

Garage : jouer dans la cour des grands
quand on est un hébergeur associatif
(ou pourquoi on a décidé de réinventer la roue)

Quentin Dufour & Alex Auvolat

Mercredi 2 décembre 2020



Pourquoi vous n'hébergez pas vos
fichiers chez vous ?

Le modèle du cloud...



- + intégrité : plus de perte de données
- + disponibilité : tout le temps accessible
- + service : rien à gérer

changement des comportements

Le modèle du cloud...



- + intégrité : plus de perte de données
- + disponibilité : tout le temps accessible
- + service : rien à gérer

changement des comportements

...et son prix



- matériel couteux et polluant
- logiciels secrets
- gestion opaque

prisonnier de l'écosystème

Ressemble à du cloud...



- + compatible avec les apps existantes
- + fonctionne avec le mobile
- + s'adapte aux habitudes prises

Ressemble à du cloud...



- + compatible avec les apps existantes
- + fonctionne avec le mobile
- + s'adapte aux habitudes prises

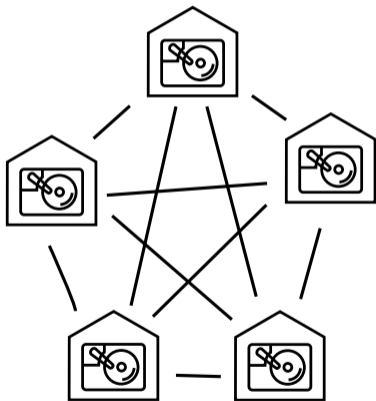
...fait du P2P



- + contrôle de l'infrastructure
- + transparent code libre
- + sobre fonctionne avec de vieilles machines à la maison

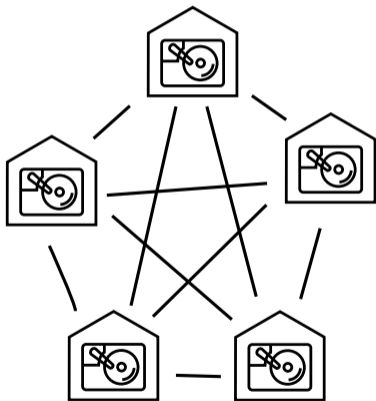
Mais donc, c'est quoi Garage ?

Un système de stockage distribué



Mais donc, c'est quoi Garage ?

