ISTITUTO NAZIONALE DI FISICA NUCLEARE

Laboratori Nazionali di Frascati

INFN-2025-03-LNF
26 Marzo 2025

Ceph per servizi general purpose

Ramon Orrù[1], Giovanni Lorenzo Napoleoni[1], Michele Antonio Tota[1]
[1)]*INFN, Laboratori Nazionali di Frascati, SICR, I-00044 Frascati, Italy*

Abstract

Progettazione, implementazione e benchmarking di un cluster Ceph a supporto dei servizi general purpose offerti dal Servizio Infrastrutturale Calcolo e Reti della Divisione Ricerca dei Laboratori Nazionali di Frascati. L'obiettivo è l'implementazione efficace di un sistema di *software defined storage* capace di integrarsi con i sistemi già adottati dal Servizio, e di permettere, inoltre, la successiva integrazione con ulteriori sistemi in fase di realizzazione. Nel permettere la fruizione dello spazio allocabile, dovrà essere garantito il più ampio supporto alle modalità d'accesso locale (*POSIX distributed filesystem*, *block device*, etc.) e ai comuni protocolli per le *storage area network* (iSCSI, NFS). Nell'ottica di favorire l'adozione di servizi *cloud oriented*, sarà auspicabile anche il supporto a protocolli ormai divenuti *standard de facto* per l'interazione con sistemi di *object storage* (Amazon S3). L'affidabilità della soluzione e la semplicità nel suo utilizzo saranno cruciali nel permettere la sovrapposizione e la consecutiva sostituzione di alcuni sistemi di storage tradizionali attualmente utilizzati.

Published by
Laboratori Nazionali di Frascati

1 Overview

Lo storage, inteso come spazio di memorizzazione permanente per i dati gestiti e manipolati dai servizi erogati dal Servizio Infrastrutturale Calcolo e Reti, ricopre un ruolo centrale nel garantire, oltre al corretto funzionamento e all'archiviazione affidabile dei dati, anche l'*interoperabilità* delle piattaforme hardware e software: i sistemi utilizzati sono vari, e utilizzano lo spazio di storage in maniera eterogenea. Alcuni esempi sono: i block device relativi alle *virtual machine* dei sistemi di virtualizzazione, i *file system* ospitanti le risorse web dei servizi locali e di rilievo nazionale e lo spazio dedicato al *backup*. Alcune delle soluzioni tecniche attualmente adottate risultano nella fase finale del loro ciclo di sviluppo/supporto, altre comportano dei costi significativi per acquisto e manutenzione, inoltre l'eterogeneità delle modalità di accesso porta spesso a dover utilizzare e manutenere differenti soluzioni tecniche. In aggiunta, le recenti implementazioni di servizi *cloud PaaS* hanno evidenziato la necessità di utilizzare dei servizi di object storage finora non considerati. Di conseguenza è necessario disporre di uno spazio storage in grado di sostituire,anche gradualmente, gli analoghi servizi esistenti, garantendone la compatibilità e ,allo stesso tempo, supportare l'operatività dei nuovi servizi in divenire attraverso la disponibilità di protocolli addizionali. Il tutto, possibilmente, eliminando o riducendo i costi di licenza associati.

2 Requisiti

I requisiti sono prevalentemente non funzionali e sono espressi come di seguito:

- **interoperabilità**: il sistema deve supportare diversi protocolli/modalità, almeno: file system distribuito (*POSIX-compliant*), block device, object store complatibile con Amazon S3, iSCSI, NFS.
- **affidabilità**: i dati memorizzati devono essere opportunamente salvaguardati impiegando opportune tecniche di ridondanza.
- **fault tolerance**: il sistema deve essere realizzato in modo da garantire alti livelli di disponibilià dei servizi ed eliminare i *single point of failure* dalla prorpria architettura di definizione.
- **scalabilità**: deve essere possibile estendere lo spazio disponbile (o eventualmente ridurlo) in modo quanto più trasparente e semplice possibile.
- **openness**: i protocolli implementati devono rispettare il più possibile gli standard (aperti) di riferimento, e le licenze d'uso del software devo essere "libere" (es. GPL, BSD-like, Creative Commons) anche al fine di ridurre costi e fenomeni di *vendor lock-in*; anche la possibilità di utilizzare "*commodity hardware*", ovvero hardware non specifico, permette di ridurre costi e vincoli.

