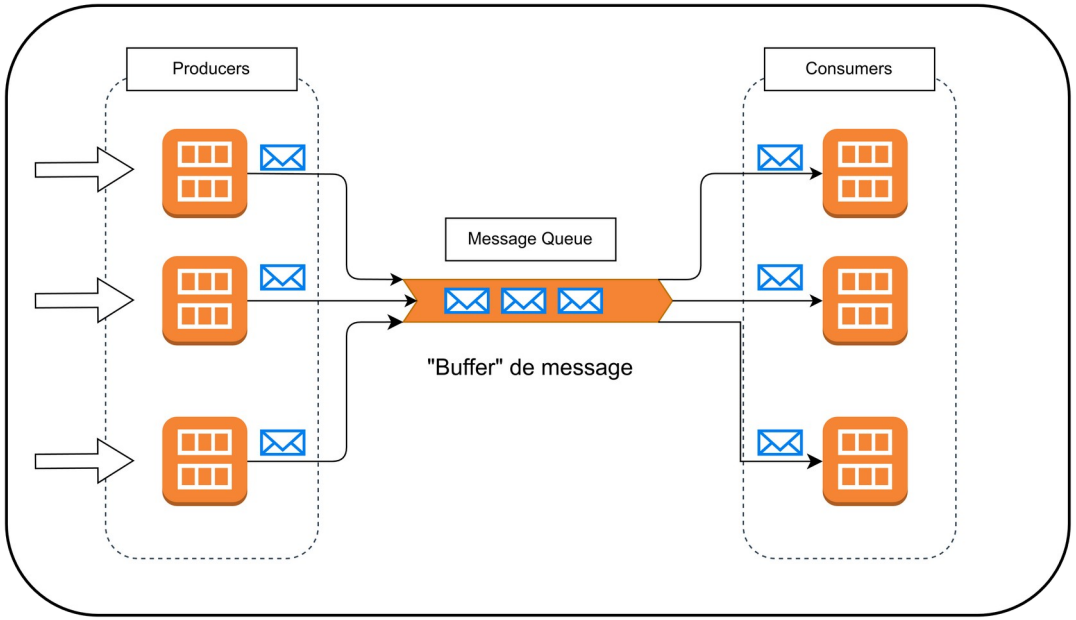
Architecture évènementielle avec Apache Pulsar

Pourquoi Pulsar ?

