



2018/07/03

投資程設科 陳宗霆

What is kafka?

最初是由Linkedin開發的分散式消息系統

2011 年開源，並轉由Apache軟體基金會管理

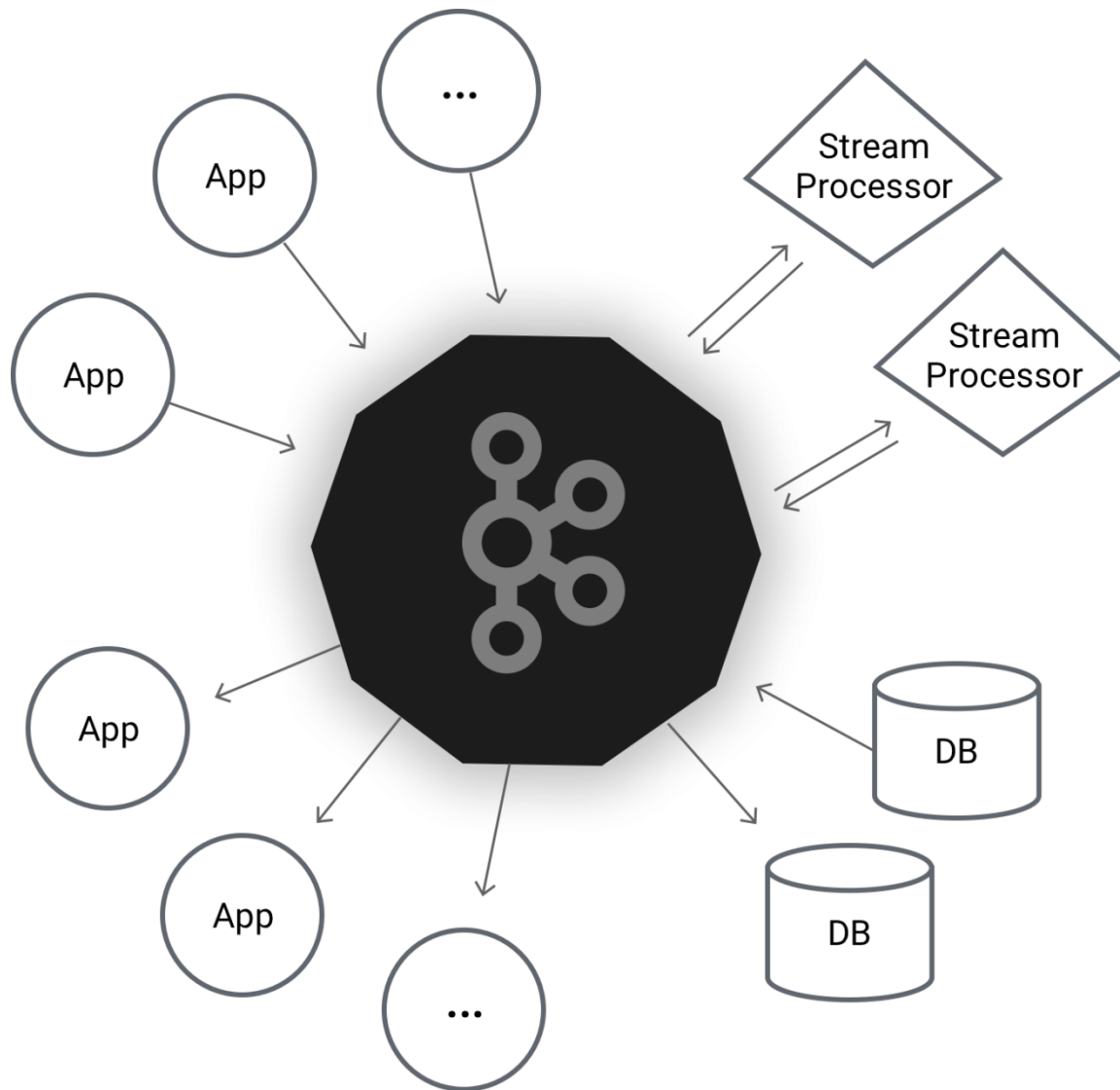
主要功能: 分散式系統的訊息處理
(Log整合、ETL、Stream 處理等)

主要特性: 高吞吐量、可擴展性、可複製、可容錯

What is kafka?

Apache Kafka® is a distributed streaming platform.

- Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.
- Store streams of records in a fault-tolerant durable way.
- Process streams of records as they occur.



Why Message Queue

建一條連線，用於傳遞訊息

Side A



Side B



Why Message Queue

建一條連線，用於傳遞訊息

Side A



Side B



訊息頻繁 => 連線放不掉

=> 當系統擴大，連線數量終究碰到上限

Why Message Queue

改用API溝通 (雖然稍慢，但沒連線問題!)

Side A



Side B



Why Message Queue

改用API溝通

Side A



Side B



Response 很久才回來

=> 不管Response (用multithread 發Request)

