INFO8002

Large-Scale Data Systems

Exercise Session #5

Academic year 2021-2022

LIÈGE université

Reminder

REMINDER :

File Systems

- A **File** is a collection of data organized by the user.
- A **File System** is the component of an operating system that is responsible for managing files:
  - Physical Allocation
  - Access → Allow the users to create/destroy/read/write/append/… in a human-friendly way.
  - Security & Permissions Management

REMINDER :

Distributed File Systems

- A **distributed file system** (DFS) is a class of distributed systems that specifies how data is stored and retrieved across a network of machines.
  - **Location Transparency** - The system is designed to provide an abstract view of a single-system to the user.
  - **Redundancy** - The system is designed to ensure resilience in case of machines failures.

REMINDER :

Distributed File Systems - Challenges

- **Performance** → How to mimic the performance of a file system while the machines can communicate at much lower speed?
- **Scalability** → How to ensure that the performance/resilience/security of the file system hold for much larger scale?
- **Data Integrity & Consistency** → How to ensure integrity & consistency of files in the presence of multiple writers?
- **Simplicity & User friendliness**

REMINDER :

Distributed File Systems - Architectures

- **Centralized File Systems** → One or several Master nodes handle the requests of the users.

- **Decentralized File Systems** → Any node can handle any type of requests of the users.

  /!\ Nodes still may have ≠ roles.

ABSTRACT

- **GFS → Master only tells where chunks are.**
- **Others → Master may gather the chunks and answer with the whole file.**

REMINDER :

Distributed File Systems – Architectures

- **File Model** → DFS might use different conceptual models for a file.
- **Sharding Mechanisms** → DFS might use different mechanism to apply sharding.
- **File/Chunks Replication** → DFS might use different mechanisms for file/shard replications.
- **File Access Model**
- **File Caching Scheme**
- **...**

REMINDER :

Distributed File Systems - Performance

- **Latency** → Not often designed for low latency data access (cost for heterogeneity).
- **Size** → Not often designed for many small files.
- **Dynamicity** → Not often designed for constantly changing data.

Distributed Computing

- **Distributed computing** provides an interface to the users to perform a restricted set of operation on data.

REMINDER :

Distributed Computing - Challenges

- **Performance** → How to extract the highest effective amount of computing power from a set of workers? (Locality?)
- **Scalability** → How to ensure that the performance/resilience/security hold for much larger scale?
- **Fault Tolerance** → How to allow the system to run with faulty workers & how the performance are impacted? (Redundant Execution? Cost?)
- **Simplicity & User friendliness**

PROBLEM 1

Longest Word Crawler

You are responsible for designing a data processing pipeline which is looking for the longest word on the World Wide Web. Your bots will periodically crawl the web add push the webpages to your servers. Then, your system will be periodically process the set of webpages to discover for the longest word found so far.

Discuss the **Storage Architecture** ***-i.e.*** file model, sharding, replication, etc as well as the operations that will be performed on these data. Please consider scenarios where worker nodes fails, how this will be detected and how the system will react to these failures.