

Building Scalable Applications

with MySQL, Redis & Cassandra

Jacob Samro

MySQL

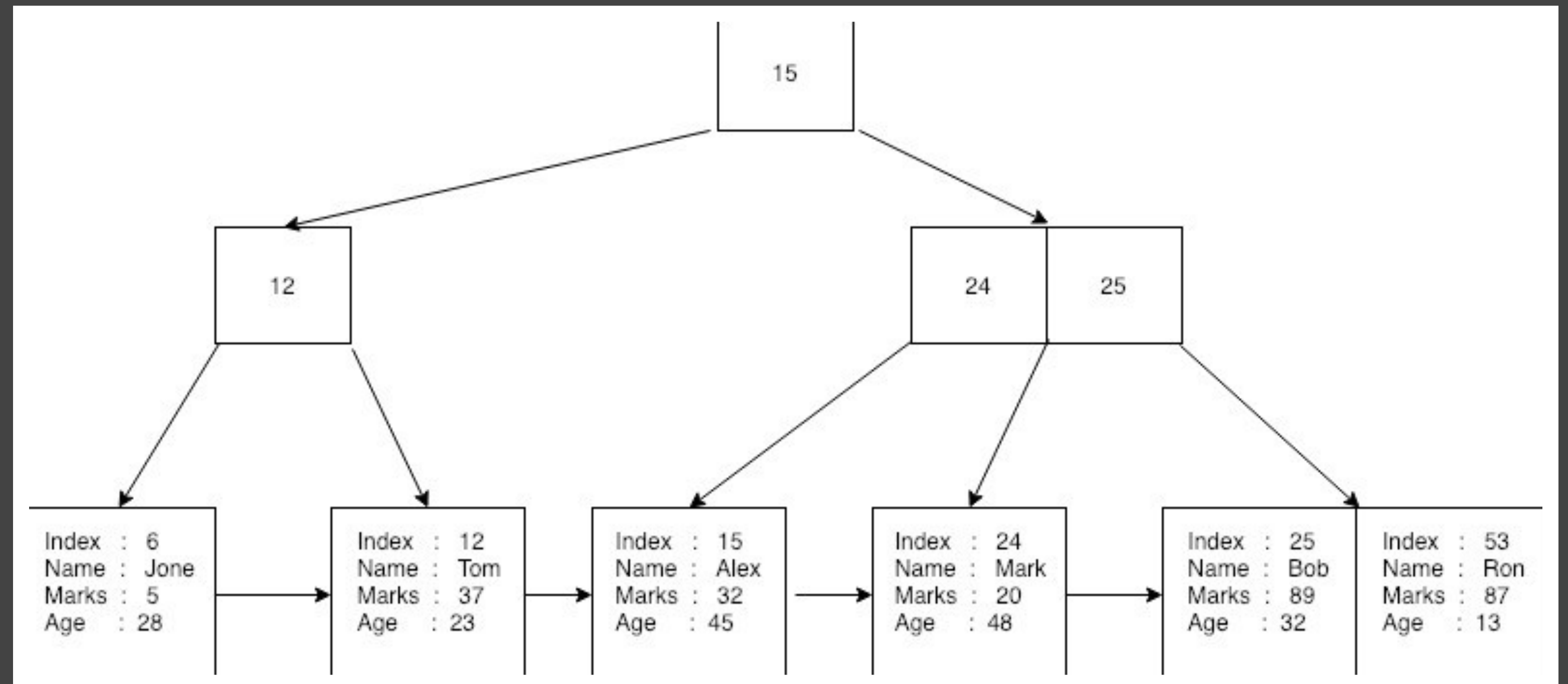
Relational Database

- Works well with Structured Data
- Reading is fast, because of structured trees (eg. BTREE)
- Distributed, Supports Replication
- Widely Adopted
- Easy to use
- ACID Compliant

MySQL

The Dark Side

- Poor write speed (INSERTion time goes up when the table grows)
- LOCKs
- Hard to Scale



Redis

REmote DIctionary Server

- Initial Release May 10, 2009
- Open Source
- Written in C
- Open Source, BSD 3 - Clause

Why Redis ?

- In memory database
- Distributed
- Persistent Key Value
- Supports Node.JS, PHP, Go, Java
- Supports almost all necessary data types
- Supports Replication & Clustering
- No dependencies
- Supports wide range of use cases than Memcache

Cassandra

NoSQL DB Known for its write speed

- Initial Release July, 2008
- Open Source
- Written in Java
- Open Source, Apache License 2.0

Why Cassandra ?

- Handles large amount of data
- Easy to scale
- No single point of failure
- Non durable writes, hence robust write speed
- Distributed Globally with minimal effort
- Linear scalability

Scenario 1 - Wikipedia

- Caching Webpages
- Articles

Scenario 1 - Wikipedia

- Caching Webpages
 - Redis
- Articles
 - MySQL

Scenario 2 - YouTube

- Live Video - Chat
- Video Comments
- YouTube Videos

Scenario 2 - YouTube

- Live Video - Chat
 - Chat - Cassandra to MySQL
 - Live View Count - Redis to MySQL
- Video Comments
 - Cassandra to MySQL
- YouTube Videos
 - MySQL

Scenario 3 - Facebook

- Posts
- Post Reactions
- Messenger

Scenario 3 - Facebook

- Posts
 - MySQL
- Post Reactions
 - Redis to MySQL
- Messenger
 - Cassandra to MySQL

When to use Redis ?

- Session Cache
- Full Page Cache
- Rate Limiting API Requests (Data Expiration)
- Shared Caching
- Leaderboards
- Realtime Analytics

When to use Cassandra ?

- Write fast, process later.
- Comments, Messages & User based content in large volume
- Logging, Transaction Logs
- Browsing History
- User Activity

Write Performance

Service / Action	MySQL	Redis	Cassandra
Write	1ms	1ms	1ms
Write after 10K Records	500ms	10ms	10ms
Write after 50M Records	5000ms	15ms	12ms

Read Performance

Service / Action	MySQL	Redis	Cassandra
Read one Record	1ms	1ms	1ms
Read one in 10K Records	500ms	500ms	5000ms (5s) *
Read one in 50M Records	1000ms (1s)	1000ms (1s)	15000ms (15s) *

Who are all using Redis ?

- GitHub
- VMware
- Twitter
- Snapchat
- Stack Overflow
- Udemy
- Slack

Who are all using Cassandra ?

- Uber
- Facebook
- Netflix
- Instagram
- Coursera
- Reddit

Managed Cloud Providers for MySQL

- AWS - RDS
- Digital Ocean
- Google Cloud Platform (GCP)
- Can be installed on any VPS

* Digital Ocean is good to start

Managed Cloud Providers for Redis

- AWS - Elastic Cache
- Digital Ocean
- Google Cloud Platform (GCP)
- Can be installed on any VPS

* Digital Ocean is good to start

Managed Cloud Providers for Cassandra

- AWS - Keyspaces
- Instacular
- Can be installed on any VPS

* AWS is good to start