Chair for Decentralized Information Systems and Data Management
TUM School of Computation, Information and Technology
Technical University of Munich

Cloud Information Systems

Exercise 11

13th January 2025

Prof. Dr. Viktor Leis, M.Sc. Till Steinert, M.Sc. Jana Vatter

1. Announcement: Guest Lecture This Week

- Guests: Hetzner Cloud
- Wednesday *15.01.2025*, 12:30 in HS1
- Content of the Talk is **relevant for the exam**

2. Recap: Throughput, Latency

Throughput:

- operations per time unit (e.g., 100 transactions per second)
- can be improved through parallelism

Latency:

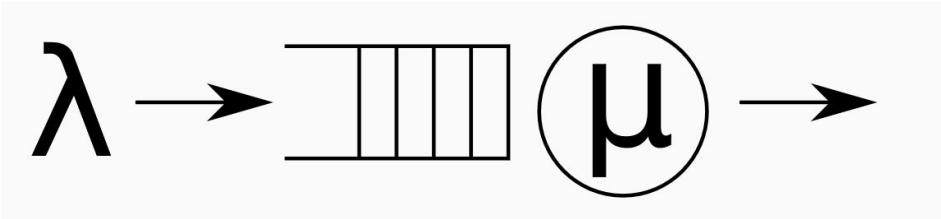
- duration of one operation (e.g., 10ms per transaction)
- harder to improve due to hardware limitations (e.g., network latency)
- scaling can actually increase latency (due to synchronization)

2. Recap: Throughput, Latency

Latency in Cloud-Native Architectures:

- cloud is about plugging together many different (micro-)services
- services call other services call other services…
- at each service: queuing delays, execution times, and network latency

2. Recap: Queuing Theory



- jobs arrive in system with rate λ
- jobs are processed at service station with processing speed μ
- jobs might have to wait (queue is unbounded)
- job inter-arrival times are exponentially distributed
- processing speed is exponentially distributed

→ M|M|1 Queue

2. Recap: Queuing Theory

parameters:

- arrival rate: λ
- processing speed: μ

metrics for M|M|1:

- system utilization: $\rho = \lambda/\mu$
- length of waiting queue: $L_q = \rho^2/(1-\rho)$
- time in waiting queue: $W_q = L_Q/\lambda$
- time in system: $W = W_q + 1/\mu$
- jobs in system: $L = \lambda W$

[Software Architecture for the Cloud, slide 65]

2. Exercise: Queuing Theory

- Repeat the calculations for M|M|1 (“Software Architecture for the Cloud”, slide 65) with
  - $\lambda = 10/s$ and $\mu = 30/s$
- How do the values change when the arrival rate doubles ($\lambda = 20$)?

2. Exercise: Queuing Theory

parameters:

- arrival rate: **λ = 10/s**
- processing speed: **μ = 30/s**

metrics for M|M|1:

- system utilization: $\rho = \lambda/\mu = 10/30 = ⅓$ (~33%)
- length of waiting queue: $L_q = \rho^2/(1-\rho) = (⅓)^2 / (1-⅓) = ⅙$ (~0.1667)
- time in waiting queue: $W_q = L_Q/\lambda = (⅙)/(10/s) = 1/60$ s (~0.01667 s)
- time in system: $W = W_q + 1/\mu = 1/60$ s + 1/(30/s) = 1/20 s (~0.05 s)
- jobs in system: $L = \lambda W = 10/s * 1/20$ s = ½ = 0.5

[Software Architecture for the Cloud, slide 65]

2. Exercise: Queuing Theory

parameters:

- arrival rate: **$\lambda = 20/s$**
- processing speed: **$\mu = 30/s$**

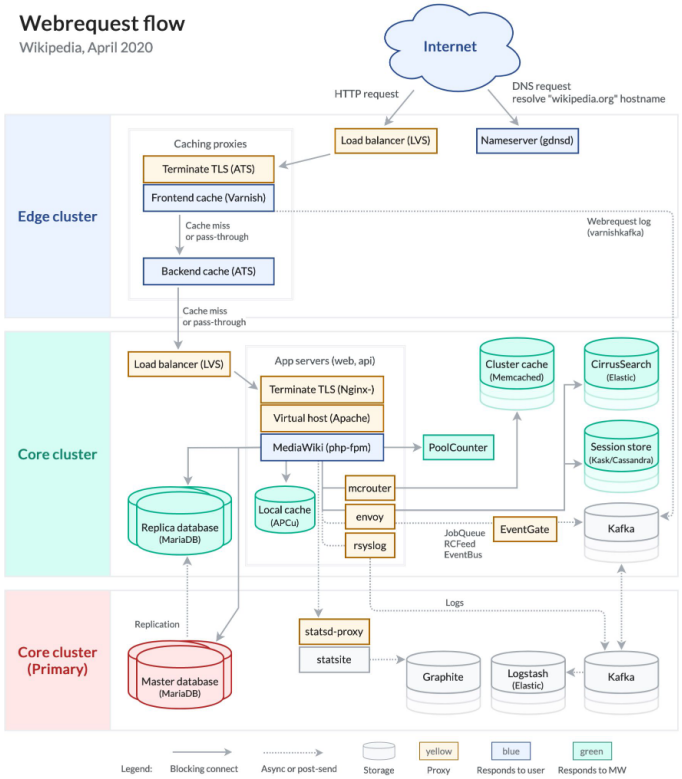
metrics for M|M|1:

- system utilization: $\rho = \lambda/\mu = 20/30 = ⅔$ (~66%)
- length of waiting queue: $L_q = \rho^2/(1-\rho) = (⅔)^2 / (1-⅔) = 4/3$ (~1.33)
- time in waiting queue: $W_q = L_Q/\lambda = (4/3)/(20/s) = 1/15$ s (~0.0667 s)
- time in system: $W = W_q + 1/\mu = 1/15$ s + $1/(30/s) = 1/10$ s (~0.1 s)
- jobs in system: $L = \lambda W = 20/s * 1/10$ s = 2

[Software Architecture for the Cloud, slide 65]

3. Exercise: Wikipedia Lift-and-Shift Architecture

- Move existing architecture to the Cloud as-is
  - Each Data Center is replaced by a deployment in a different AWS region
  - Every on-premises server becomes one EC2 instance
- Estimated 2500 EC2 instances needed at the current scale ($3000/year each) = $7.5M/year
- High egress cost from AWS to public internet = $12M/year
- TCO ≈ $20M



3. Exercise: Cloud-Native Architecture

- Every component is replaced by an off-the-shelf AWS service
- New architecture:
  - geo-distributed HTTP caching → Amazon CloudFront [$8.4M/year]
  - load balancer → Amazon Elastic Load Balancer (ELB) [$1.9M/year]
  - application server →Keep custom PHP implementation [$1.5M/year]
  - database → AWS Aurora [$400K/year]
  - image storage → S3 [$42K/year]
- TCO ≈ $12M
- Where should we focus to reduce cost further?

3. Replace S3?

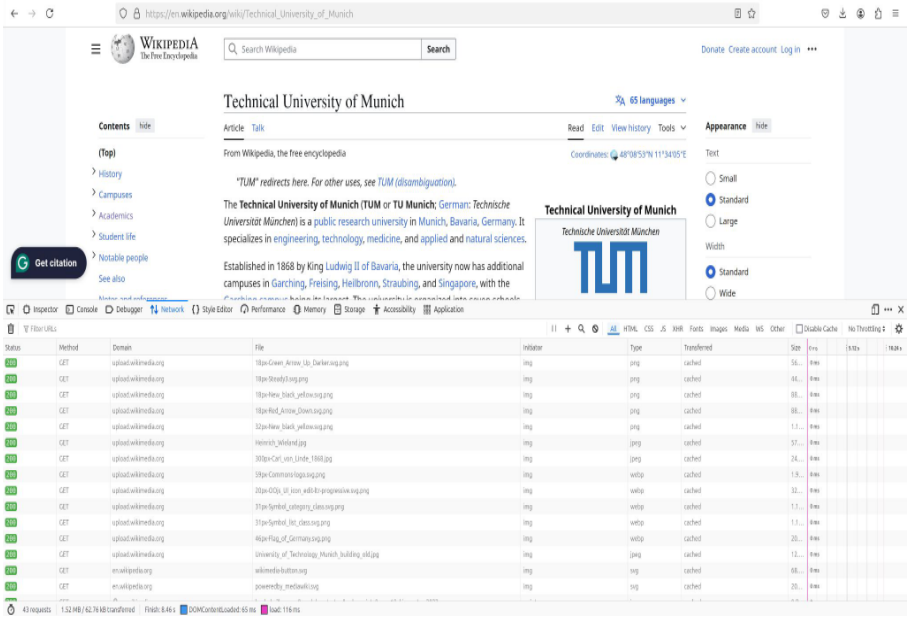
- Competing Object Stores with S3-compatible API
- S3 only $42K of TCO; how will this help us?

https://developers.cloudflare.com/r2/pricing/

https://docs.hetzner.com/storage/object-storage/overview/

3. Replacing S3 with Cloudflare R2

- Cost dominated by AWS Cloudfront CDN
- Minimum traffic cost: $0.025 per GB
- Request cost: $1 per 1M HTTP requests
- Wikipedia:
  - Requests (assuming 10 requests per view): $2.4M/year
  - Traffic (assuming 1MB per view): $6M/year
  - Total: $8.4M/year
- ~50% of Page consist of Images



3. Replacing S3 with Cloudflare R2

	AWS S3	Cloudflare R2
Storage Cost [100TB/year]	$24K	$18K
GET Request Cost [GET/sec]	$12K	$11K

- But, we also save on Egress Cost:
  - R2 does not charge for Egress to Public Internet
  - Assuming 50% of Traffic is Images: $6M/year * 50% = $3M

Potential Issues with R2 Migration?

- Latency?
  - Seems to be (slightly) worse than S3 (See Comparison)
- Undisclosed Egress/Rate Limits?
  - Bandwidth is not free (that's why other companies are charging for it)
  - At some Traffic Volume, Cloudflare will likely fail our Requests
- Added Architecture Complexity
  - Interoperability with AWS Cloudfront?
  - Loose advantage of having everything from one vendor

3. Why should Wikipedia (not) move to the Cloud?

- Strategic Reasons
  - Avoid Vendor Lock-In
  - AWS contrary to Wikipedia Ideology
- Lack of Benefit from Cloud Architecture
  - Stable Workload
  - Already an established Player with stable growth
- Years of Expertise
  - Architecture purpose-built for Wikipedia Use Case
  - Employee Salary can amortized over the large Cluster

4. Questions