

Redpanda vs Apache Kafka

v0.2



7Last



Versioni

Ver.	Data	Redattore	Verificatore	Descrizione
0.2	2024-04-22	Matteo Tiozzo	Antonio Benetazzo	Limitazioni di Apache Kafka, Vantaggi di Redpanda
0.1	2024-04-22	Elena Ferro	Antonio Benetazzo	Confronto Redpanda vs Apache Kafka

Indice

1	Introduzione	3
1.1	Apache Kafka	3
1.2	Redpanda	3
2	Limitazioni di Apache Kafka	4
2.1	Architettura	4
2.2	Facilità di utilizzo e rilascio	4
2.3	Dispendioso e limitante	4
2.4	Prestazioni	4
3	Vantaggi di Redpanda	5
3.1	Performance	5
3.2	Costi	5
3.3	Semplicità di configurazione	5
3.4	BYOC (<i>Bring Your Own Cluster</i>)	5
3.5	Compatibilità con API di Kafka	5
3.6	<i>Self-healing</i>	6
4	Benchmark	7
5	Tabella riassuntiva	9
6	Sitografia	11

Indice delle tabelle

1	Riassunto del confronto tra <i>Apache Kafka</i> e <i>Redpanda</i>	10
---	---	----

Indice delle immagini

1	Architettura di Kafka con ZooKeeper	6
2	<i>Risultati del benchmark di latenza.</i>	7
3	<i>Costo relativo di esecuzione di Redpanda vs Kafka.</i>	8



1 Introduzione

Questo documento si pone l'obiettivo di confrontare Redpanda e Apache Kafka. In particolare, verranno analizzate le caratteristiche, i vantaggi e gli svantaggi di entrambe le piattaforme.

1.1 Apache Kafka

Apache Kafka è una piattaforma di *streaming* di dati scritta in Java e Scala. È stato originariamente sviluppato da LinkedIn e successivamente donato alla Apache Software Foundation.

1.2 Redpanda

Redpanda (ex Vectorized) è una piattaforma di *streaming* di dati sviluppata in C++. È un'alternativa ad Apache Kafka, progettata per offrire prestazioni elevate mantenendo la compatibilità con le API e il protocollo di Kafka.



2 Limitazioni di Apache Kafka

2.1 Architettura

Può diventare sempre più complessa da gestire su larga scala, in quanto, a differenza di Redpanda, Apache Kafka **non** è un unico binario.

2.2 Facilità di utilizzo e rilascio

Il progetto originale di Kafka pone un limite alle prestazioni che si possono ottenere dall'hardware moderno. La sua dipendenza dalla cache delle pagine e dalla garbage collection può essere problematica, a causa dei componenti estranei che si trascina dietro, come la JVM e Apache ZooKeeper. Tutto ciò rende l'esecuzione di Kafka su scala più difficile del necessario.

2.3 Dispendioso e limitante

Il progetto originale di Kafka pone un limite alle prestazioni che si possono ottenere dall'hardware moderno. La sua dipendenza dalla cache delle pagine e dalla garbage collection può essere problematica e non parliamo poi dei problemi di distribuzione e gestione, a causa dei componenti estranei che si trascina dietro, come la JVM e Apache ZooKeeper.

2.4 Prestazioni

Non ci sono problemi con la dimensione individuale del messaggio. Tuttavia, i broker e i consumatori iniziano a comprimere questi messaggi man mano che la dimensione aumenta. A causa di ciò, quando vengono decompressi, la memoria del nodo viene utilizzata lentamente.



3 Vantaggi di Redpanda

3.1 Performance

Redpanda è scritto in C++ e utilizza il *framework* Seastar, offrendo un'architettura *thread-per-core* ad alte prestazioni. Ciò permette di ottenere un'elevata *throughput* e latenze costantemente basse, evitando cambi di contesto e blocchi. Inoltre, è progettato per sfruttare l'*hardware* moderno, tra cui unità NVMe, processori *multi-core* e interfacce di rete ad alta velocità.