- **performance adeguate**: lo storage deve essere in grado di supportare workload generici con buone prestazioni, evitando il più possibile colli di bottiglia nella relativa architettura. Sono espressamente escluse le applicazioni per supporto al calcolo parallelo e HPC.
- **autenticazione/autorizzazione**: deve essere disponbile un sistema affidabile di autenticazione e autorizzazione integrato.
- **manutenibilità**: il tool utilizzato deve rendere la propria manutenzione semplice e sicura (es. nessun *down-time*); le funzionalità di monitoraggio delle prestazioni e del corretto funzionamento delle componenti devono essere disponibili. Incide sull'aspetto di manutenibilità anche l'effort necessario a manutenere in maniera adeguata un cluster di medie dimensioni, riducono questa necessità la disponibilità di procedure automatiche, la linearità delle procedure di aggiornamento e ovviamente la presenza di addetti con adeguata expertise.
- **integrabilità**: la soluzione proposta deve potersi integrare con i sistemi per poter affiancare e poi eventualmente sostituire i prodotti storage attuali.

2.1 Integrazione

Per definire meglio i requisiti di integrabilità, è necessario stabilire quali funzionalità devono essere garantite rispetto ai sistemi esitenti. In particolare:

- **Virtual disks**. Deve essere garantito il supporto ai block device virtuali per gli hypervisor in uso: VMware vSphere, oVirt, Proxmox VE.
- **POSIX distributed filesystem**. Il filesystem deve necessariamente permette l'accesso in maniera concorrente da utenti organizzati secondo apposite access policy, previa autenticazione.
- **Piattaforme PaaS**. Deve essere disponibile un servizio compatibile S3 di supporto alle piattaforme PaaS (OKD, OCP). Devono anche essere disponibili, per la medesima PaaS, dei driver CSI (*container storage interface*) in grado di fornire spazio storage nella forma di *persistent volume* allocati dinamicamente.
- **Backup**. Diversi sistemi utilizzano dello spazio NAS (*network attached storage*) per ospitare dati di backup a medio termine, prevalentemente acceduto attraverso NFS.

3 Valutazione soluzioni tecniche

Per individuare la soluzione tecnica che possa rispondere ai requisiti indicati nella sezione 2, è stata effettuata una breve indagine comparativa per valutare diverse possibilità di

implementazione. Sono quindi stati individuati un insieme di prodotti per includerli in una scheda di valutazione rispetto ai requisiti definiti; il criterio di inclusione considerato inizialmente è l'utilizzo di licenze opensource. Di seguito, in tabella 1, un riepilogo delle valutazioni effettuate.

Tabella 1: Valutazione soddisfacibilità requisiti.

	GlusterFS	**HDFS**	**Lustre**	**MinIO**	**Ceph**
interoperabilità	1/3	1/3	1/3	1/3	Sì
affidabilità	Sì	Sì	Sì	Sì	Sì
fault tolerance	Sì	Sì	Sì	Sì	Sì
scalabilità	Sì	Sì	Sì	Sì	Sì
performance	Sì	Sì	Sì	Sì	Sì
openness	Sì	Sì	Sì	Sì	Sì
authn & authz	Parzialmente	Sì	Sì	Sì	Sì
manutenibilità	Parzialmente	Sì	No	Sì	Sì
integrabilità	Parzialmente	Parzialmente	Parzialmente	Parzialmente	Sì

Alla luce dei risultati della valutazione, si è ritenuto opportuno scegliere Ceph per l'implementazione.

4 Ceph

Il progetto Ceph[1] ha lo scopo di rendere possible l'implementazione di sistemi di software defined storage attraverso una serie di componenti raccolte in un progetto opensource. Nel caso di infrastrutture *on-premise* (su dispositivi ad accesso privato), permette di ottenere un livello di servizio paragonabile a sistemi di storage di livello enterprise senza imposizione di vincoli sull'hardware operante o dovuti a licenze proprietarie. Lo stesso storage può offrire tre tipologie di servizi: object store, block access e filesystem, ciascuna delle quali prevede diversi protocolli di accesso, come mostrato in fig.1. I componenti logici che implementano i servizi sono , rispettivamente, Object Gateway (ex RADOSgw), RBD (RADOS block device) e CephFS.

