THE UNIVERSITY of EDINBURGH
informatics

Revision Lecture

Luo Mai

University of Edinburgh

Exam Time

09:30 GMT to 11:30 GMT

19 December 2023

Location: TBD

Scope

- Data-parallel collection / key-value pairs
  - Scala, Spark RDD
- Query processing
  - Spark SQL
- Graph processing
  - Pregel
- Machine learning frameworks
  - PyTorch / TensorFlow
- Distributed machine learning systems
  - Data parallel, model parallel, pipeline
  - Parameter servers, collective communication
- Distributed file systems
  - Google file system
- Cloud resource management
  - VM / Container / Container Orchestration

Exam type

- Open-book exam (2 hours)
- 3 Compulsory Questions (Different marks)
- Total marks: 50 (20 + 15 + 15)
- ALL contents come from lecture slides and compulsory reading
- Guest lectures are not examined

Office Hours:

Informatics Forum 2.03, 13:00 – 15:00, 12 December 2023

Exam materials

- You can bring printed lecture slides and Scala APIs.
- You can also bring handwritten and self-organized notes. There is no page limit for printed lectures, Scala APIs, and your notes (either handwritten or typed). These materials need to be in A4 size.

- No other printed materials.
- No electronic devices.
- No printed compulsory reading.

Question type

- Book work
  - Course material and discussion (explain concepts & items)
- Algorithms + Programming
  - Programming models for systems (key steps)
- Technology match + System design
  - Suitable scenario (provide reasoning)
  - Performance trade-off

Sample Question - Book Work

(a) Computational graph

(i). What is a computational graph in ML frameworks? Solution:

- A computational graph is an intermediate representation of a ML program. Its graph nodes represent computational operators while graph edges represent computational dependencies. (1 mark)

(ii). Describe three reasons that ML frameworks use computational graphs. Solution:

- Automatic differentiation (1 mark)
- Discovering parallelism (1 mark)
- Decoupling frontend and backend (1 mark)

Sample Question - Algorithms

(d) Given an undirected graph G(V, E) and a set of seed vertices V' (which is a subset of all vertices V), design an algorithm to compute the 3-hop neighbours for each seed vertex using the Pregel programming model (i,e., vertex-centric programs). The 3-hop neighbours are those can be reached from seed vertices with exact 3 hops. Please describe main steps of this algorithm in details.

- Following the BSP model (3 marks)
- Keeping an integer value as internal state and increasing it when messages are received (2 marks)
- Sending the message if there is a change in the internal state (1 mark)
- Sending the internal state (or its delta) to the neighbours (1 mark)
- Terminating condition for each of the nodes (1 mark)

Sample Question - Algorithms

Solution:

Each vertex stores an integer value, which stores the number of hops from the seed nodes to it. Initially, the value at each vertex is initialised to -1 or a default value. Seed vertices are set to 0.

At each super-step, each vertex

(i). Receives messages from its neighbours containing their vertex values.

(ii). Updates its value by setting it to the maximal value it has received and +1

(iii). If after updating, its internal set has changed, the updated value is sent to the neighbours.

(iv). Algorithm terminates when the number of super steps is greater than 3.

If the value of a vertex is 3 in the end, then it is the 3-hop neighbour of one of the seed nodes.

Sample Question - Programming

Write a Scala program using functional collections that returns a list containing the double of odd elements for an input list of type List[Int]. You are only allowed to use one invocation of flatMap, and you are not allowed to use other collection functions (e.g., map, filter, flatten, foldLeft, foldRight).

Solution:

```
l.flatMap(x => if(x % 2 == 1) List(x * 2) else List())
```

Sample Question - Technology Match

(b) Distributed machine learning

(i). Describe the differences between data parallelism and model parallelism. Solution: Data parallelism partitions the data dimension (i.e., mini batch) and model parallelism partitions the program dimension (i.e., model weights) . (1 mark)

(ii). Mention the use-cases for data parallelism and model parallelism, respectively. Solution: Data parallelism should be used in the case of experiencing computational bottlenecks and memory bottlenecks (1 mark), and Model parallelism should be used in the case of experiencing memory bottlenecks. (1 mark)

(iii). Describe the benefits and potential issues of using small micro-batch sizes in ML systems that use pipeline parallelism. Solution:

- Benefits: Small micro-batches will make the bubble size small, thus increasing device utilisation. (1 mark)
- Issues: Using small micro-batches will incur large micro-batch scheduling overhead. (1 mark)

Sample Question - System Design

(c) In the original design of Google File System (GFS), a single master maintains all file system metadata. Suppose now metadata is too large to be stored on a single machine, consider how to extend GFS where metadata is distributed on multiple machines. (Hints: you might consider the fault-tolerance, throughput performance, latency performance, etc.)

(i). Describe a brief design.

(ii). Briefly analyse the advantage of your design.

(iii). Briefly analyse the disadvantage of your design.

Design (4 marks):

- Considering how to choose different machines for different files. (router, etc) (2 marks)
- Considering the fault-tolerance. (2 marks)
- Considering the throughput performance. (2 marks)
- Considering the latency. (2 marks)
- Considering the elasticity. (2 marks)

Example Solution:

Design: We can distribute the metadata into multiple machines. Now, we need a router layer between masters and other roles to redirect requests to the corresponding master. (1 mark) In such a case, however, the router will be the performance bottleneck of the entire system. Therefore, we should put each file-to-machine mapping on multiple route machines to improve the throughput. (2 marks)

Advantages: With multiple routers, the system throughput will be high. No need to change the clientend code, only need to redirect requests from the master to routers.

Disadvantages: Because each request to the master must be processed by a router, the latency is higher than the original GFS. Making multiple routers consistent is difficult.

