

Redpanda vs Apache Kafka

v1.0



7Last



Versioni

Ver.	Data	Redattore	Verificatore	Descrizione
1.0	2024-04-23	Elena Ferro	Antonio Benetazzo	Aggiunte conclusioni
0.3	2024-04-23	Elena Ferro	Antonio Benetazzo	Correzioni e aggiunte
0.2	2024-04-22	Matteo Tiozzo	Antonio Benetazzo	Benchmark, Tabella riassuntiva
0.1	2024-04-22	Elena Ferro	Antonio Benetazzo	Vantaggi di Redpanda, Vantaggi di <u>Apache Kafka</u> _G

Indice

1	Introduzione	3
1.1	Apache Kafka	3
1.2	Redpanda	3
2	Vantaggi di Redpanda	4
2.1	Performance	4
2.2	Costi	4
2.3	Semplicità di configurazione	4
2.4	BYOC (<i>Bring Your Own Cluster</i>)	4
2.5	Compatibilità con API di Kafka	4
2.6	<i>Self-healing</i>	4
3	Vantaggi di Apache Kafka	6
3.1	Maturità	6
3.2	Licenza	6
3.3	Comunità e supporto	6
3.4	Integrazione con altri servizi	6
3.5	Scalabilità	6
3.6	Protocollo di replicazione	7
4	Benchmark	8
5	Tabella riassuntiva	10
6	Conclusioni	12

Indice delle tabelle

1	Riassunto del confronto tra <u>Apache Kafka_®</u> e Redpanda	11
---	--	----

Indice delle immagini

1	<u>Architettura di Kafka con ZooKeeper</u>	5
2	<u>Confronto di latenza tra Kafka e Redpanda con e senza <i>fsync</i>.</u>	7
3	<u>Risultati del <i>benchmark</i> di latenza.</u>	8
4	<u>Costo relativo di esecuzione di Redpanda vs Kafka.</u>	9



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 delle due 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 Vantaggi di Redpanda

2.1 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.

2.2 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.

2.3 Semplicità di configurazione

Il binario di Redpanda include, oltre al *message broker*, anche un *proxy* HTTP e uno *schema registry*.

2.4 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*).

2.5 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.

2.6 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.

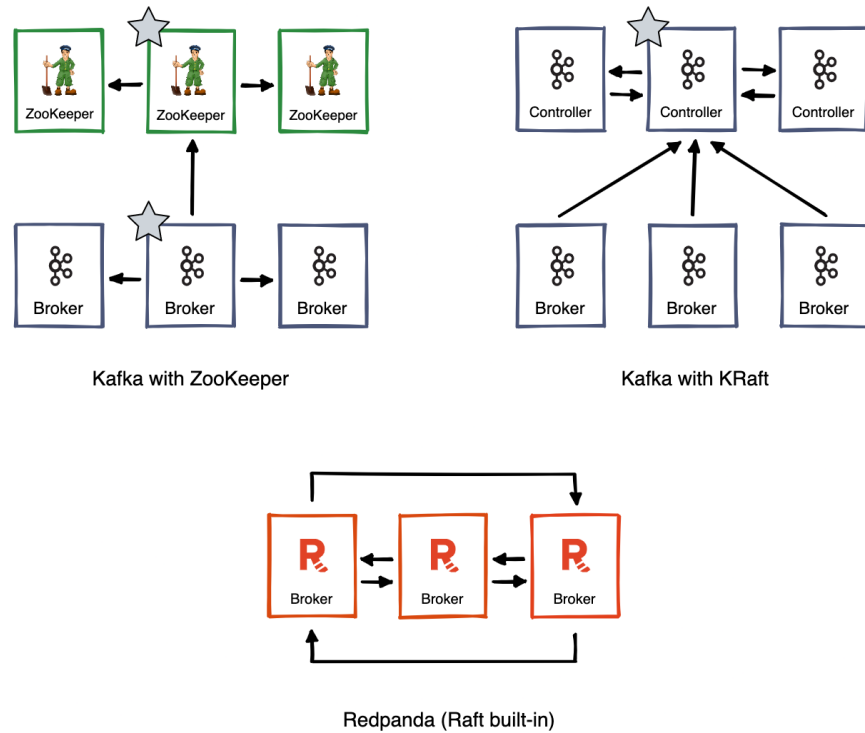


Figure 1: Architettura di Kafka con ZooKeeper



3 Vantaggi di Apache Kafka

3.1 Maturità

Redpanda è stato rilasciato per la prima volta nel 2019, mentre Apache Kafka nel 2011. Quest'ultimo dunque ha potuto svilupparsi e stabilizzarsi nel tempo, raggiungendo un livello di maturità più elevato rispetto a Redpanda.