Queste componenti logiche trovano la loro controparte fisica nei processi che compongono effettivamente il cluster, come indicato in fig.2:

- **OSD (object storage device)** è il demone che si occupa di gestire le risorse di archiviazione di massa, quindi hard disk e solid state drive, organizzando i dati che devono essere poi letti o scritti attraverso le opportune richieste al processo OSD.
- **MON** è il servizio che coordina il funzionamento dell'intero cluster: gestisce la mappa dei dati, la comunicazione iniziale con i client, l'autenticazione e l'attività dei processi nel cluster.

[1] `https://ceph.io`

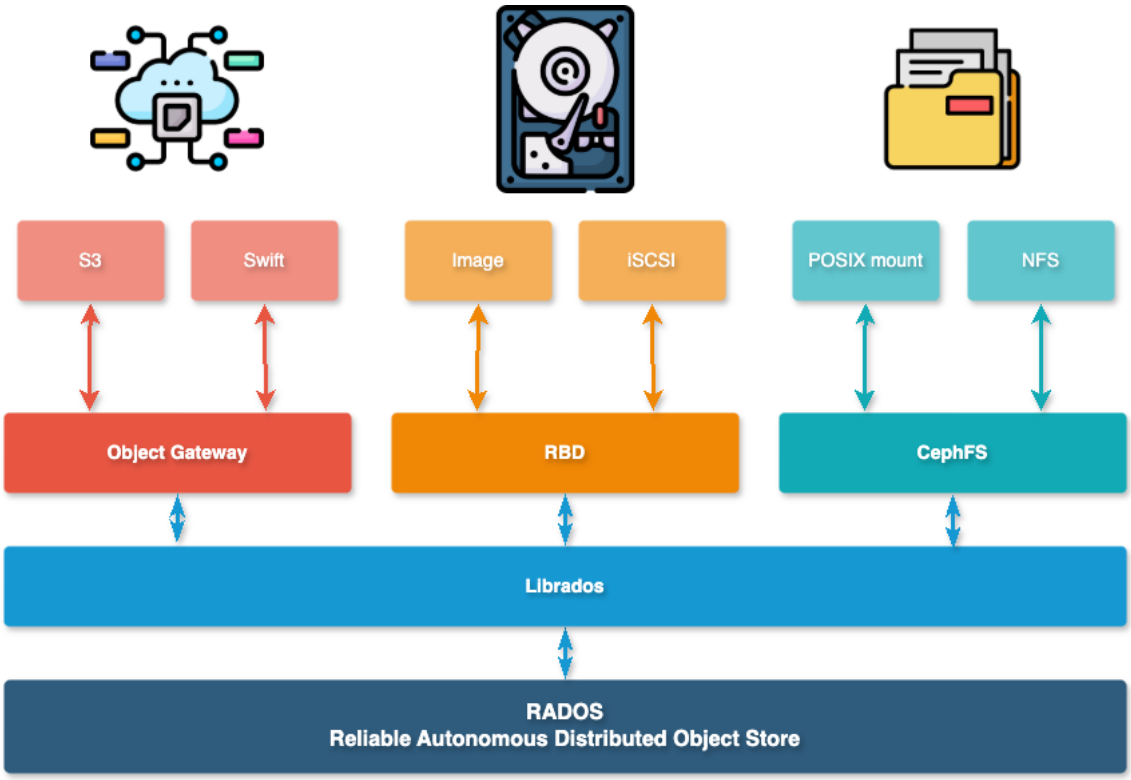

Figura 1: Ceph - architettura logica.

- **MGR** implementa il sistema di monitoring e la dashboard di gestione del cluster.
- **MDS** gestisce e manipola i metadati dei volumi CephFS per garantire la semantica di accesso POSIX ai file, mantiene le informazioni sul mapping di file e firectory rispetto agli oggetti RADOS che ne ospitano effettivamente il contenuto.
- **Object Gateway - RADOSgw** implementa il protocollo Amazon S3 per accedere agli oggetti RADOS.
- **iSCSIgw** implementa il protocollo iSCSI per accedere alle immagini RBD.