=> 資料順序出問題

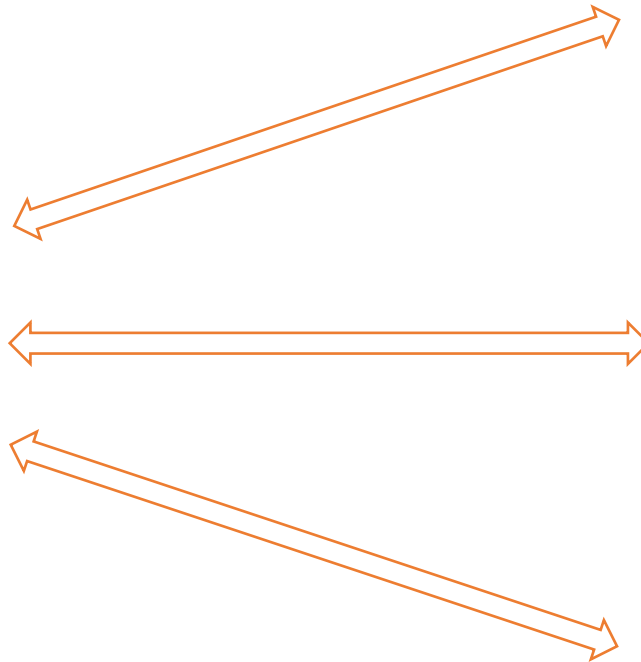
Why Message Queue

如果Loading 都在 Side B

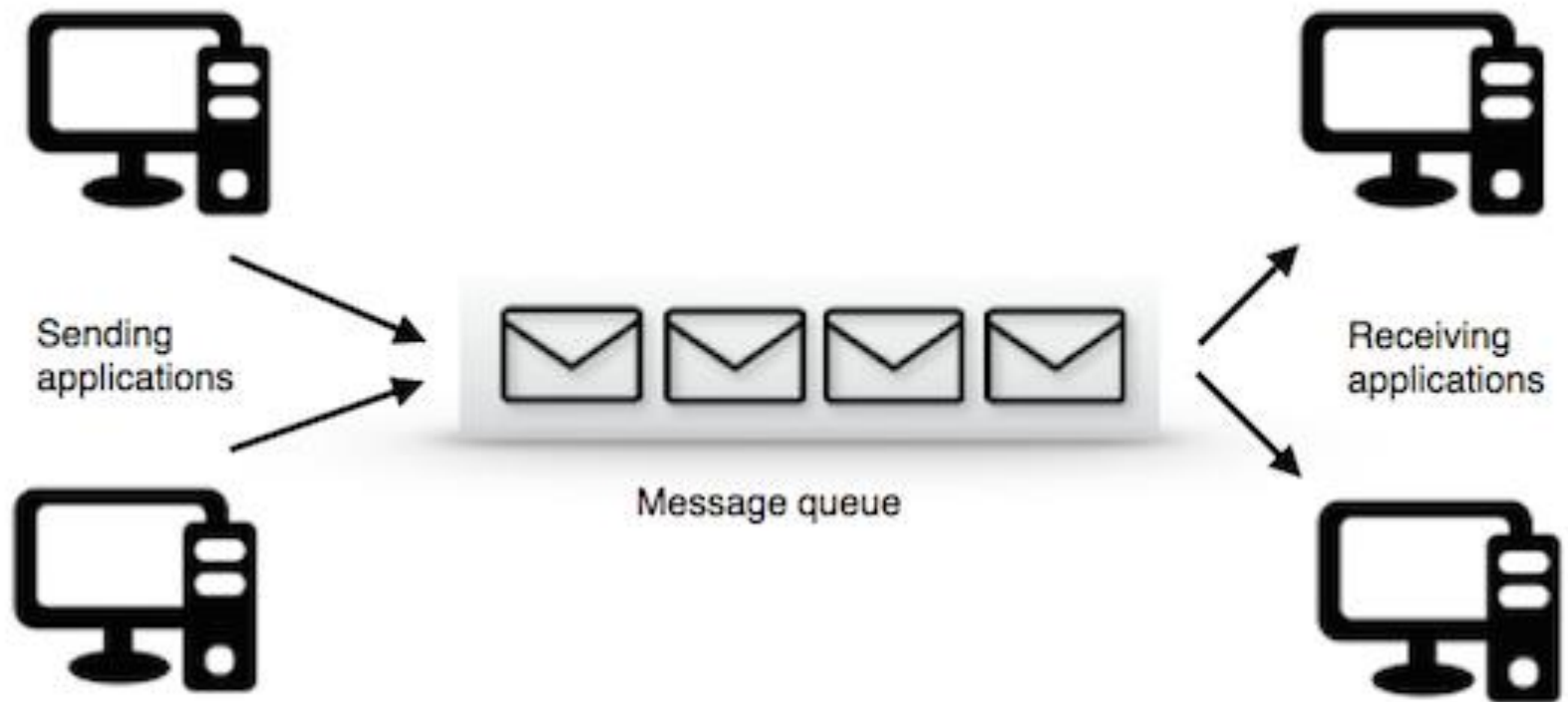
Side A

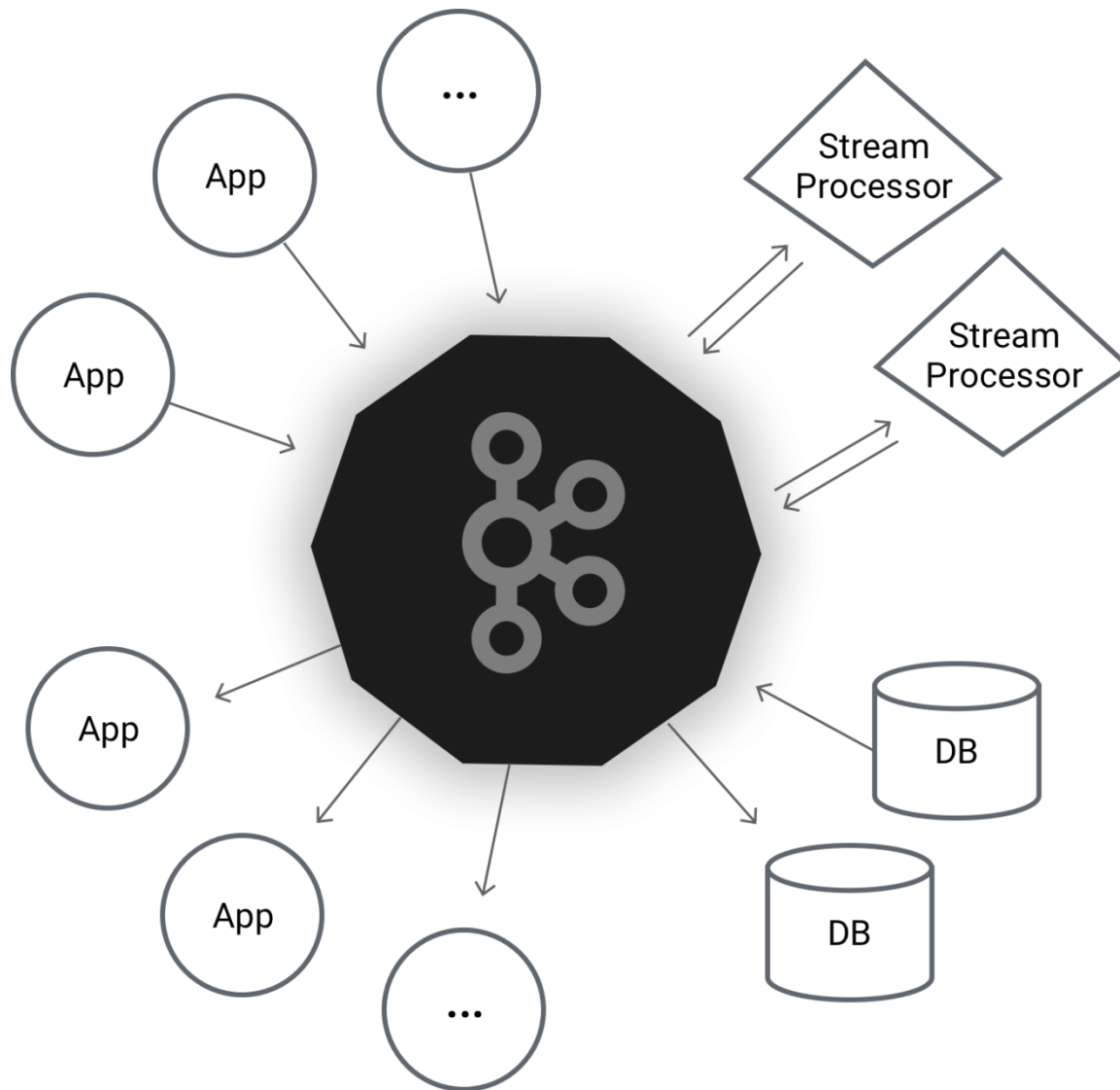


