# CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

## DATACENTER OUTAGE STUDIES

Lecture C

FACEBOOK OUTAGE

# Overview

- Outage occurred on 23rd September, 2010
- FB unreachable for 2.5 hours (worst in past 4 years)
- Facebook published post-mortem
  - https://www.facebook.com/notes/facebook-engineering/more-details-on-todays-outage/431441338919
- Not our goal to say Facebook is a bad infrastructure
  - In fact, after the outage, Facebook is still market leader in social networks
  - Facebook fixed infrastructure to prevent recurrence

# Background

- Data stored in a persistent store and cache
  - Persistent store = many servers
  - Cache = many servers running a distributed cache system
- Includes configuration data
- FB has automated system for verifying configuration values in the cache
  - And replace invalid values with updated values from the store

# Timeline

- On Sep 23, FB made a change to the persistent copy of a configuration
  - Change was invalid
- All clients (FB cache servers) saw invalid value
  - All attempted to fix it
  - All queried cluster of databases
  - Databases overwhelmed quickly by 100K’s queries per second
- Team fixed the invalid configuration

(Is it over yet?)

# Timeline (2)

- When client received error from DB, it interpreted it as invalid and deleted cache entry
  - When DB failed to respond => client created more queries
  - No back off
  - Rinse-n-repeat
  - (Cascading failures)

# Timeline (3)

- FB's solution
  - Turn off entire FB website
  - Stop all traffic to DB cluster
  - DB recovers
  - Slowly allow users back on: allowed clients to slowly update caches
  - Took until later in day for entire site to be back up

# Lessons Learnt

- New configuration system design
- When cannot access resource
  - Don’t retry aggressively
  - But instead, back off
  - Each time a request fails, wait twice as long as last time
    - Called “Exponential backoff”
    - Used in networking protocols like 802.11 and TCP to avoid congestion