Questions?

Beyond exam – future job interview

Example system design questions asked at Google

- How would you design Google's database for web indexing
- How would you design Google Docs
- How would you design Google Home (voice assistant)
- How would you design Amazon's books preview
- How would you design a social network
- How would you design a task scheduling system
- How would you design a ticketing platform
- How would you design a system that counts the number of clicks on YouTube videos
- How would you design a webpage that can show the status of 10M+ users including: name, photo, badge and points
- How would you design a function that schedules jobs on a rack of machines knowing that each job requires a certain amount of CPU & RAM, and each machine has different amounts of CPU & RAM? Multiple jobs can be scheduled on the same machine as long as it can support it

You should know industry solution patterns like:

- Sharding Data
- Replication Types
- Write-ahead Logging
- Separating Data and Metadata Storage
- Basic Kinds of Load Distribution

[1] https://www.interviewkickstart.com/interview-questions/google-system-design-interview-questions
[2] https://workat.tech/system-design/article/google-system-design-interview-prep-doc-m684to5zzkj4

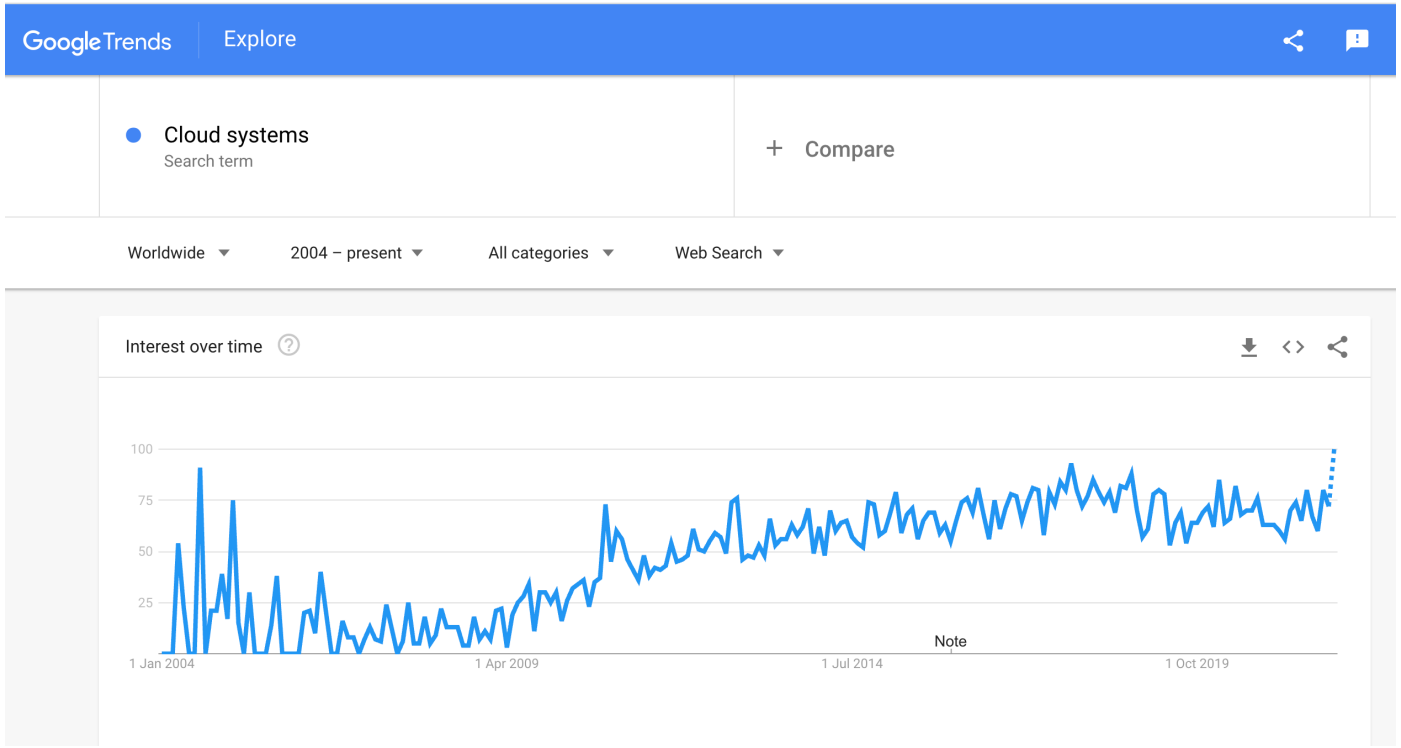
Beyond exam – future job interview (2)

```
Latency Comparison Numbers (~2012)
----------------------------------
L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns                      14x L1 cache
Mutex lock/unlock                           25   ns
Main memory reference                      100   ns                      20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy             3,000   ns        3 us
Send 1K bytes over 1 Gbps network       10,000   ns       10 us
Read 4K randomly from SSD*             150,000   ns      150 us          ~1GB/sec SSD
Read 1 MB sequentially from memory     250,000   ns      250 us
Round trip within same datacenter      500,000   ns      500 us
Read 1 MB sequentially from SSD*     1,000,000   ns    1,000 us    1 ms  ~1GB/sec SSD, 4X memory
Disk seek                           10,000,000   ns   10,000 us   10 ms  20x datacenter roundtrip
Read 1 MB sequentially from disk    20,000,000   ns   20,000 us   20 ms  80x memory, 20X SSD
Send packet CA->Netherlands->CA    150,000,000   ns  150,000 us  150 ms
```

[1] https://gist.github.com/jboner/2841832

Beyond exam – related jobs

- Software developer
- Site reliability engineer
- Data scientists
- Researchers / Scientists
- Entrepreneur



Questions?