3.2 Costi

Anche per carichi di lavoro ridotti, l'utilizzo di Kafka può essere fino a 3 volte più costoso rispetto a Redpanda. Per carichi di lavoro più complessi, questa differenza può aumentare fino a 5 volte o più.

3.3 Semplicità di configurazione

Redpanda è un contenuto in un singolo binario. Lo *schema registry*, il *proxy* HTTP e il *message broker* sono tutti integrati. Ciò significa che non ci sono dipendenze da JVM, ZooKeeper o KRaft.

3.4 BYOC (*Bring Your Own Cluster*)

Redpanda offre una terza opzione oltre alla gestione autonoma di un *cluster* di *streaming* dati e all'utilizzo di un servizio *cloud* completamente gestito: *Bring Your Own Cluster* (BYOC). Questa alternativa consente agli utenti finali di implementare una soluzione parzialmente gestita dal fornitore nella propria infrastruttura (come il proprio *data center* o il proprio *VPC cloud*).

3.5 Compatibilità con API di Kafka

Redpanda è progettato per essere compatibile con le API di Kafka, consentendo di utilizzare i *client* Kafka esistenti senza modifiche.



3.6 Self-healing

Redpanda è self-healing e redistribuisce continuamente i dati e la *leadership* tra i nodi per mantenere il *cluster* in uno stato ottimale mentre il *cluster* evolve o quando i nodi falliscono.

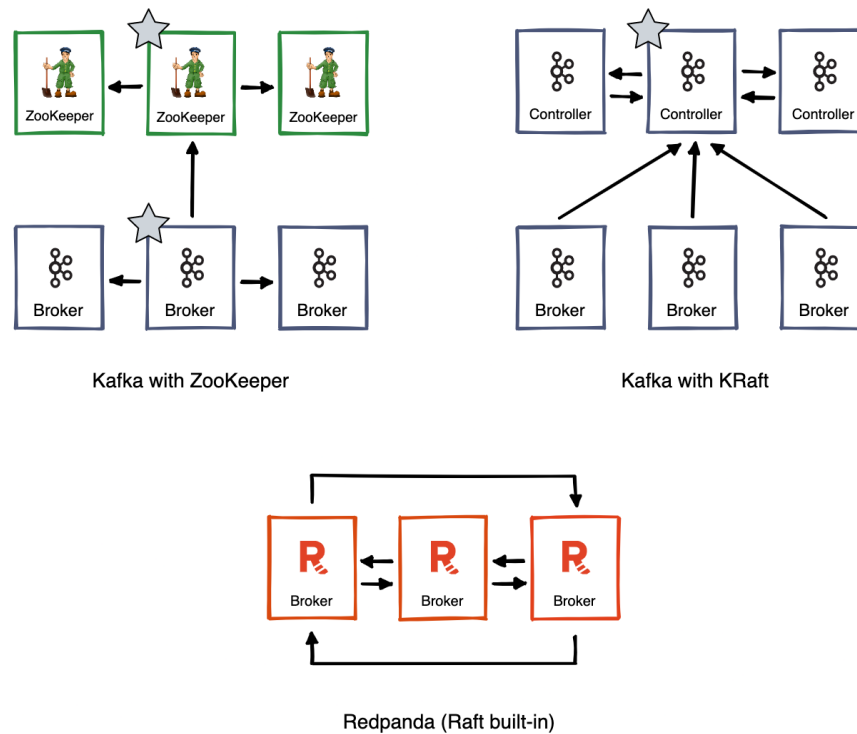


Figure 1: Architettura di Kafka con ZooKeeper



4 Benchmark

Seguono i risultati dei *benchmark* effettuati dal team di sviluppo di Redpanda, confrontati con quelli di Apache Kafka.

