

Client de chat distribué

- CHAHI Hamza
- LAMINE Marouane
- PHILY Tom
- ROSTAN Estelle
- TRAORE Aichatou

PLAN

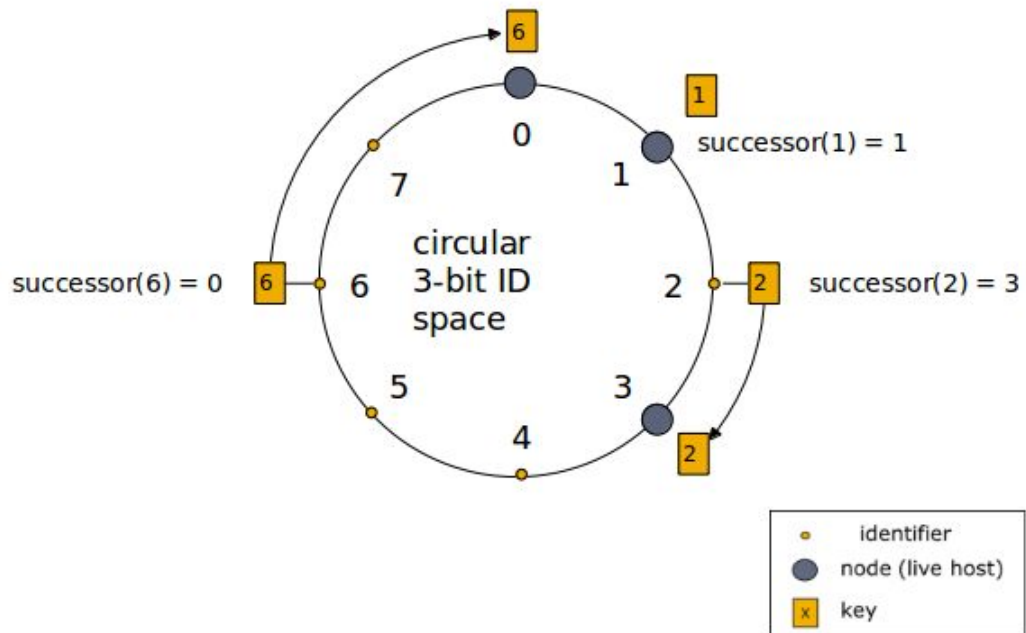
- ▶ Introduction
- ▶ Le Chord (simple/renforcé)
- ▶ Les Salons
- ▶ L'annuaire
- ▶ Fonctionnalités additionnelles

Introduction

Objectif : mettre au point un service de chat distribué P2P.

- ▶ Travail sur Github
- ▶ Projet Maven
- ▶ Architecture Chord
- ▶ Plateformes utilisées : Eclipse, IntelliJ, NetBeans
- ▶ Plugins utilisés : jersey-json, junit

Le Chord



Communication TCP

- Envoyer/recevoir des données en utilisant les sockets (java.net)
- Ecouter les ports des noeuds
- Traiter les données en Byte (plus général)



Les requêtes

- Des requêtes pour quitter/rejoindre le chord.
- Échanger des requêtes entre les pairs.
- Traitement des données sous forme de chaînes de caractères (différencier entre les Types de requêtes).
- Traiter les requêtes sous forme de JSON.
- Envoyer des requêtes avec les objets des pairs (jersey-json).

Les fonctionnalités :

- ▶ Création salon
- ▶ Joindre un salon
- ▶ Transmettre un message dans un salon
- ▶ Lire le dernier message d'un salon
- ▶ Lire tous les messages d'un salon
- ▶ Lister les salons créés

L'annuaire

- ▶ L'intérêt de l'annuaire : chaque "user" peut connaître les personnes connectées
- ▶ Au départ, un annuaire simplifié : chaque noeud créé connaît le tout premier noeud créé
- ▶ L'annuaire amélioré :
 - ▷ Renvoie la liste des noeuds connectés
 - ▷ Renvoie le dernier noeud connecté
 - ▷ Renvoie la liste des salons disponibles

Fonctionnalités à améliorer

Le chord amélioré:

- revoir l'implémentation de la stabilisation des FingerTables, au moment où un pair rejoint/quitte le chord

Les salons:

- L'envoi de message dans les salons

MERCI ET
MAINTENANT DEMO