Ne consegue dunque che Kafka è maggiormente diffuso e utilizzato in ambienti di produzione.

3.2 Licenza

Apache Kafka è rilasciato con la licenza *open source* Apache 2.0, la quale consente di utilizzare, modificare e distribuire il software liberamente. Al contrario, sia l'edizione *community* che quella *enterprise* di Redpanda hanno licenza Business Source License (BSL), che nonostante renda il codice sorgente disponibile, impone delle restrizioni sull'utilizzo e la distribuzione del software.

3.3 Comunità e supporto

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

Redpanda al contrario ha una comunità più piccola e meno attiva, con un numero ridotto di risorse disponibili.

3.4 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 tuttavia Redpanda è compatibile).

3.5 Scalabilità

Redpanda dimostra bassa latenza e alto throughput su *workload* semplici. Tuttavia esso è stato studiato per essere ottimizzato per il *random IO*, e non per il *sequential IO* come Kafka.



Questo significa che in situazioni con un alto numero di produttori, un utilizzo del disco superiore al 30%, l'abilitazione delle chiavi dei messaggi, l'abilitazione di TLS o l'esecuzione per più di 24 ore, le prestazioni di Redpanda possono degradarsi significativamente.

3.6 Protocollo di replicazione

Il protocollo Raft utilizzato da Redpanda per la replicazione e la scrittura su disco è sincrona.

Nei sistemi Linux *fsync* garantisce che i dati siano persistiti in modo sincrono, tuttavia è un'operazione costosa in termini di prestazioni.

Apache Kafka_® può essere configurato per utilizzare anche un protocollo di replicazione asincrono, che non richiede l'utilizzo di *fsync*. Nonostante ciò, Redpanda è in grado di garantire prestazioni migliori rispetto a Kafka, come mostrato nel grafico sottostante.

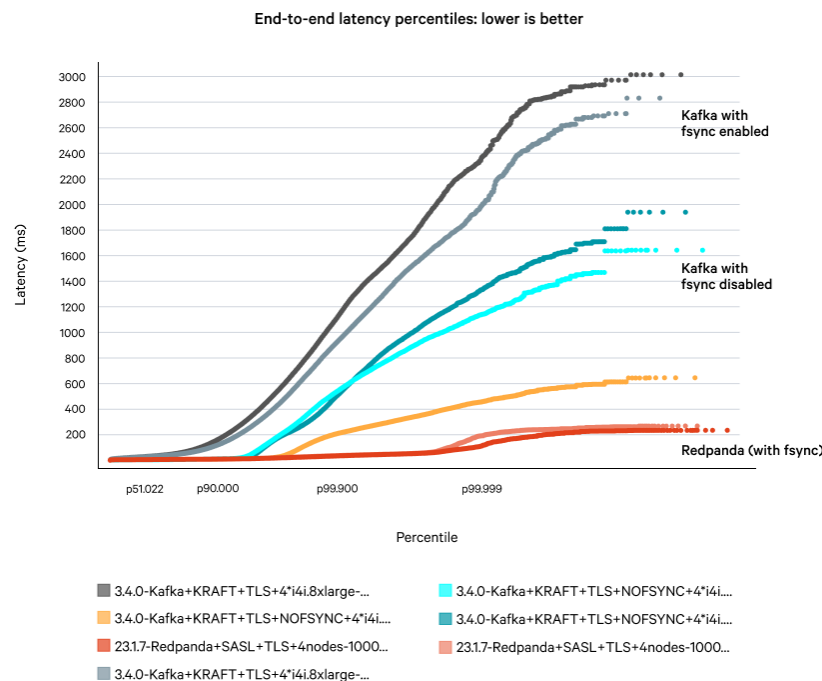


Figure 2: Confronto di latenza tra Kafka e Redpanda con e senza *fsync*.