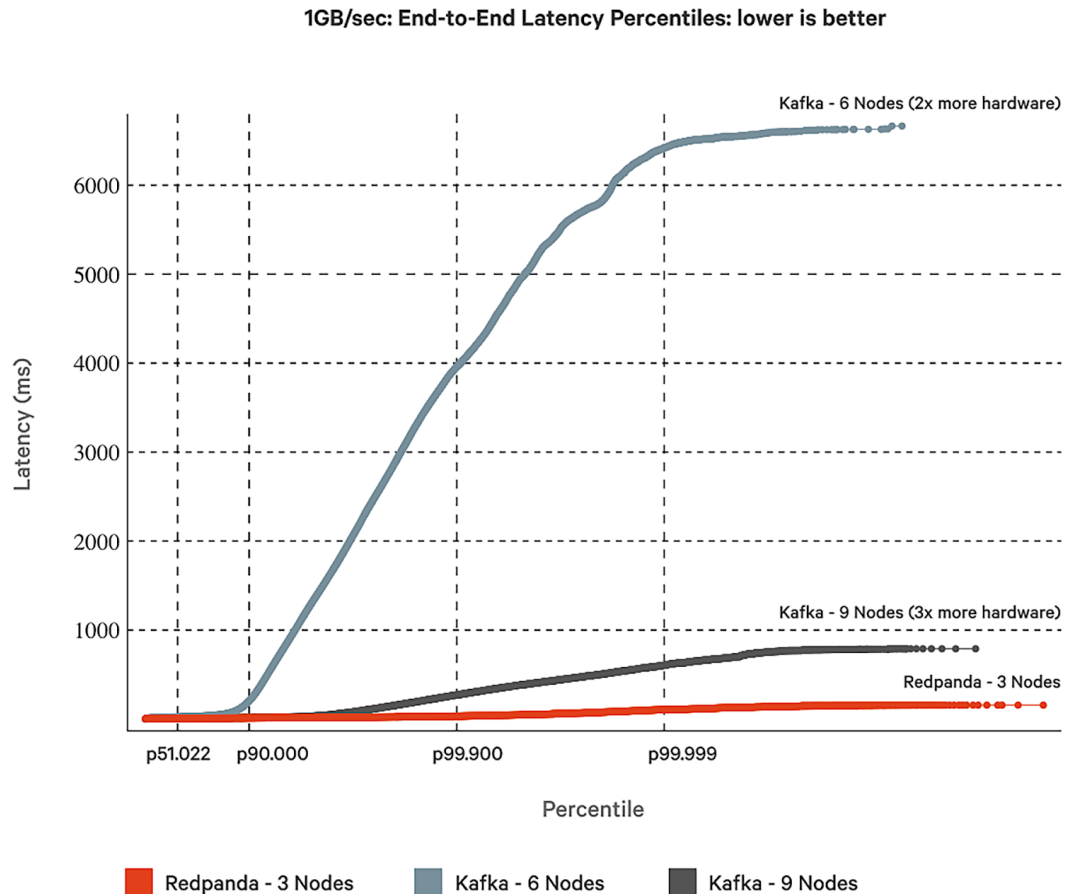


Figure 2: Risultati del benchmark di latenza.



