

Distributed Systems Project Topics (Suggestions)

The project is designed for you to freely extend the lab material to a broader context. You will have the opportunity to build a running distributed system. The learning outcome of the project is for you to:

- understand the technology stack of building distributed systems, and
- explore issues, trade-offs and opportunities in terms of performance, stability and complexity of distributed system design.

Several topics are suggested. Under each topic, a brief outline is given to help you focus on the tasks at hand.

The details are up to your group's interest and technical expertise. You are free to choose your operating system, programming language and tools.

The topics are not necessarily exclusive. It is possible (in fact likely) that your project may touch on several topics.

1. Resilient Systems

- Build a service. For example, it can be a simple weather reporting system that serves feeds based on existing Weather web services (e.g. https://weather.gc.ca/business/index_e.html).
- Extend your service to have a *heart-beat* service similar to ping ([http://en.wikipedia.org/wiki/Ping_\(networking_utility\)](http://en.wikipedia.org/wiki/Ping_(networking_utility)))
- Design and implement ways to make your service **resilient**. You should start with the set of failure scenarios you wish to combat (e.g. process termination (due to unforeseen runtime errors such as NullPointerException), resource inavailability such

as network failure, or server failure). Your strategy may include designing a secondary monitoring service that restarts the primary service when heart-beat is interrupted.

- Your final design must span over multiple machines, so you can cope with server failure.

2. Web Services Mash-up

- Pick a set of web services available online.
- Propose a way to compose them in an interesting way. For instance, you can consider to pull blogs using Tumblr API (<https://www.tumblr.com/docs/en/api/v2>), and invoke Google Translate API (<https://cloud.google.com/translate/v2/faq>) to provide multi-lingual version of the article. (NOTE: Google Translate API is a paid service, so you will need to register for it).
- Design a **new** service that is built on up of the existing services.
- Design an user interface for the new service.
- Implement a prototype for the new service.

3. Distributed Algorithms

You are to pick one of the following distributed algorithms. You need to understand how it works, and provide a proof-of-concept implementation demonstrating the algorithm.

- Gossip algorithms
- Distributed consensus and leadership election
- Distributed spanning tree

4. Monitoring and logging

- Build a simple peer-to-peer (p2p) network. It can be several processes, running on several different machines, exchanging tokens randomly at some rate.
- Design a monitoring strategy to gather the following metrics:
 - the total tokens-sent rate of any peer

- the total tokens-received rate of any peer
- Emphasis on the scalability and stability of your monitoring system:
 - What if your p2p network grows over time?
 - What if your p2p network grows to millions?
 - What if some peers are unstable (leaving the network and rejoining because they are mobile clients)?
 - What if your monitoring facility is faulty (machine crash, disk failure)?
- Provide a proof-of-concept implementation based on your design.
- Design and implement (real-time) graphical visualization of the metrics being gathered by the monitoring system.

5. Programming languages for distributed systems

In recent years, several industrial strength programming languages have been introduced. Many of them feature native syntactic and runtime support for programming concurrent systems. You are asked to investigate these features, and implement a distributed system using one or more modern languages with concurrency language feature.

1. Go with go routines and channels.
2. Scala with actors and message passing.

6. A discussion forum website

Learn to use the Play framework for developing Web applications. Using Play, design and develop a website to allow users to participate in a discussion forum. It can be as simple or as sophisticated as you have the capability to develop.

7. You can also further develop more complex distributed systems that stem from any of the labs in this course.

And of course, you have the complete freedom to come up with an original project idea and carry it out.