4 Benchmark

Seguono i risultati dei *benchmark* effettuati dal team di sviluppo di Redpanda, che confrontano le prestazioni dei due strumenti.

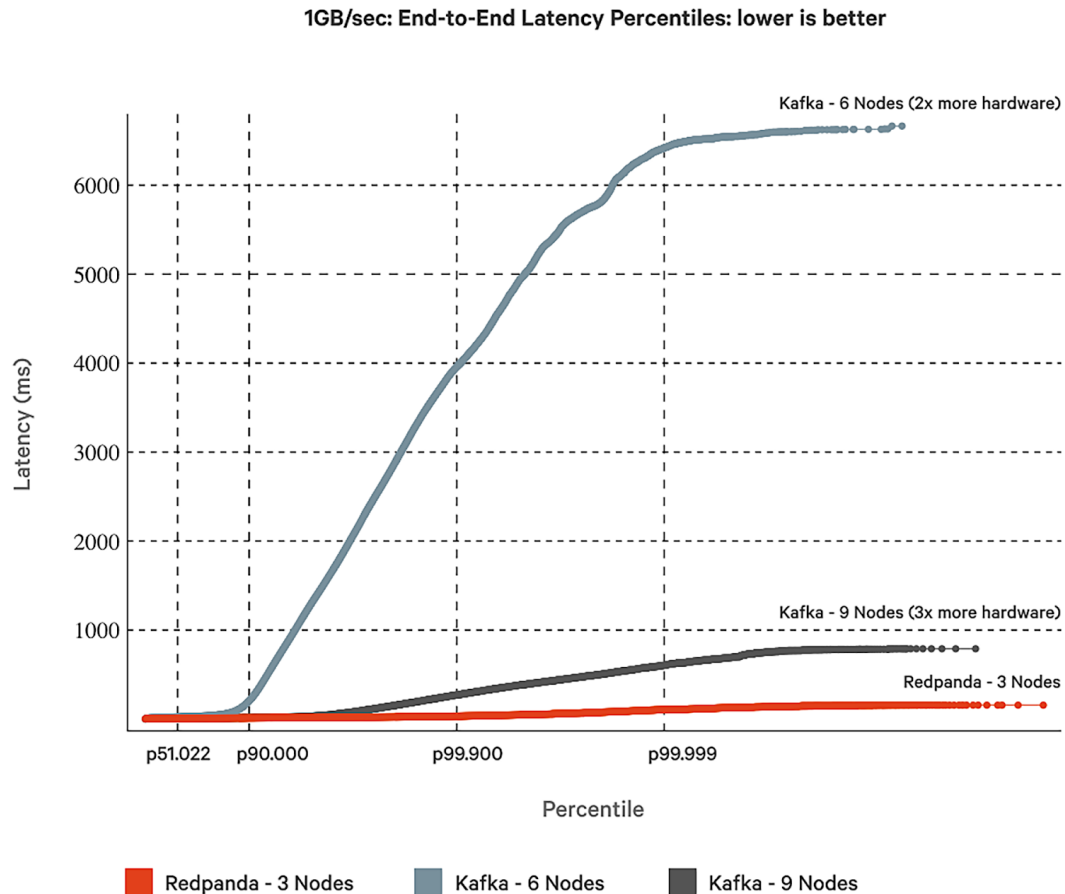


Figure 3: Risultati del *benchmark* di latenza.