Annual Operating Costs - Redpanda and Apache Kafka

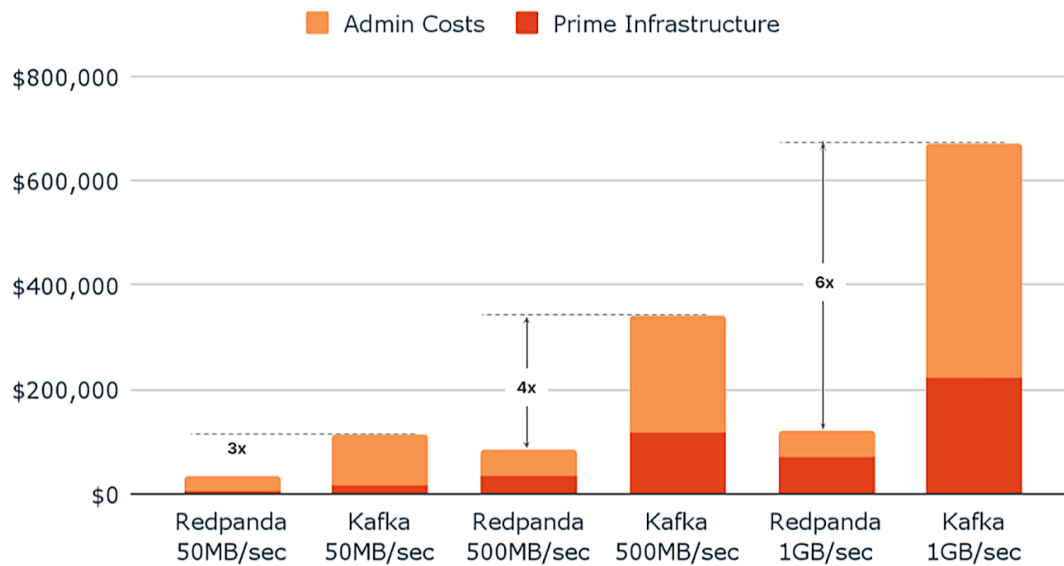


Figure 3: *Costo relativo di esecuzione di Redpanda vs Kafka.*



5 Tabella riassuntiva

Paragone	Apache Kafka	Redpanda
Adozione	Utilizzato da migliaia di compagnie (tra cui LinkedIn, Airbnb, e Netflix)	Non chiaro quante organizzazioni lo usino. Adottato da Cisco e Vodafone.
Community	Migliaia di contributori	<i>Community</i> più piccola ed emergente.
Maturità	Stabile, sviluppato dal 2011	Emergente, lanciato nel 2019.
Documentazione, risorse	Documentazione dettagliata, forum, tutorial, e corsi online	Documentazione dettagliata, ma non altrettante risorse. Tutorial creati dal team di Redpanda.
Client	Ampia varietà di <i>client</i> per i principali linguaggi di programmazione	Lista di <i>client</i> <u>ufficialmente testati</u> , ma secondo la documentazione qualsiasi <i>client</i> Kafka dovrebbe funzionare.
CLIs	Include un set di strumenti per gestire i topic, messaggi, cluster...	Include <code>rpk</code> , un'interfaccia per gestire topic, messaggi, debugging, interazione con Redpanda Cloud.
Monitoraggio	Richiede configurazioni di sistemi di monitoraggio (JMX, Grafana, Prometheus)	Integrato direttamente con Prometheus e Grafana.



Paragone	Apache Kafka	Redpanda
Facilità di utilizzo	Complesso da configurare e gestire	Facile da installare e configurare, indipendente da Zookeeper
Licenza	Open source, Apache 2.0	Edizioni <i>Community</i> e <i>Enterprise</i> , BSL (Business Source License).
<i>Deploy self-hosted</i>	<i>Bare-metal</i> , macchine virtuali, <i>cloud</i> , <u>Docker</u> , Kubernetes	<i>Bare-metal</i> , macchine virtuali, <i>cloud</i> , <u>Docker</u> , Kubernetes
<i>Managed deploy</i>	Numerosi servizi di terze parti, come Confluent Cloud, AWS MSK...	Offre 3 opzioni: <i>cluster</i> dedicati gestiti da Redpanda, BYOC (<i>Bring Your Own Cloud</i>), <i>cluster serverless</i> su architettura gestita da Redpanda.

Tabella 1: Riassunto del confronto tra *Apache Kafka* e *Redpanda*



6 Sitografia

- Confronto tra Redpanda, Kafka e Confluent
- KRaft
- Benchmark
- Opzioni di deployment