

Redpanda vs Apache Kafka

v0.1

7Last

Versioni

Ver.	Data	Redattore	Verificatore	Descrizione
0.2	2024-04-22	Matteo Tiozzo	Antonio Benetazzo	Limitazioni di Kafka, compara- zioni
0.1	2024-04-22	Elena Ferro	Antonio Benetazzo	Vantaggi di Redpanda, Van- taggi di Kafka, Benchmark

Indice

1	Introduzione	3
1.1	Apache Kafka	3
1.2	Redpanda	3
2	Limitazioni di Apache Kafka	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
3.7	Architettura e Sicurezza	6
4	Svantaggi di Redpanda/Vantaggi di Apache Kafka	8
4.1	Maturità	8
4.2	Comunità e supporto	8
4.3	Integrazione con altri servizi	8
5	Benchmark	9
6	Tabella riassuntiva	11
7	Altre tecnologie equivalenti	13
7.1	Amazon MSK	13
7.2	Confluent Kafka	13
8	Sitografia	14

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

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.

3.7 Architettura e Sicurezza

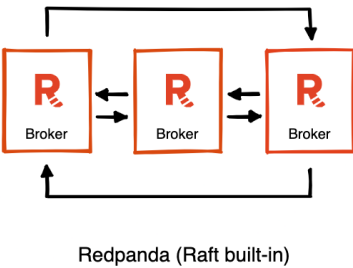
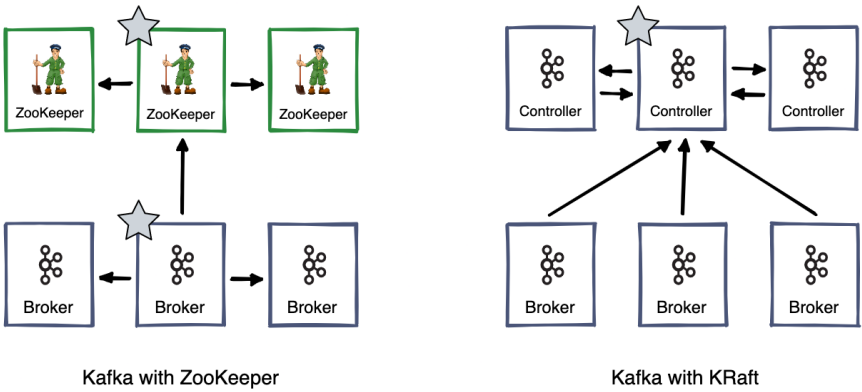
La versione più recente di Kafka ha introdotto KRaft come sostituto di Apache ZooKeeper. KRaft, come suggerisce il nome, sfrutta il protocollo Raft per gestire i metadati del *cluster*. KRaft è stato marcato come pronto per la produzione in Apache Kafka 3.4.0. Tuttavia, alcune limitazioni rimangono, tra cui:

- Nessuna possibilità di aggiornare i *cluster* esistenti da ZooKeeper a KRaft
- Nessun supporto per l'autenticazione SASL/SCRAM
- Modifiche al *tooling* da riga di comando che non funzionano con TLS

Con KRaft, solo i metadati e lo stato del *cluster* sono memorizzati nel *quorum* di KRaft, mentre i *topic* e le partizioni utilizzano ancora il meccanismo ISR.

Per contrasto invece, Redpanda utilizza Raft per la configurazione del *cluster*, lo stato e il mantenimento del consenso attorno alle repliche delle partizioni. Questo utilizzo di Raft per la replicazione consente a Redpanda di offrire garanzie di sicurezza dei dati superiori e di evitare scenari documentati in cui Kafka può perdere lo stato e richiedere un'elezione del *leader* non sicura.

Fonte



4 Svantaggi di Redpanda/Vantaggi di Apache Kafka

4.1 Maturità

Redpanda è stato rilasciato per la prima volta nel 2019, mentre Apache Kafka è stato rilasciato per la prima volta nel 2011. Questo significa che Kafka ha avuto più tempo per svilupparsi, maturare e stabilizzarsi, e ha un livello di maturità e stabilità che Redpanda non ha ancora raggiunto.

Per lo stesso motivo, Kafka ha una maggiore adozione e diffusione.

4.2 Comunità e supporto

Kafka ha una vasta e attiva comunità di sviluppatori, che forniscono supporto, risorse e strumenti per estendere e migliorare il progetto. Kafka ha una documentazione molto completa e ben strutturata, e sono forniti molti tutorial, guide e risorse online per imparare ad utilizzarlo.

Redpanda, al contrario, ha una comunità più piccola e meno attiva, e quindi meno risorse e strumenti disponibili.

4.3 Integrazione con altri servizi

Kafka è supportato da una vasta gamma di strumenti e librerie di terze parti che lo integrano con altri sistemi e servizi (con cui Redpanda dovrebbe essere compatibile, ma non è garantito).