Si noti che il servizio RBD, indicato tra le componenti dell'architettura logica, non è implementato da uno specifico processo, poiche è compito del relativo modulo del kernel Linux presente sul client interfacciarsi direttamente con i processi OSD e MON attraverso il protocollo RADOS.

5 Conformità ai requisti

Considerando i requisiti di interoperabilità, Ceph offre gli strumenti per integrare lo storage verso i diversi hypervisor in uso, offrendo un endpoint iSCSI per VMware vSphere e oVirt, mentre per Proxmox VE è possibile utilizzare direttamente le immagini in maniera nativa (RBD). Per quanto riguarda invece il filesystem distribuito, questo può essere direttamewnte montato su host Linux utilizzando il client nativo CephFS disponibile per

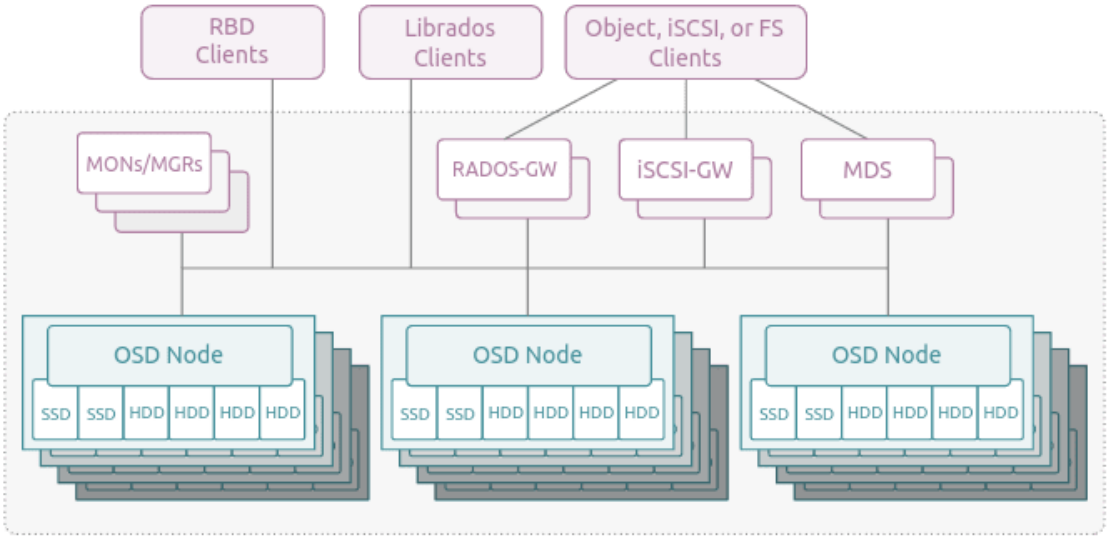

Figura 2: Ceph - cluster daemon.

le comuni distribuzioni; ove non fosse possibile, è prevista un'alternativa tramite l'export secondo protocollo il NFSv4, come per gli attuali NAS. Sulle PaaS sarà disponibile il servizio di object storage S3, e sia RBD che CephFS dispongono di un apposito driver CSI per l'allocazione dei persistent volume.

Ceph implementa 2 diversi schemi di ridondanza: uno basato su fattore di replica e uno basato su techiche di *erasure coding*. Sotto le opportune condizioni, entrambe le soluzioni forniscono un elevato grado di protezione dalla perdita di dati. Mentre la prima soluzione permette un più veloce rispristino dei dati in caso di guasti a discapito dello spazio effettivamente utilizzabile, la seconda permette una maggior densità in termini di capacità effettiva, richiedendo però maggiori risorse in fase di recovery.