Message queue



Message Queue : riche fonctionnellement

Programmation évènementielle / La communication interprocessus

Mécanisme d’aiguillage

Consumer

Logique de rejeux / Dead Letter Queue

Gestion de priorité

Variété de protocole

Authentification

IHM d’administration très fine

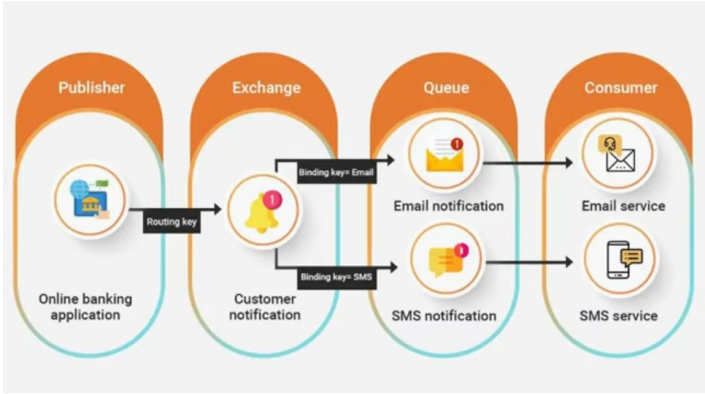
Mais :
mécanisme HA : Leader / Follower

scalable verticalement

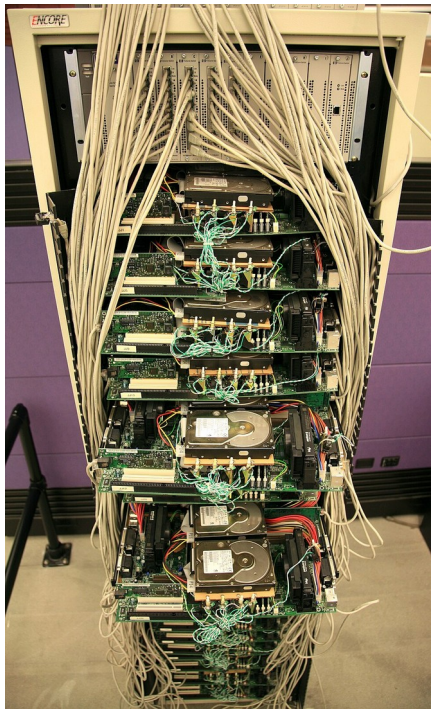
RabbitMQ

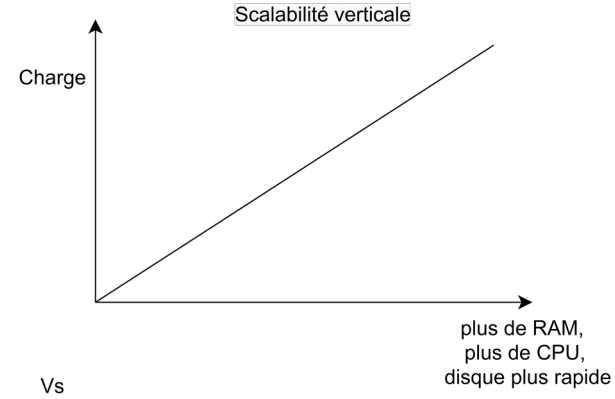
APACHE ACTIVEMQ

APACHE ARTEMIS™

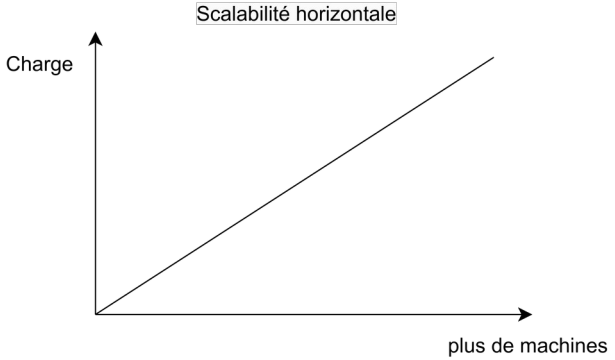


« Big Data »