Side B

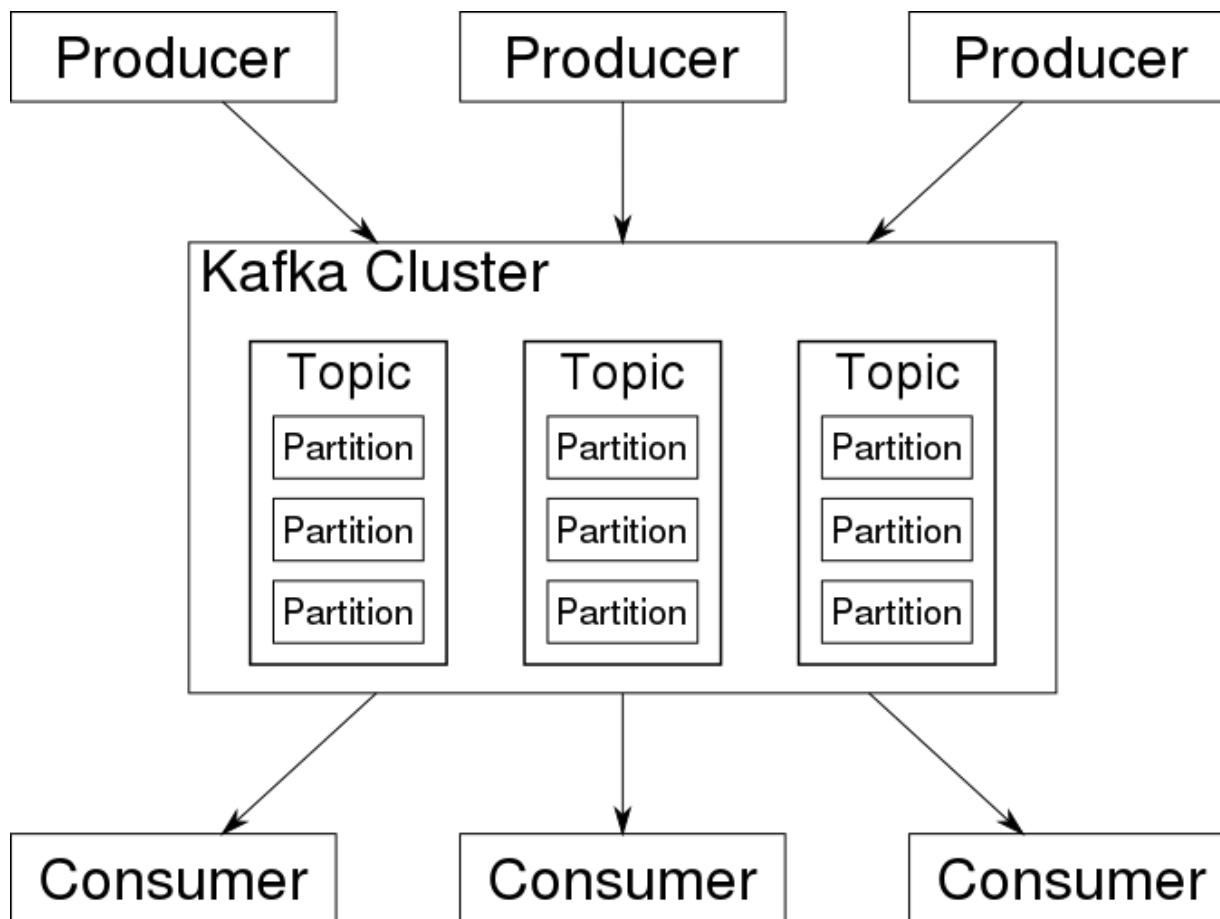


資料相依性、資料順序 被破壞了
同時有Load Balance問題





Kafka 架構



Kafka 成員

Producer

產生Message並送到Kafka

