Distributed Systems

Tran Giang Son, tran-giang.son@usth.edu.vn

ICT Department, USTH

Course Introduction

Expected Learning Outcomes

- Understand key concepts of distributed systems
- Explain structure of distributed software
- Develop distributed algorithms and programs
- Improve team work

Content

- [6h] Introduction to Distributed Systems
- [6h] Communication: RPC / Message Passing
- [3h] MapReduce pattern
- [3h] Distributed file systems
- [6h] Peer-to-peer: Bittorrent, Bitcoin
- [6h] Cloud computing

Format

- 3 ECTS = 36 hours
- Lecture / Practical work : 27h / 9h
- Prerequisites: Some programming language
- Environment: Linux/Mac
  - Windows isn't supported, as usual ☺
- Assessment:
  - Attendance / Project / Final Exam : 10% / 30% / 60%

Policy

- Collaborations!
  - During exercises
  - During the project
  - Not in the final exam

Projects

- 4-5 students per group. Do it **now**.
- Only one pass: before exam
- Any learnt language at USTH
  - C/C++/Java (preferred)
  - PHP
  - Javascript
  - Go
  - Bash (!)
  - Assembly (!!)

Projects

1. Fault-tolerant Key-Value server/client using RPC (Redis clone)
2. Fault-tolerant Key-Value server/client using MPI (Redis clone)
3. HTTP over RPC (act as HTTP proxy)
4. HTTP over MPI (act as HTTP proxy)
5. Distributed file system (GlusterFS clone)
6. Distributed database

Note: you ALL need scalability and consistency!

Projects

7. Hybrid centralized and peer-to-peer chat system using socket
8. Hybrid centralized and peer-to-peer chat system using MPI
9. Peer-to-peer file transfer
10. Remote shell using RPC (multiple clients)
11. Remote shell using MPI (multiple clients)
12. Remote shell using raw socket (multiple clients)

Note: you ALL need scalability and consistency!

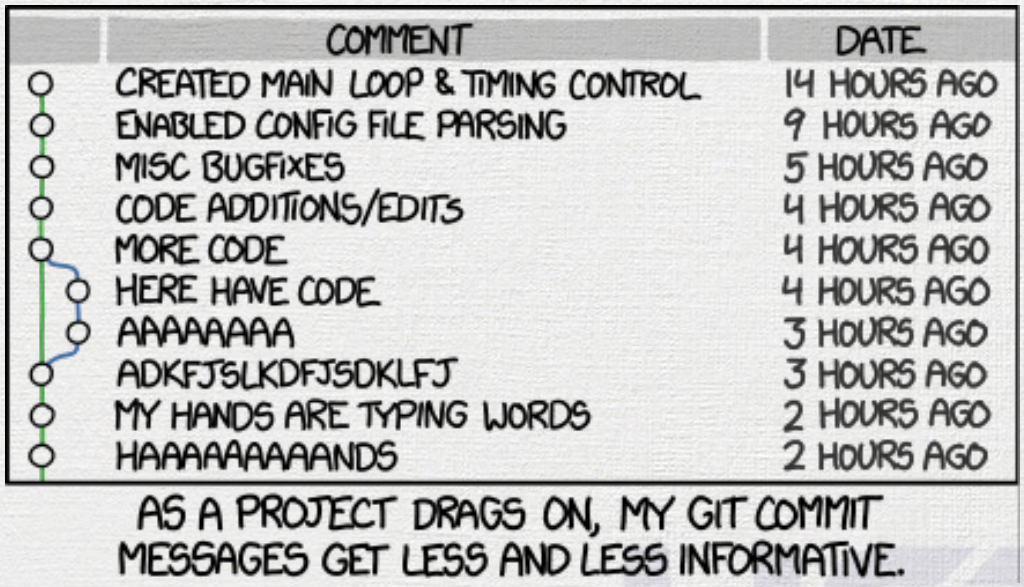
Labwork

- Several C/C++/Java Programs
  - Understand what you learnt
  - Show your ability to apply it to new problems
- Compilable on Linux/Mac
  - Windows is not supported (by me, as usual ☺ !)
- Don't copy paste. **Write your own code yourself**
  - I have checker tools ☺
- Should be well organized and well written

> *Bad programmers worry about the code. Good programmers worry about data structures and their relationships.*
>
> - Linus Torvalds

Labwork

- Git : Version Control System
- Github
  - Initial repository and instruction
  - https://github.com/SonTG/ds2018



Don't be lazy with your commit messages

Exams

- 3 sheets of A4 documents are allowed
- No laptop / mobile phone / internet
- No discussion, of course

References

- Books:
  - Tanenbaum et al., Distributed systems: principles and paradigms, 2nd Edition, Pearson.
  - [Optional] An Introduction to Parallel Algorithms, Joseph JáJá. Addison-Wesley Professional. 1992.