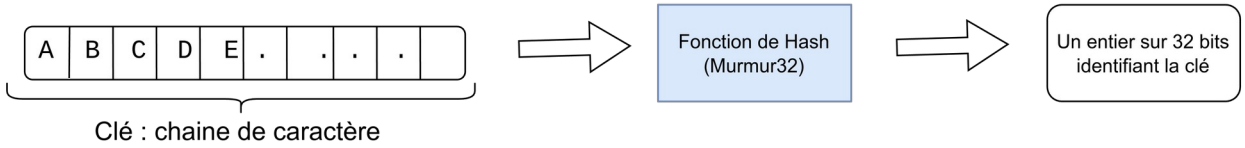


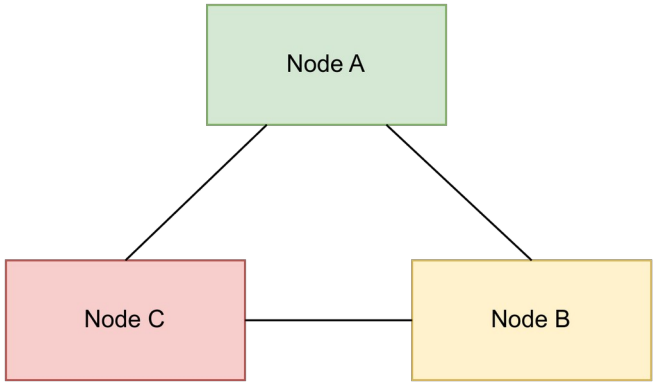

Scalabilité verticale
Charge
plus de RAM,
plus de CPU,
disque plus rapide

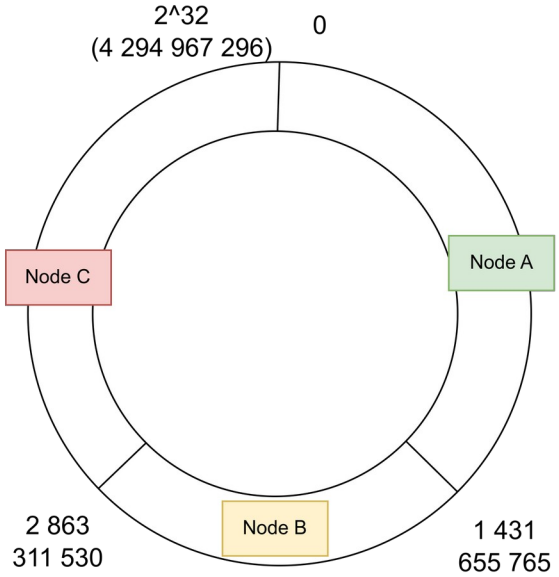
Vs


Scalabilité horizontale
Charge
plus de machines

Consistent Hashing : algorithme de partition automatique







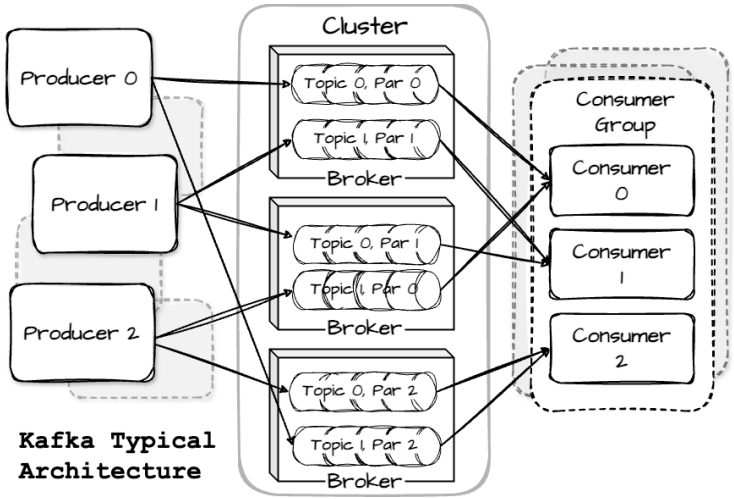
Data broker / Streaming plateform

Écriture et lecture distribuée sur plusieurs machines

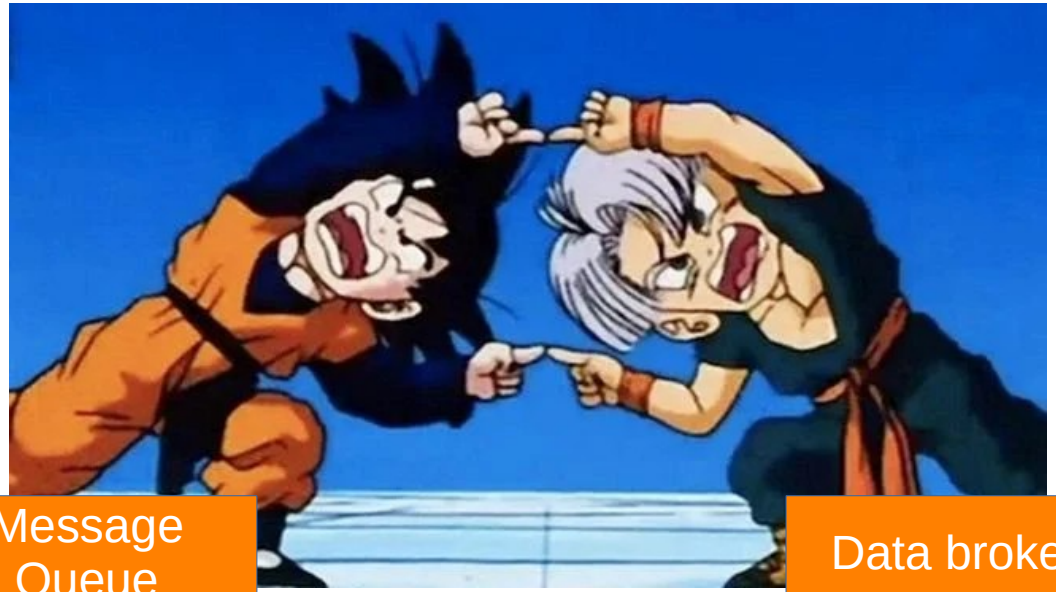
Dédié au traitement en masse de donnée :
- Gère les pics de charge
- Million de messages par secondes