Consumer

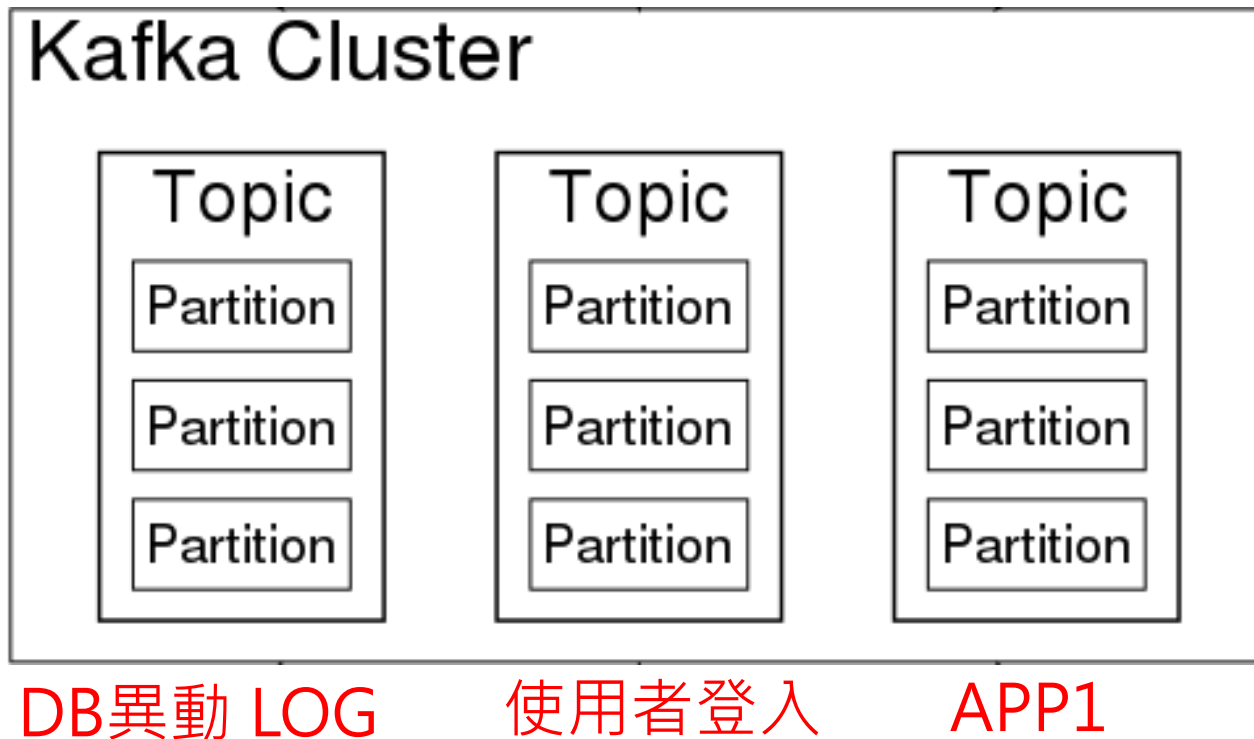
消費Kafka上的Message

Broker

Kafka Server, 每台Server都會有一個 broker id

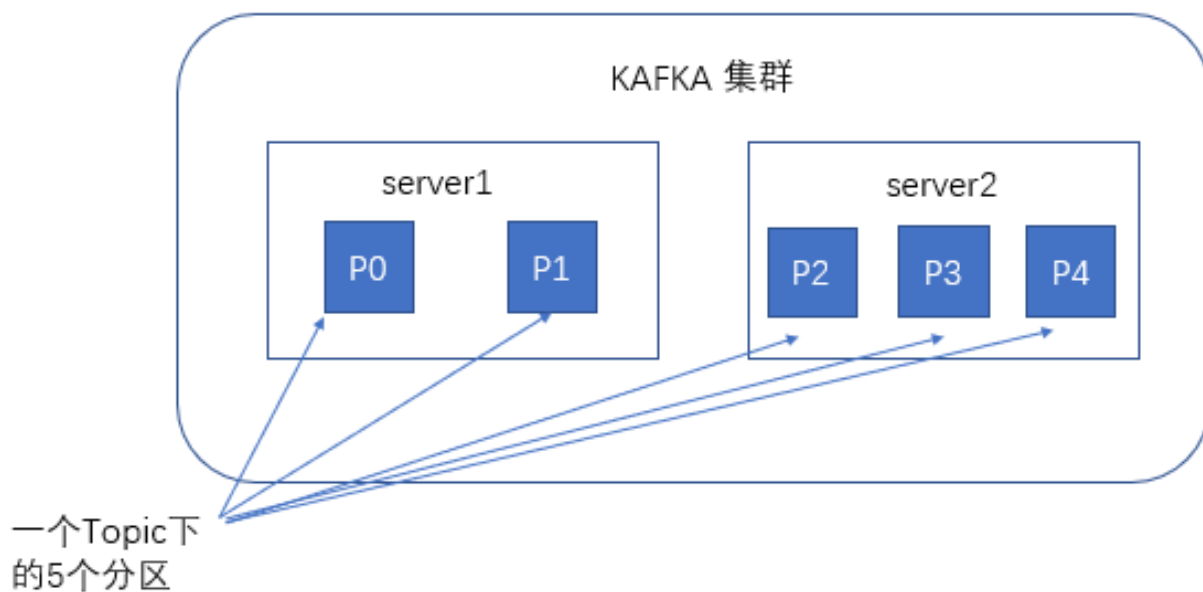
Topic (主題)