5 Benchmark

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

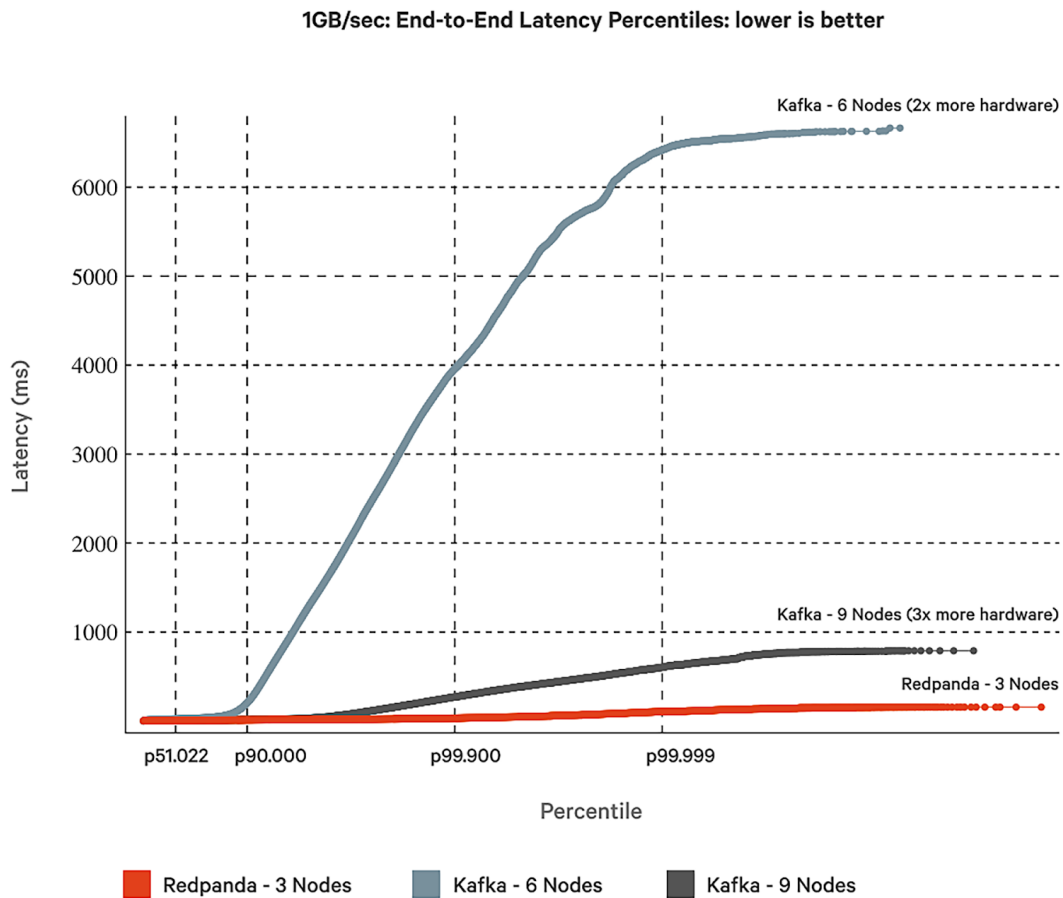


Figure 1: *Risultati del benchmark di latenza.*

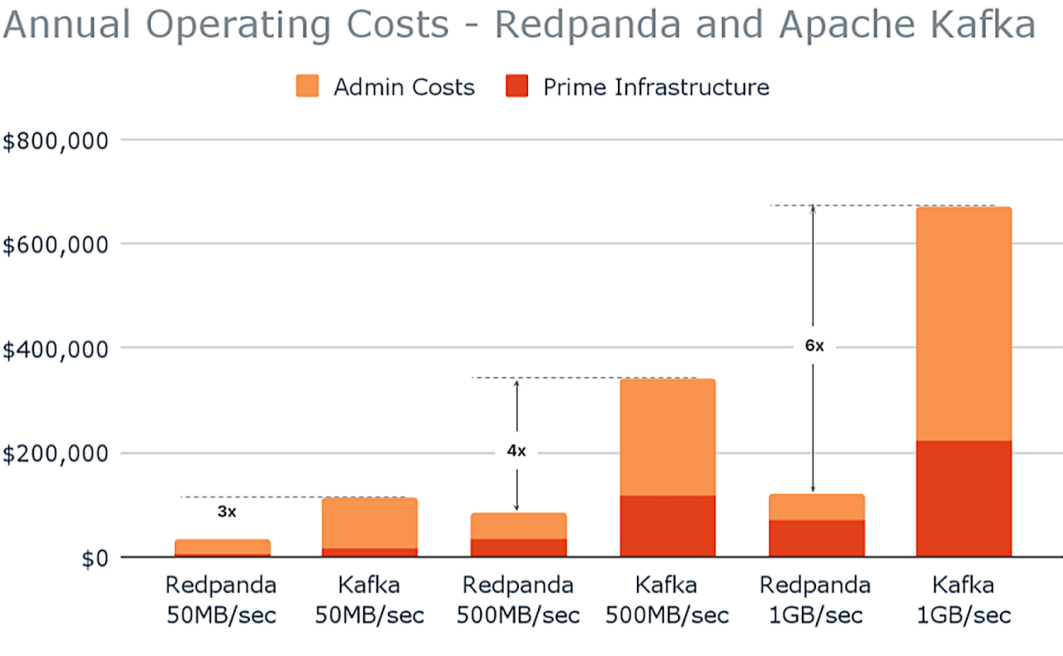


Figure 2: *Costo relativo di esecuzione di Redpanda vs Kafka.*

6 Tabella riassuntiva

Paragone	Apache Kafka	Redpanda
Paragone	Apache Kafka	Redpanda
Stelle su Github	27.5k	8.8k
Fork su Github	13.5k	539
Adozione	Utilizzato da migliaia di compagnie (tra cui LinkedIn, Airbnb, e Netflix)	Non chiaro quante organizzazioni lo usino. Adottato da Cisco e Vodafone.
Comunità	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 <u>client ufficialmente testati</u> , ma secondo la documentazione qualsiasi client 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.
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).
Rilascio <i>self-hosted</i>	<i>Bare-metal</i> , macchine virtuali, <i>cloud</i> , Docker, Kubernetes	<i>Bare-metal</i> , macchine virtuali, <i>cloud</i> , Docker, Kubernetes
Rilascio <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 comparativa tra Apache Kafka e Redpanda

7 Sitografia

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