Les messages sont délivrés dans l'ordre

Mais :
- peu de fonction : publish / subcribe seulement
- partitionnement « statique » au topic



Fuuuusiiiionnn



Message Queue

Data broker

Pulsar :
data broker et message queue

Pulsar

Inventé par Yahoo en 2012, open sourcé en 2016

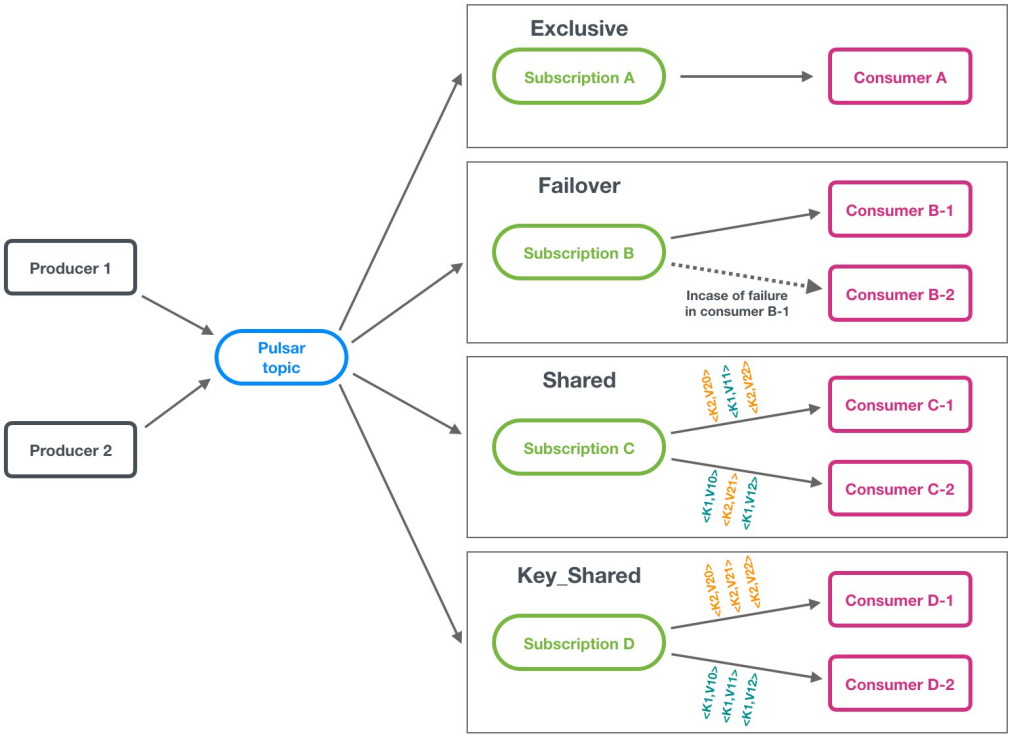
- Écriture / lecture de message distribuées sur plusieurs machines
- Capable de tenir les volumes de message équivalent à Kafka
- Garantie de délivrer les messages dans l’ordre
- Client officiel pour Java, Python, C++, Go, node.js, c#
- Persistent / Non persistent
- Acknowledge / negative acknowledge / acknowledge timeout
- Retry letter topic / dead letter topic
- Delayed Message
- Géo réplication
- Tier Storage
- Authentification / possibilité d'isoler les données entre équipe (Multi tenant)
- Function