通常對應到業務



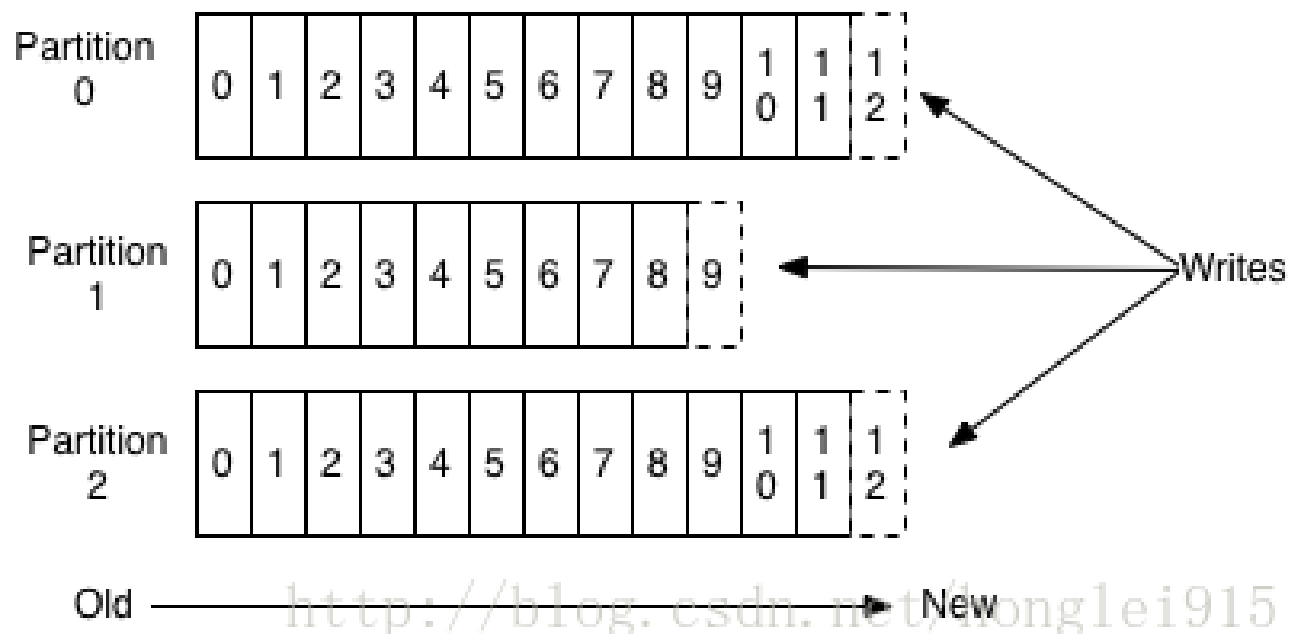
Partition

- 同個 topic，分散在不同partition

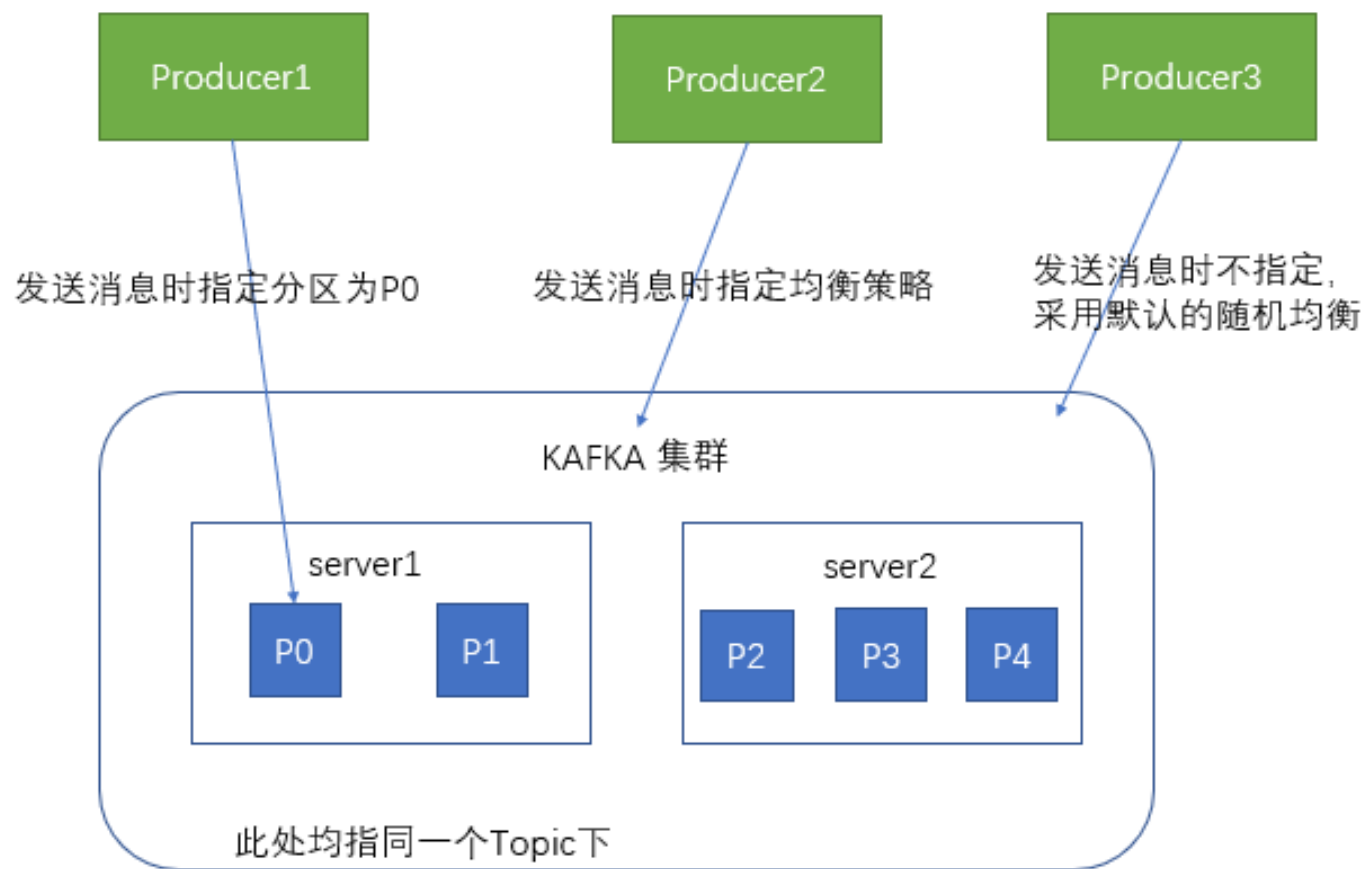


Offset

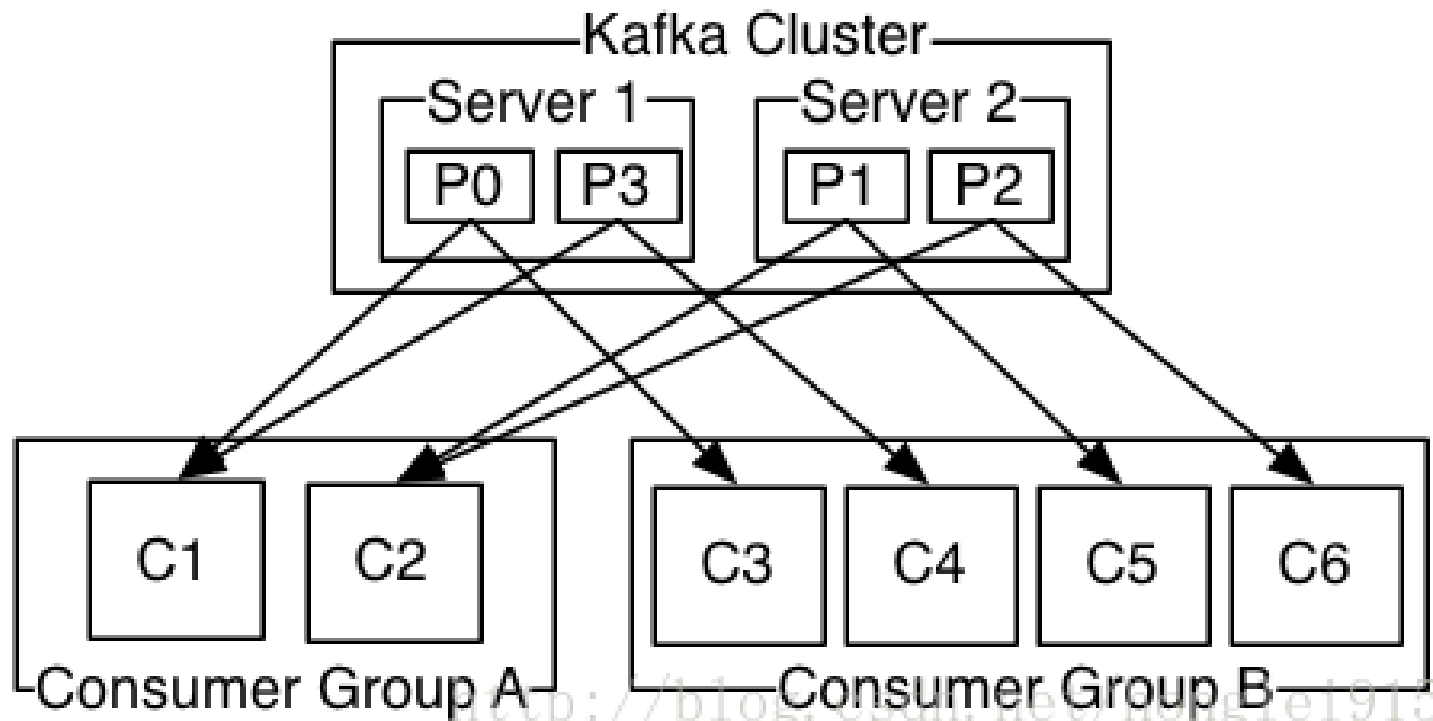
Anatomy of a Topic



Producer with Partition



Consumer Group



實作時間!!!