Annual Operating Costs - Redpanda and Apache Kafka

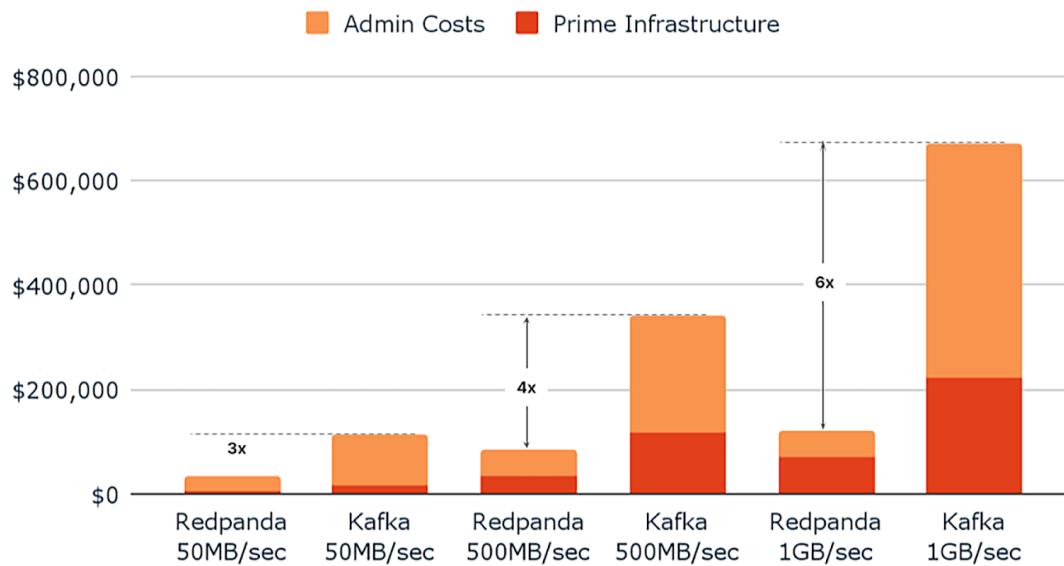


Figure 4: Costo relativo di esecuzione di Redpanda vs Kafka.



5 Tabella riassuntiva

Paragone	<u>Apache Kafka</u> _G	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 client ufficialmente testati, ma <u>qualsiasi client Kafka è compatibile</u> .
CLIs	Include un set di strumenti per gestire i <u>topic</u> _G , messaggi, cluster...	Include <code>rpk</code> , un'interfaccia per gestire <u>topic</u> _G , messaggi, debugging, interazione con Redpanda Cloud.
Monitoraggio	Richiede configurazioni di sistemi di monitoraggio (JMX, <u>Grafana</u> _G , Prometheus)	Integrato direttamente con Prometheus e <u>Grafana</u> _G .



Paragone	<u>Apache Kafka_G</u>	Redpanda
Architettura	Complesso da configurare e gestire su larga scala. Solo a partire dalla versione 3.4.0 è possibile eseguirlo senza ZooKeeper.	Facile da installare e configurare, indipendente da Zookeeper, integrato con una web UI (<u>Redpanda Console</u>).
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_G</u> , Kubernetes	<i>Bare-metal</i> , macchine virtuali, <i>cloud</i> , <u>Docker_G</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.
<i>Schema registry integrato</i>	No	Sì
Protocollo di replicazione	Sincrono o asincrono	Sincrono
Modello di contribuzione	Open source, supporto dalla community e da aziende	Sviluppato solamente dal <i>team</i> di Redpanda

Tabella 1: Riassunto del confronto tra Apache Kafka_G e Redpanda



6 Conclusioni

Kafka e Redpanda sono due strumenti molto simili, ma rispondono ad esigenze differenti. Nel caso si debba gestire un progetto in ambiente di produzione, Kafka è la scelta ottimale, in quanto è più stabile, testato e affidabile. Redpanda invece si presta meglio per progetti più semplici e con carichi di dati minori. Inoltre, risulta maggiormente adatto a utenti più inesperti, in quanto richiede meno configurazioni.

Un altro aspetto da considerare è la licenza: Kafka è *open source*, mentre Redpanda è un prodotto commerciale; nel caso di budget limitato, Kafka risulta dunque più conveniente.

Nelle valutazioni per la scelta dello strumento più adatto, è importante tenere conto che i *benchmark* sono stati eseguiti dai creatori dei *software*, perciò potrebbero essere stati studiati in modo da favorire uno strumento rispetto all'altro.

Ai fini della realizzazione del *Proof of Concept_G* e del *Minimum Viable Product_G* non sono richieste prestazioni elevate in quanto il carico di dati sarà limitato, perciò pensiamo che sia sufficiente utilizzare Redpanda. Essendo il progetto didattico il primo approccio a questo tipo di tecnologia per alcuni membri del gruppo, Redpanda permetterebbe a tutti i componenti di apprendere il funzionamento in modo più semplice e veloce.

Data la compatibilità tra le due tecnologie, in un secondo momento si potrebbe facilmente passare a Kafka, senza dover riscrivere il codice.

Infine, nel caso in cui il progetto dovesse evolvere oltre il *Minimum Viable Product_G*, riterremmo più opportuno passare a Kafka.