Concept

- Topic
  - Partition
- Message
  - Key
  - Value

- Client :
  - Producer
  - Consumer
    - Ack / negative ack
    - Subscription
      - Cursor
  - Reader
  - Tableview

Pulsar : Topic Subscription



Demo : Subscription Key Shared

Key Shared : recommandation

- Attention aux hots spots : rediriger les clés vers un autre topic
- Cardinalité des clés : ok si batching est activé (par défaut)
- Ajout de consumers = redistribution du « ring » : peut créer des duplicas
- Negative ack : change l’ordre des messages temporairement
- Ne pas hésiter à augmenter le nombre de thread dans le consumer

Le projet Open Source

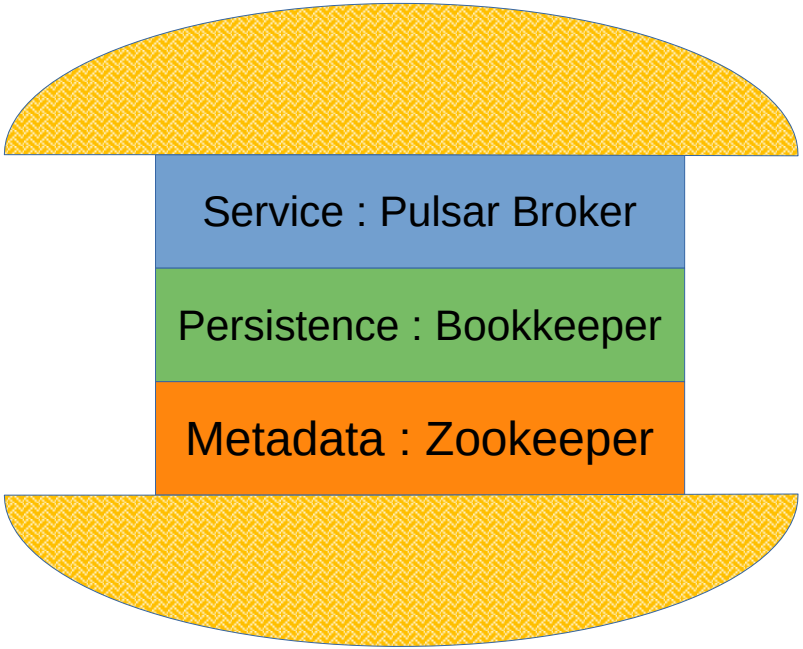
https://pulsar.apache.org/

- Top projet de la fondation Apache depuis 2018
- 15k stars sur Github
- Communauté active sur Slack, Github...

- Projet Solide :
  - Rédaction de PIP pour toutes les améliorations
  - Branche LTS et Release