Tutti i processi software coinvolti sono orchestrati da un apposito componente di Ceph e vengono distribuiti adeguatamente sui vari host. Per evitare casi di indisponibilità del servizio, questi sono replicati e gestiti secondo delle politiche di load balancing active/active o active/passive. L'effettivo posizionamento del dato è anch'esso gestito in autonomia da Ceph, che viene istruito in fase di configurazione sulla topologia dell'hardware a disposizione, in modo che le repliche del dato possano essere distribuite adeguatamente per ridurre la probabilità di perdita dei dati o la loro indisponibilità. Questa organizzazione dei dati è possibile attraverso l'utilizzo di una struttura dati chiamata CRUSH map, che ha proprio il compito di definire quale sia la gerarchia fisica del cluster e distribuire di conseguenza i dati. In fig.3 [2] è mostrato uno schema di come Ceph organizzi i dati al fine di ottenere un elevato grado di affidabilità e fault tolerance. Tutti i protocolli utilizzati

[2] `https://documentation.suse.com/ses/7.1/html/ses-all/images/device_classes.png`

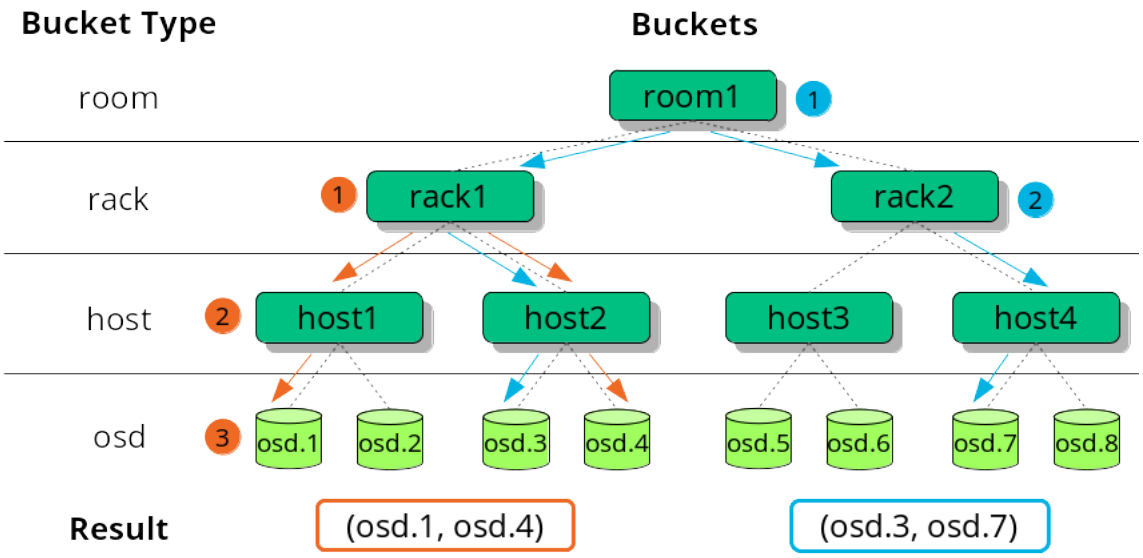

Figura 3: Ceph - CRUSH map.

sono standard aperti e il software che compone Ceph è rilasciato sotto licenza opensource LGPL2.1 o LGPL3.0, la relativa documentazione è rilasciata sotto licenza Creative Commons.

Sul lato delle performance, le prestazioni delle implementazioni realizzate seguendo le best practices indicate nella documentazione sono più che soddisfacenti per la maggior parte dei workload general-purpose. Chiaramente il livello prestazionale è stabilito dall'hardware sottostante e dalle configurazioni apportate. Si noti che Ceph è uno storage distriubuito, e come tale soffre anche di inevitabili problemi di latenza, e non può essere paragonato a sistemi di storage parallelo che eccellono prestazionalmente. La motivazione alla base di Ceph è invece quella di avere uno storage flessibile, scalabile e versatile. Nella sezione 7 verranno mostrati i risultati di alcune sessioni di benchamrk in modo da esprimere in maniera quantitativa le prestazioni offerte.