- 建起你的 kafka cluster

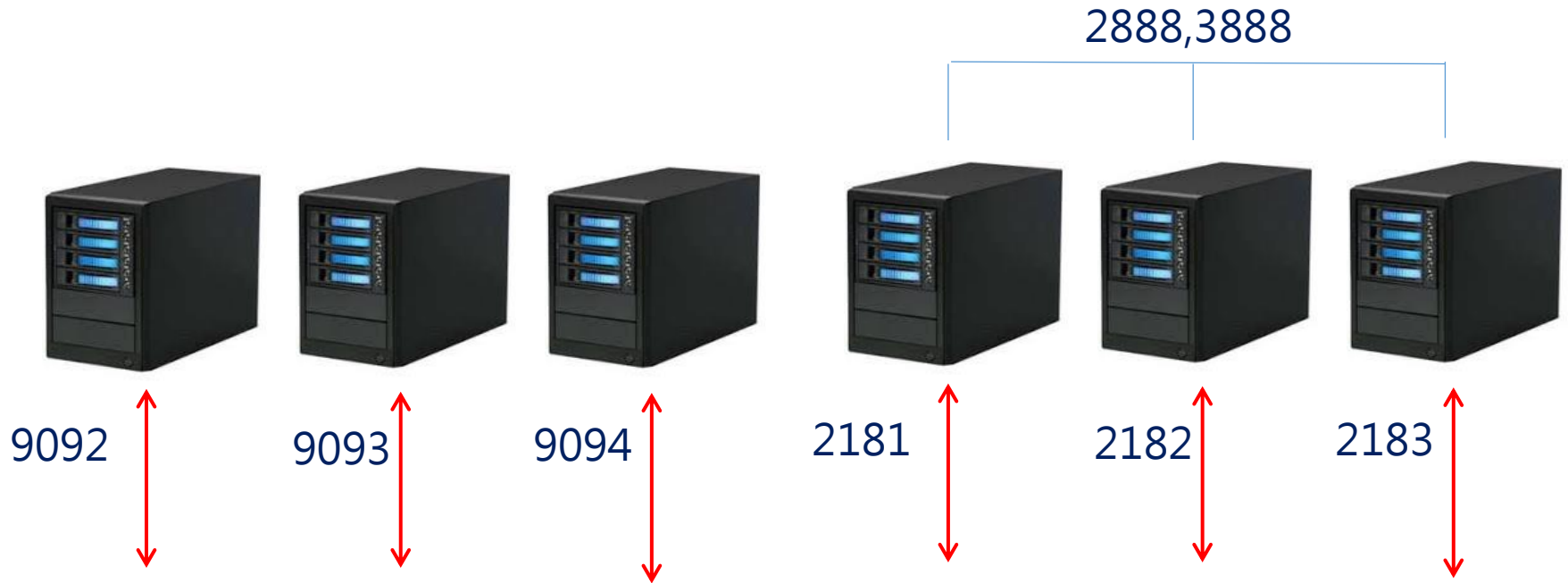
Docker-compose

```
kafka3:
  image: wurstmeister/kafka
  hostname: kafka3
  container_name: kafka3
  depends_on:
    - zoo1
    - zoo2
    - zoo3
  ports:
    - 9094:9092
  environment:
    KAFKA_LOG_DIRS: /kafka
    KAFKA_BROKER_ID: 3
    KAFKA_ADVERTISED_PORT: 9094
    KAFKA_ADVERTISED_HOST_NAME: localhost
    KAFKA_ZOOKEEPER_CONNECT: zoo1:2181,zoo2:2181,zoo3:2181

zoo1:
  image: zookeeper
  restart: always
  container_name: zoo1
  hostname: zoo1
  ports:
    - 2181:2181
    - "2888"
    - "3888"
  environment:
    ZOO_MY_ID: 1
    SERVERS: zoo1, zoo2, zoo3
```