Architecture interne

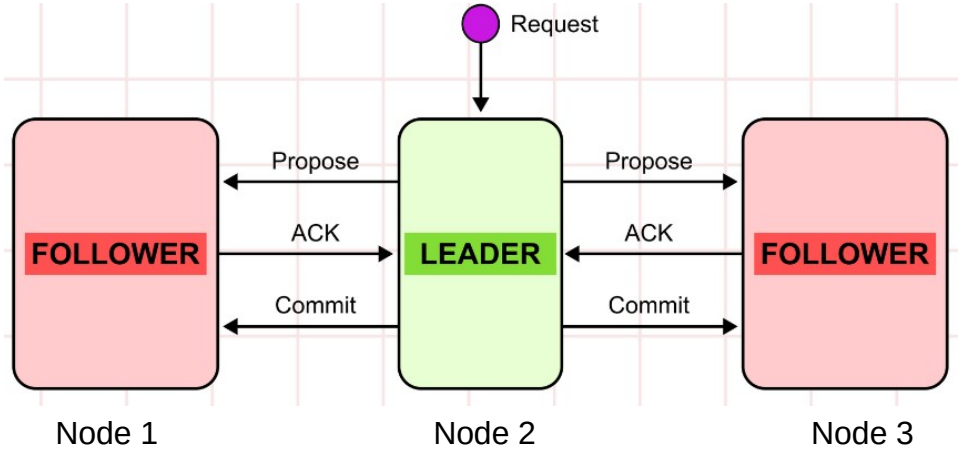
Pulsar : Architecture en 3 couches



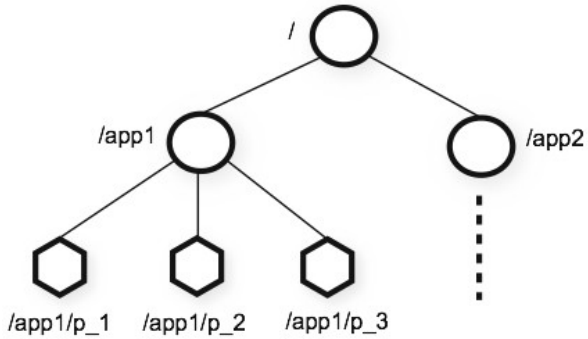
Apache Zookeeper

Technologie de référence « Big Data » qui résous le problème de l'élection de leader en système distribué.

Ce socle de coordination est nécessaire pour le fonctionnement de Apache Pulsar et Apache Bookkeeper :
- gestion la configuration du cluster Pulsar et Bookkeeper
- tache de coordination entre les nœuds du cluster



Stocke des metadatas sous forme d'arbre (Z-node)



En phase d'être remplacé par oxia en version 5.0

Bookkeeper

Distributed write-ahead log (WAL)

Réplication par écriture simultanée

Stockage persistant des messages et des positions des subscriptions (*cursors*) des consumers

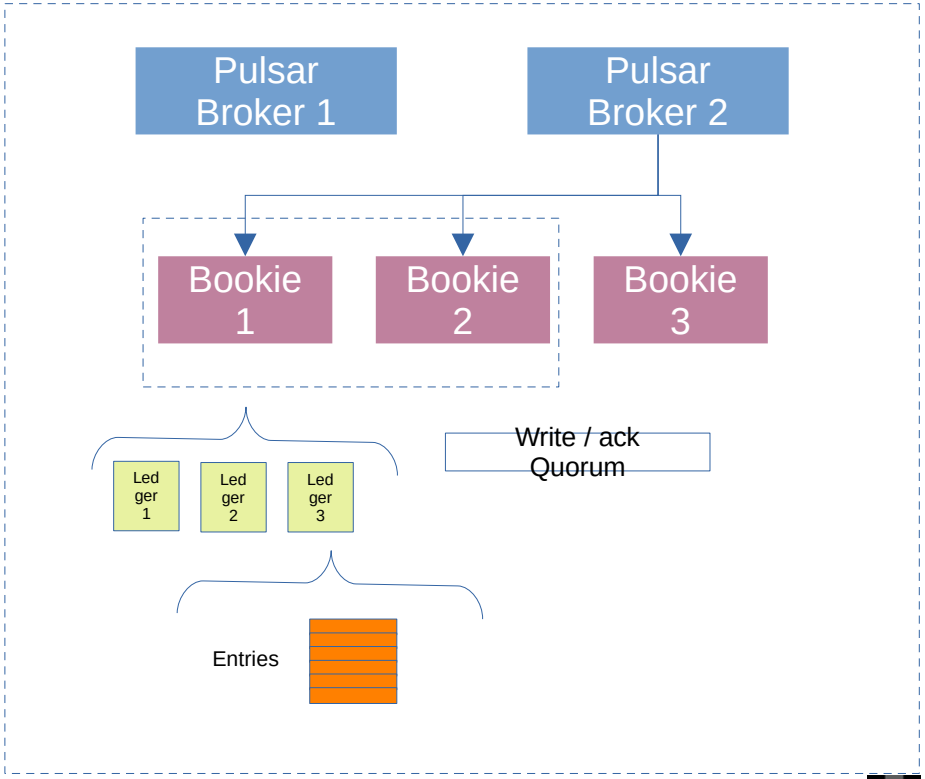
Message “entries” stockée dans des *ledgers*

