Introduction et rappels

Architectures distribuées

Introduction

Qu'est-ce qu'un système distribué?

WEB

1. Connect to www.wooclap.com/JXQNSR
2. You can participate

SMS

1. Not yet connected? Send @JXQNSR to 06 44 60 96 62
2. You can participate

Qu'est-ce qu'un système distribué?

- Ensemble d'éléments (composants) logiciels autonomes et indépendants
- Collaboration entre ces composants par communications réseaux
- Apparaît pour l'utilisateur comme un seul et même système malgré les différents éléments impliqués (transparence de la distribution)

Les challenges techniques et scientifiques associés

- Gestion des processus
- Architecture entre les différents éléments impliqués
- Communications entre les éléments
- Nommage et découverte des éléments
- Consistance des informations et réplication
- Sécurité des communications et isolation des composants
- Coordination des éléments
- Tolérance aux pannes

Livre de référence gratuit :
https://www.distributed-systems.net/index.php/books/ds3/
Les 3 livres de Michel Raynal chez Springer :
https://team.inria.fr/wide/team/michel-raynal/

Pourquoi les systèmes distribués sont inévitables ?

- Infrastructures et systèmes distribués associés (OS)
  - Clusters pour le calcul haute performances (HPC)
  - Cloud computing, Fog/Edge computing
  - IoT systems and Cyber-Physical Systems
- Applications distribués
  - Applications web
  - Applications mobile
  - Applications à base de services et microservices

Déroulement de l'UE

Déroulé des 4 premières séances

Travail par groupes de 2 étudiants

1. **Aujourd'hui (5 octobre)**
   a. Introduction et rappels
   b. Cours et tutos (API client-serveur et patrons de communication)
   c. TP (API REST, microservices Flask et OpenAPI)
2. **12 octobre**
   a. **Quizz** et discussion sur les réponses
   b. Cours et tutos (RPC et gRPC)
   c. TP (gRPC)
3. **10 novembre**
   a. **Quizz** et discussion sur les réponses
   b. Cours et tutos (GraphQL)
   c. TP (GraphQL)
4. **16 novembre**
   a. **Quizz** et discussion sur les réponses
   b. Travail en groupes élargis sur une thématique (forme à définir :TD, TP, classe renversée)
   c. En parallèle **présentation de vos codes** par groupes de 2 des 3 TP précédents
   d. Restitution de la séance par présentation et discussion

Projet

- Développer un système de collecte et de restitution de données
- Patron de communication **publish/subscribe** avec `MQTT`
- Base de données **noSQL** `REDIS`
- Programmation des (micro)services en `Go`
- Animé par un intervenant extérieur : **Laurent Guérin (Capgemini France)**

Séances :

- 24 novembre
- 30 novembre
- 4 janvier
- 11 janvier : séance en autonomie totale
- 25 janvier : **soutenance** pour présenter votre projet

Séance sur les patrons architecturaux

Avant la soutenance de vos projets, une séance en plus le **18 janvier** pour aborder les **patrons architecturaux** de façon plus générale !

- Cours
- TD/TP en groupes élargis

Compétences évaluées

- **CG1** : Comprendre et analyser, synthétiser un problème et/ou une situation complexes
  - Quizz
- **CG2** : Résoudre un problème complexe en alliant théorie et pratique
  - Codes des TPs
- **CG3** : Concevoir et réaliser des systèmes et des organisations
  - Projet
- **CG9** : Communiquer
  - Soutenance de projet

Quelques rappels

HTTP

WEB

1. Connect to www.wooclap.com/JXQNSR
2. You can participate

SMS

1. Not yet connected? Send @JXQNSR to 06 44 60 96 62
2. You can participate

HTTP

- HTTP for “HyperText Transfer Protocol”
- Développé par Tim Berners-Lee au CERN (Suisse) avec d’autres concepts qui ont servi de base à la création du World Wide Web comme le HTML et l’URI
- Protocole de la couche applicative du modèle OSI
- Protocole de communication client-serveur / requête-réponse

Les versions de HTTP

- **[1991 HTTP/0.9]** GET sur du html uniquement
- **[1996 HTTP/1.0]**
  - sans connexion (connexion coupée à chaque fin de requête)
  - sans statut (client et serveur, s'oublient mutuellement d'une connexion à l'autre)
  - GET sur tout type de fichiers
- **[1997 HTTP/1.1]**
  - connexions persistantes
  - pipelining (envoi d'autres requête avant la réponse)
  - POST/PUT (envoi de contenu au serveur)
  - TRACE (suivi du chemin du client au serveur)
  - caching (conserver du contenu en mémoire tampon)
  - ...
- **[2015 HTTP/2]** - performance !
  - données binaires au lieu de fichiers texte
  - requêtes parallèles du côté client et serveur
  - compression des en-têtes
  - PUSH (envoi en cash client de données à l'avance)
- **[2020 HTTP/3 (draft)]** UDP au lieu de TCP

Méthodes (à partir de HTTP/1.1)

Méthode	Description
GET	Requête de la ressource située à l'URL spécifiée
POST	Envoi de données au programme situé à l'URL spécifiée
PUT	Envoi de données à l'URL spécifiée
DELETE	Suppression de la ressource située à l'URL spécifiée
HEAD	Requête de l'en-tête de la ressource située à l'URL spécifiée
...	...

JSON & YAML

WEB

1. Connect to www.wooclap.com/JXQNSR
2. You can participate

SMS

1. Not yet connected? Send @JXQNSR to 06 44 60 96 62
2. You can participate

JSON & YAML

JSON
- JavaScript Object Notation -
Requêtes de services
Bases de données NoSQL

YAML
- Yet Another Markup Language - (version 1.0)
- YAML Ain't Markup Language - (version 1.1)
Fichiers de configuration
Spécifications

- Formats standardisés de données textuelles
- Représentation structurée de l'information textuelle
- Facilement lisible par un être humain
- Formats sérialisables (communication entre différents langages)
- Formats comparables à `XML` mais plus légers

Démo

JSON & YAML

http://www.yamllint.com/

https://www.json2yaml.com/

https://www.youtube.com/watch?v=1uFVr15xDGg