Kafka

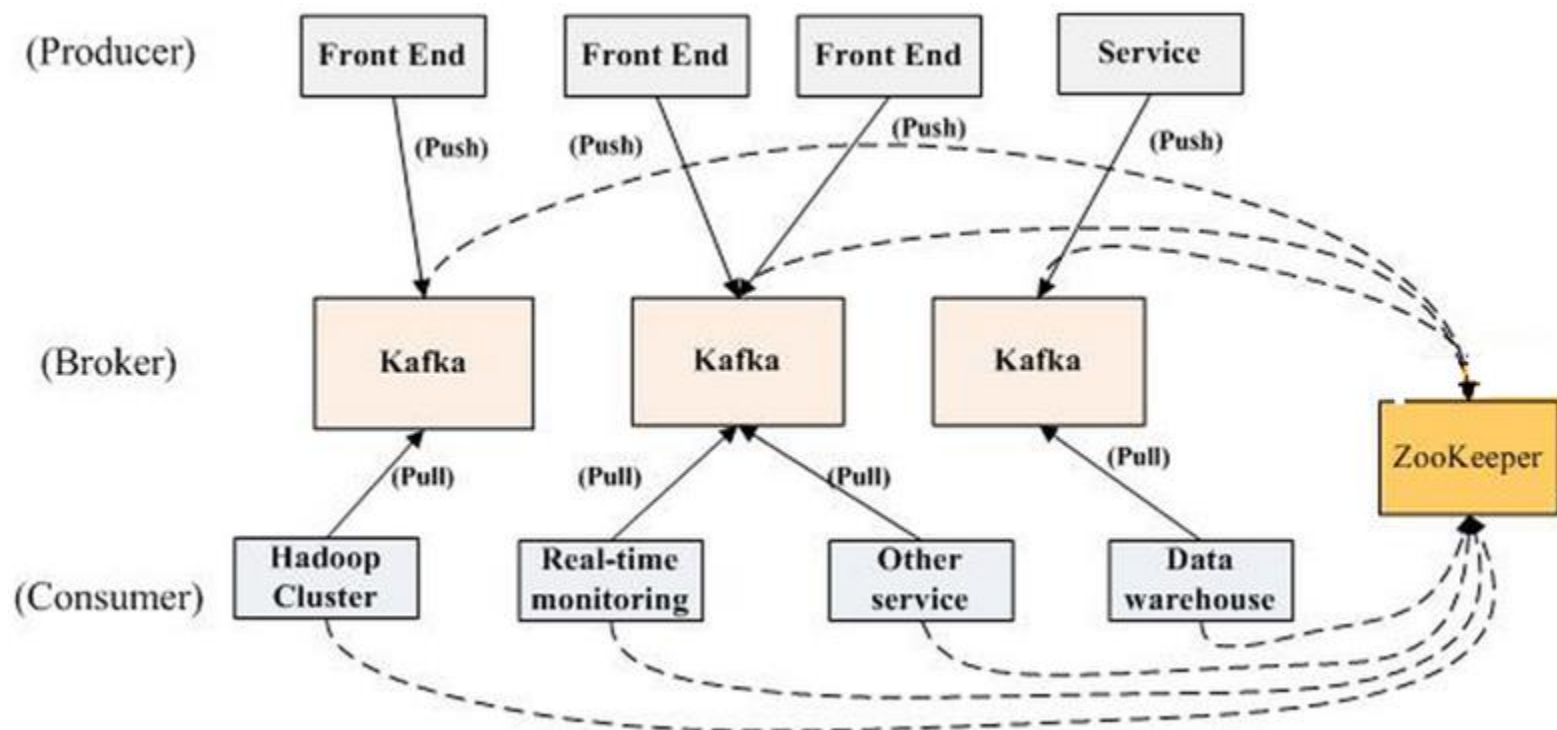
Zookeeper



Client

任務

- 確認Kafka broker id有正常註冊
- 透過 Kafka TOOL 建立Connection
- 建立TOPIC
- 建立Message



Recall

- Zookeeper
- Distributed file system

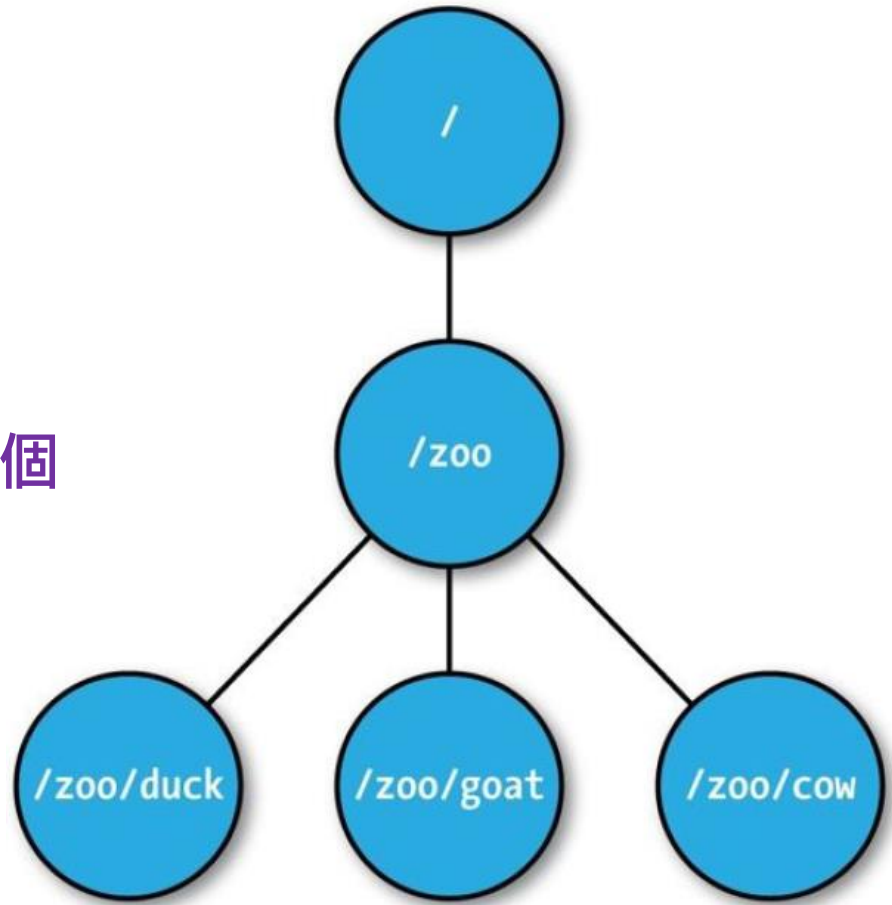
主要功能: 分散式系統的協調
(統一命名、狀態同步、集群管理、系統配置
管理)



Recall

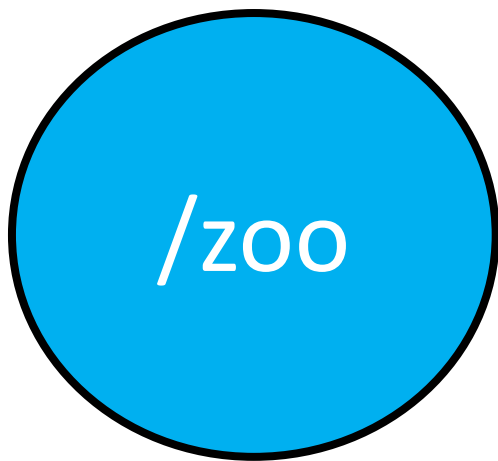
- Zookeeper

每個節點都是一個
znode



Recall

- Zookeeper



CreateMode類型

Ephemeral 臨時節點

Persistent 永久節點

Session關閉後是否保留

順序節點(自動加編號)