Ledgers sont en « append-only »

Le ledger est supprimée une fois les données lues

HA dans le client écriture sur plusieurs nœud configurable

Lecture consistante des données sur le cluster

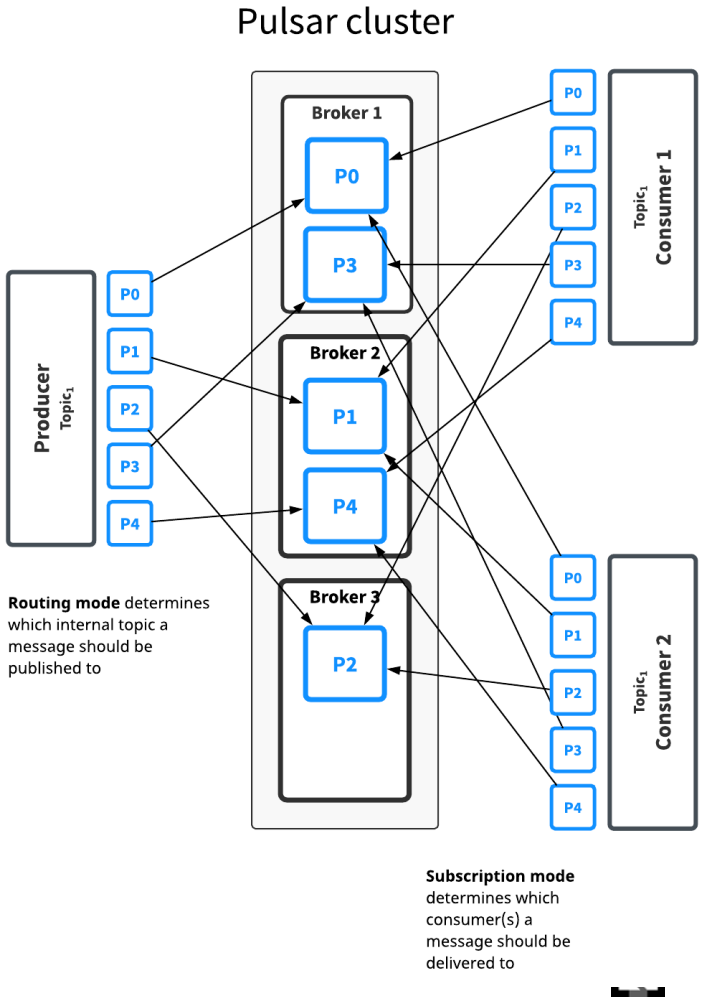


Pulsar Broker

- La couche service
  - Gère le fonctionnel Apache Pulsar
  - Permet la répartition des topics sur plusieurs noeud
  - Pas de persistance tout est stocké dans Bookkeeper (data) et Zookeeper (metadata)

Topic Partitioning

- Ce sont simplement des topics avec un préfixe “topic-name-partition-0...”
- Permet la scalabilité sur différent noeud
- Choix de partition par consistent hashing coté producer
- Possibilité d’augmenter le nombre de partitions dynamiquement



