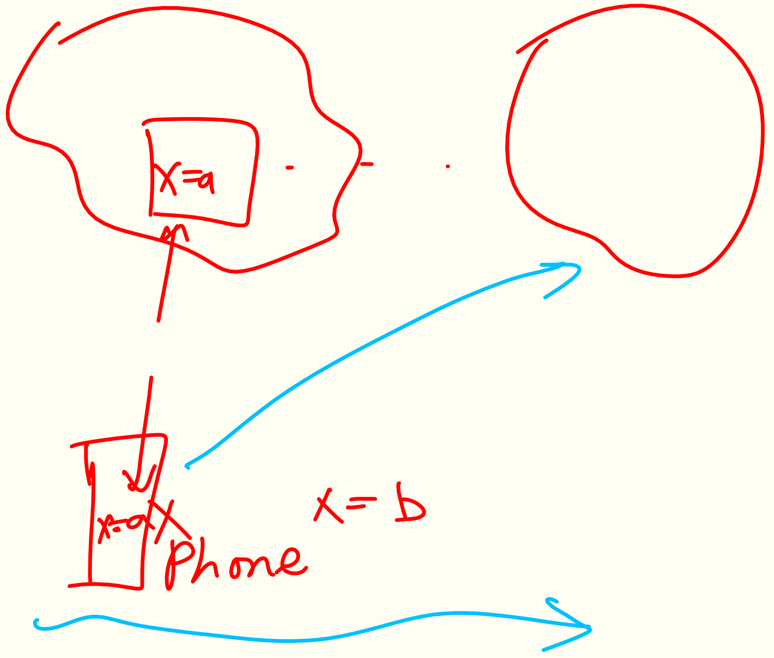
AWS DynamoDB

- Also a distributed NoSQL store
- Uses a key to partition items
- Optionally, another key for sort items
- Many policies available for partitioning and replicating items to different geographical regions, indexing, auto-scaling, replication, etc.



Google Cloud Firestore

- NoSQL
- Includes support for data synchronization between clients and server
- Offline mode for mobile applications
- Strong consistency
- Multi-region replication



Data Centers

Data Center Efficiency

- Efficient data centers are the key to low-cost cloud services
- Many location considerations: access to networking, power, proximity to users, ...
- Key metric is Power Usage Efficiency
- PUE = $\frac{\text{Total power consumption}}{\text{IT power consumption}}$
- Conventional enterprise data centers: PUE $\sim$ 2
- PUE = 2 $\rightarrow$ half the power is spent on non-IT stuff
- Most modern cloud data centers have PUE $\sim$ 1.2

~ $10 - 10^2$ servers

Servers + Networking Equipment + Storage

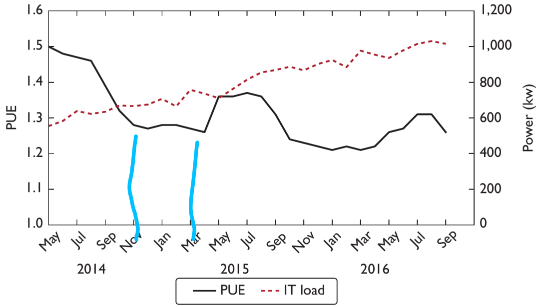
Power Consumption In Data Centers

1. Servers and network switches
2. Cooling
3. Electricity distribution losses (UPS etc)
4. Data center operation (lights etc.)

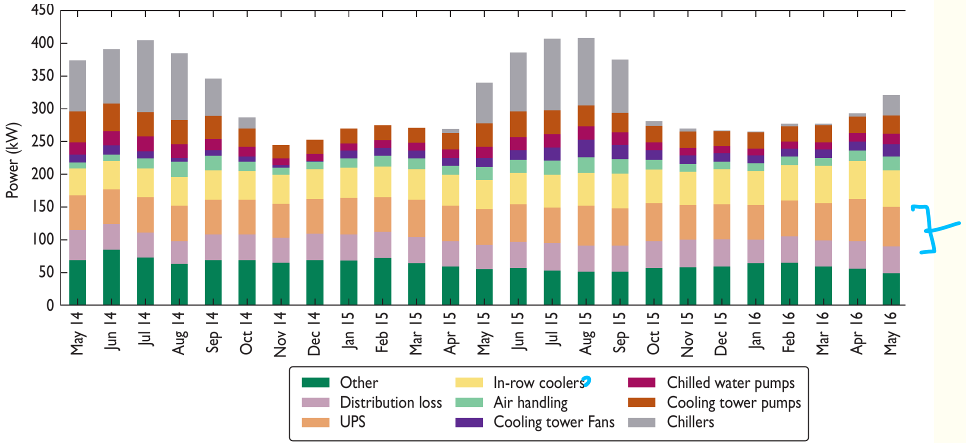
Data Center PUE

- Minimzing PUE primarily depends on cooling efficiency
- “Free air cooling”: Use outside cold air instead of conventional chillers
- Run servers as hot as possible
- Build data centers in cooler regions (near the artic circle, underwater, . . .)

- With free-air cooling, PUE depends on weather conditions
- Data from Massachusetts Green High Performance Data Center :



Energy Breakdown



- No chillers needed during winter months
- In-row-coolers circulate air, and are always operational