< path >01 , < path >02

Recall

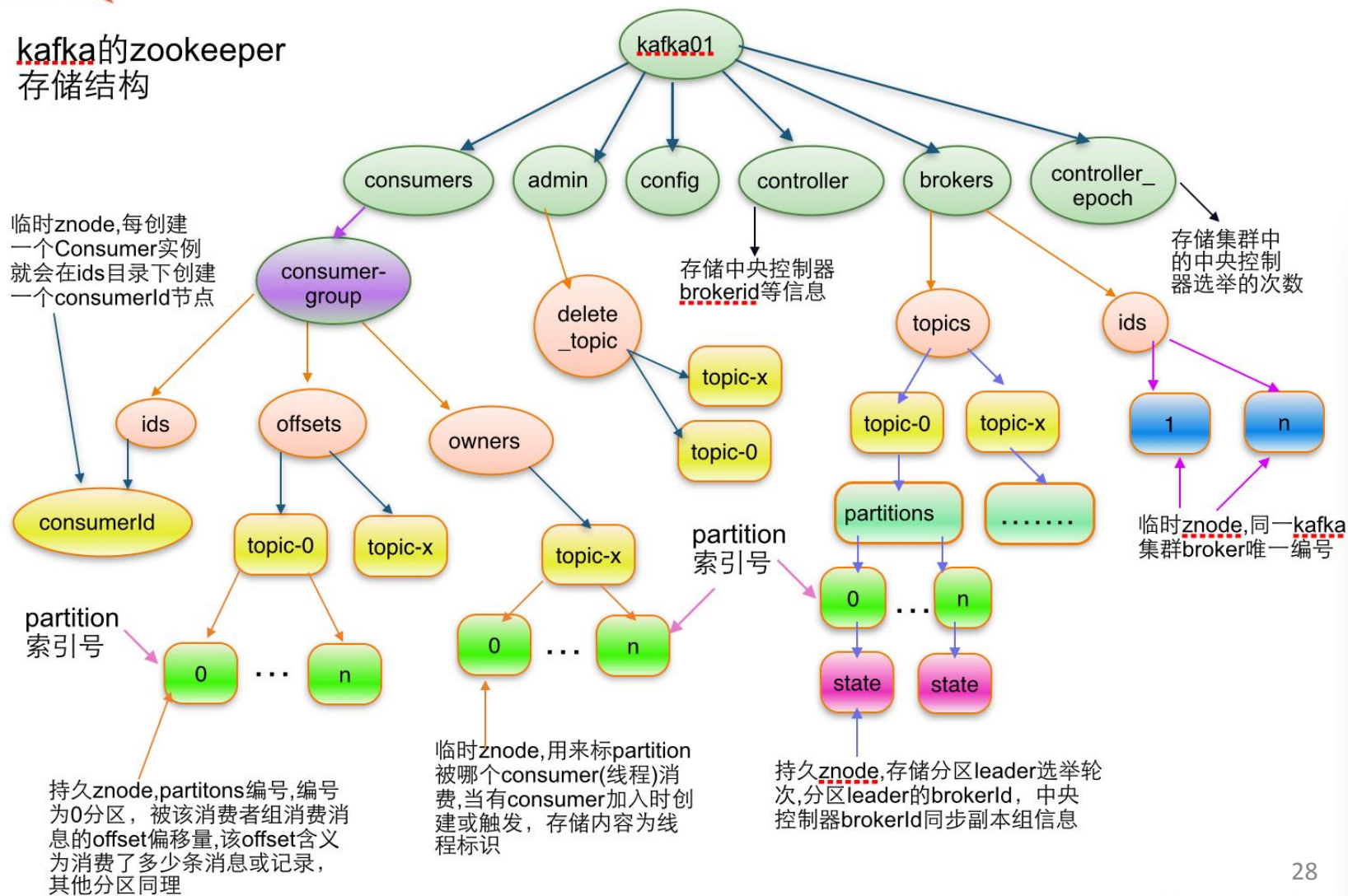
- Zookeeper

觀察模式Watches 可以使client端在某一個znode發生變化時得到通知

---	Watch trigger			
Watch creation	create znode	create child	delete znode	delete child
exists	NodeCreated	-	NodeDeleted	-
getData	-	-	NodeDeleted	-
getChildren	-	getChildren	NodeDeleted	NodeChildrenChanged

Kafka v.s. Zookeeper

kafka的zookeeper
存储结构



Kafka v.s. Zookeeper

- Broker 註冊
- Topic 註冊
- Partition 註冊

Kafka v.s. Zookeeper

- Producer Load Balance

Kafka v.s. Zookeeper

- Consumer Group ID
- Consumer LoadBalance
- Consumer offset

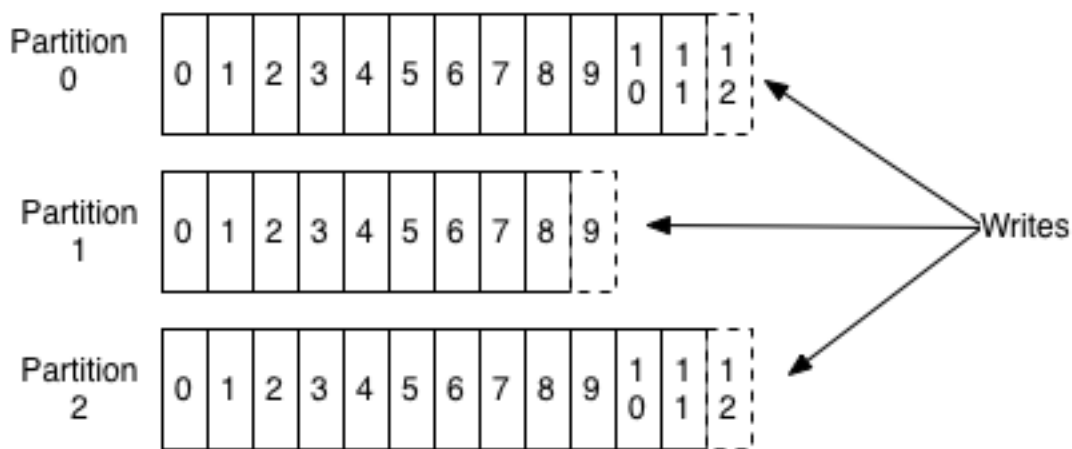
Consumer 讀取資料

- High Level Consumer
使用系統紀錄的offset
- Low Level Consumer
指定offset

Kafka的持久性

- 被讀取過的offset，Message不會立刻刪除，是根據保留時間、Partition大小!!! 藉此規避了數據遺失風險~

Anatomy of a Topic



Old <http://blog.csdn.net/donglei915> New

性能優化

- Message set
為了避免過多零碎的I/O， Message可批量處理
- zero copy
- 數據壓縮

Consumer

- Pull or push?

Kafka 是pull, why?

Kafka的保證

- Producer 發到特定partition的消息，offset會依序
- 消費者除非指定，收到的消息也會相同順序
- 因為有replicate，可以允許 N-1 服務器遺失

Thank you for listening



Kafka Part2就等到Spark Streaming後囉!!!