Load balancing & High Availability

- Exemple de stratégie de « Load balancing » dynamique personnalisable (bundle unloading)
  - ThresholdShedder (par défaut): utilisation des ressources (CPU, RAM, réseau) au dessus de la moyenne du cluster
  - OverloadShedder : dépassement de seuil de ressources en dur
  - UniformLoadShedder : redistribution lors que le broker plus sollicité et le moins sollicité sont déséquilibrés
- Persistence des donnée sur Bookkeeper
  - Les messages sont stockés selon un replication factor (RF)

Pulsar en pratique

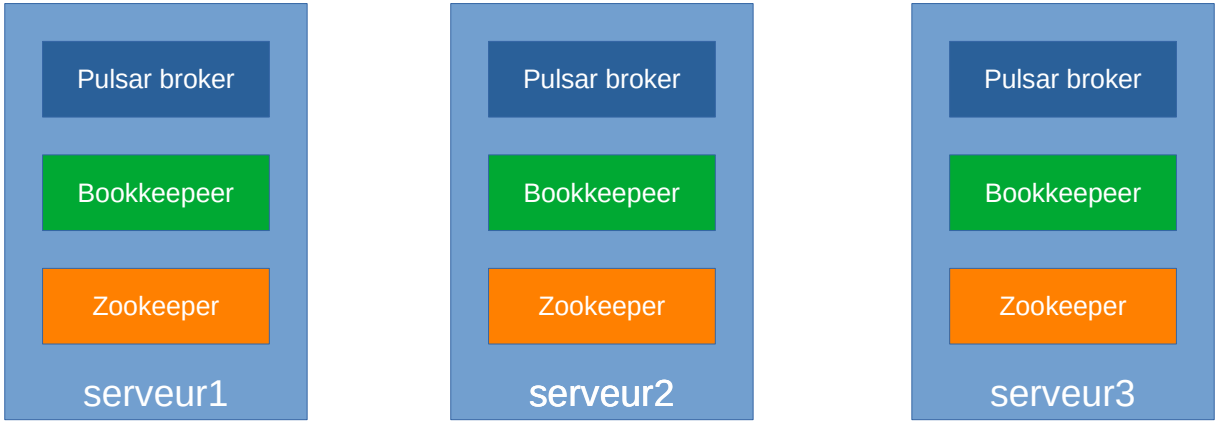
Pulsar en Prod

- 3 Machine Bare metal (voir même public cloud)
- Kubernete :
  - Helm Chart Officiel : https://pulsar.apache.org/docs/4.1.x/deploy-kubernetes/
  - Operator
    - DataStax Kaap (Operator) : https://docs.datastax.com/en/kaap-operator/0.2.0/index.html
    - Streamnative (Propriétaire)
- En Saas : Stream native, Clever Cloud, Sur les plateformes Kubernete SaaS

- Metrics Prometheus

Pulsar en prod

- 3 machines Advance 1 OVH
  - AMD EPYC 4245P 6c / 12t 3,9 GHz / 5,4 GHz
  - RAM 32GB
  - Disque NVMe 2*960Gb
- Déploiement par Ansible



Fonctionnel manquant et « Workaround »

- une solution pour savoir quels sont les messages en cours de traitement
  - Utilisation Reader + API
- une solution de priorisation de message
  - Développement d’une solution interne

Conclusion

- Un projet Open Source mature
- Alternative si vous utilisez à AWS MSK, AWS MQ, GCP PUB/SUB ou GCP Dataflow
- Création facile de chaînes de traitement de flux de donnée en parallèle (Subscription)
- Augmentation du volume prévisible à court / moyen terme
- Scalabilité et cloisonnement des traitements par identifiant métier (key_shared, partition)
- Fonctionnel avancé : délais, fonction, authentification, ACL et tier storage

Source

- La doc du site (fabuleuse) : https://pulsar.apache.org
- Conférence de Lari Hotari sur Key Shared
  https://www.youtube.com/watch?v=AkSGvYP4r88
- Conférence de Julien Jakubowski au Devoxx France
  https://devconf.net/index.php/talk/apache-pulsar-enfin-une-alternative-a-kafk

Merci pour votre attention, avez vous des questions ?