Un système de stockage distribué

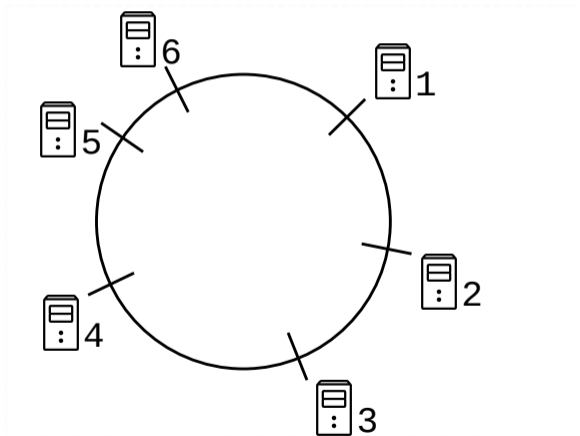


qui implémente l'API S3

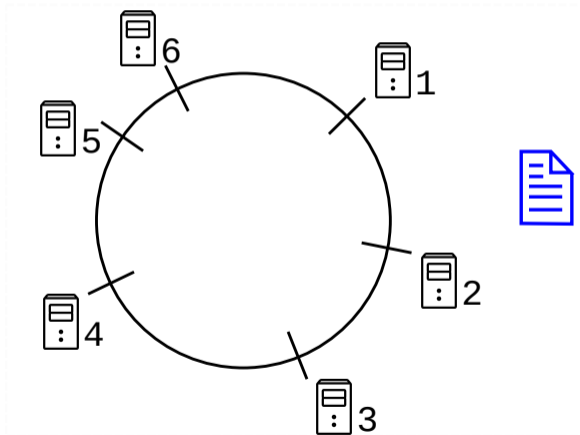


Amazon S3

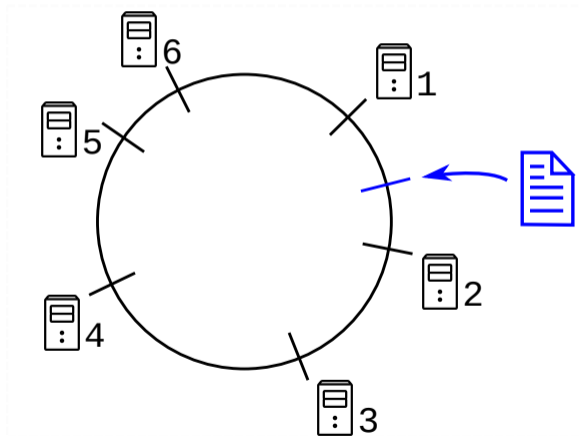
Comment répartir les fichiers sur les différentes machines ?



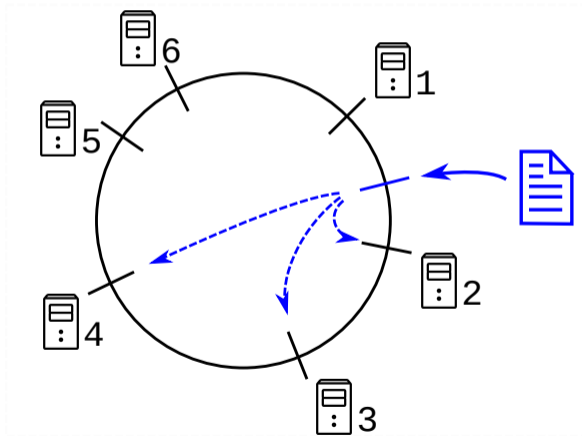
Comment répartir les fichiers sur les différentes machines ?



Comment répartir les fichiers sur les différentes machines ?

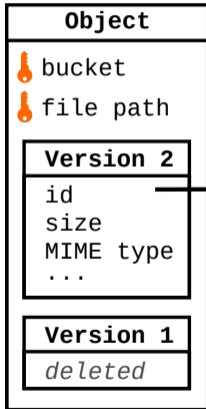


Comment répartir les fichiers sur les différentes machines ?

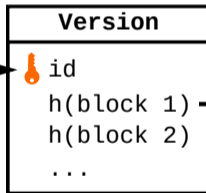


Garage Internals : 3 niveaux de consistent hashing

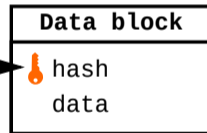
Objects table



Versions table



Blocks table



Garage utilise un modèle de cohérence relativement faible :

- Objets répliqués 3 fois, quorum de 2 pour les lectures et les écritures
→ cohérence **“read your writes”**

Garage utilise un modèle de cohérence relativement faible :

- Objets répliqués 3 fois, quorum de 2 pour les lectures et les écritures
→ cohérence **“read your writes”**
- Types de donnée CRDT + mécanisme d'anti-entropie
→ cohérence **à terme** (eventual consistency)

Garage utilise un modèle de cohérence relativement faible :

- Objets répliqués 3 fois, quorum de 2 pour les lectures et les écritures
→ cohérence **“read your writes”**
- Types de donnée CRDT + mécanisme d'anti-entropie
→ cohérence **à terme** (eventual consistency)
- Cela s'applique pour chaque fichier individuellement :
pas de linéarisabilité ou de cohérence causale entre les opérations
sur des fichiers différents

Garage utilise un modèle de cohérence relativement faible :

- Objets répliqués 3 fois, quorum de 2 pour les lectures et les écritures
→ cohérence **“read your writes”**
- Types de donnée CRDT + mécanisme d'anti-entropie
→ cohérence **à terme** (eventual consistency)
- Cela s'applique pour chaque fichier individuellement :
pas de linéarisabilité ou de cohérence causale entre les opérations
sur des fichiers différents
- **Avantage** : convient bien à un déploiement géodistribué (multi-datacenter)

Garage est entièrement écrit en Rust !

Points forts :

- Langage compilé, très rapide
- Typage fort, beaucoup de sécurités
- Le meilleur de plusieurs paradigmes: fonctionnel, orienté objet, impératif
- Un écosystème de bibliothèques très complet: serialisation, async/await, http, ...



Points faibles :

- Les temps de compilation...
- Compliqué à apprendre