Ceph include un sistema di autenticazione integrato chiamato cephx, che permette l'attribuzione granulare di permessi attraverso la definizione di utenti ai quali vengono assegnate delle capabilities. Queste sono un'insieme di operazioni che è possibile svolgere sui vari componenti del sistema; ad esempio, è possibile che l'utente "userA" abbia i permessi per creare delle immagini RBD ma possa solo accedere in lettura ad uno specifico filesystem "fsX", mentre l'utente "userB" potrebbe avere pieno accesso a "fsX" ma nessuna capability che gli permetta di utilizzare immagini di tipo block device. Ceph espone inoltre un insieme di API per l'interazione con il sistema, e anch'esse supportano l'autenticazione.

Data la ridondanza introdotta anche per quanto riguarda i processi di sistema del cluster, è possibile effettuare le operazioni di manutenzione in maniera sicura, riducendo o eliminando i tempi di indisponibilità del servizio. Un apposito componente orchestra i servizi

come da configurazione, quindi, quando uno di essi diventa indisponibile, il servizio può essere ripristinato su un altro nodo oppure un processo in stand-by può diventare attivo per gestire le richieste. Inoltre è presente un sistema completo di monitoring integrato che colleziona le principali metriche dei componenti e gestisce i relativi allarmi in caso di malfunzionamento o sovraccarico. Le operazioni di manutenzione ordinaria possono essere eseguite sia da apposita CLI che da dashboard web, la quale include le informazioni e i grafici del sistema di monitoring già citato.

Sono quindi presenti i presupposti per la sostituzione di alcuni dei servizi di storage attuali, anche se questo richiederà un periodo di studio ulteriore per determinarne le corrette configurazioni e scongiurare eventuali complicazioni inattese.

6 Implementazione

Per l'implementazione sono stati utilizzati 6 host, separati in 2 insiemi:

- **Controlplane hosts**. Ospitano i processi MON, MGR, MDS, Object Gateway, NFSGW, iSCSIGW.
- **OSD hosts**. ospitano i supporti di memorizzazione di massa e i processi OSD che li controllano.

6.1 Risorse hardware

In tabella 2 e in tabella 3 sono indicate le caratteristiche tecniche delle due tipologie di host usate per il deploy del cluster.

Tabella 2: Scheda tecnica controlplane host.

Model	**DELL PowerEdge R450**
CPU	2x CPU Intel Xeon 4310
RAM size	128 GB
Root disk	2x 480GB SATA Mix Use
Ethernet adapter	2x 1GBase-T Broadcom BCM5720
FC adapter	2x 10G SFP+

Tabella 3: Scheda tecnica OSD host.

Model	**Supermicro SSG-640P-E1CR24H**
CPU	2x CPU Intel Xeon 4314
RAM size	256 GB
Root disk	2x Micron 5300 Pro SSD 240GB
SSD	8x D3-S4510 1.92TB SATA Mix Use
HDD	16x MG07SCA12TE SAS 12TB 7.2Krpm
Ethernet adapter	2x 10GBase-T Intel® X550
FC adapter	4x 25G SFP28

Per quanto riguarda gli host della controlplane, le risorse assegnate sono più che sufficienti a supportare l'esecuzione dei processi correlati alle dimensioni della parte OSD, ma sono stati volutamente sovradimensionati per garantire il funzionamento anche dopo eventuali successive espansioni.

Avendo a disposizione un'infrastruttura adeguata, il comparto di rete è stato dotato di interconnessione in fibra ottica supportando quindi velocità ampiamente al di sopra del collo di bottiglia rappresentato dalle prestazioni attese del singolo nodo OSD, anche tenendo conto di un eventuale intenso traffico inter-cluster. I dispositivi FC sono stati scelti in numero sufficiente per garantire un'adeguata tolleranza ai guasti, come indicato nella sezione successiva. Non sono stati invece utilizzati i dispositivi basati su cavo metallico (presenti di serie sui modelli scelti): il loro impiego può essere sfruttato per realizzare una rete di management che consentirebbe di raggiungere gli host anche in caso di completa indisponibilità della rete pubblica.

Per il costo troppo elevato non sono state considerate configurazioni corredate da device NVMe (nonvolatile memory express) che potenzialmente avrebbero consentito prestazioni più elevate.

6.2 Interconnessione di rete

Lo schema di cablaggio è indicato nella fig.4, dove si evidenziano le interconnessioni di rete.

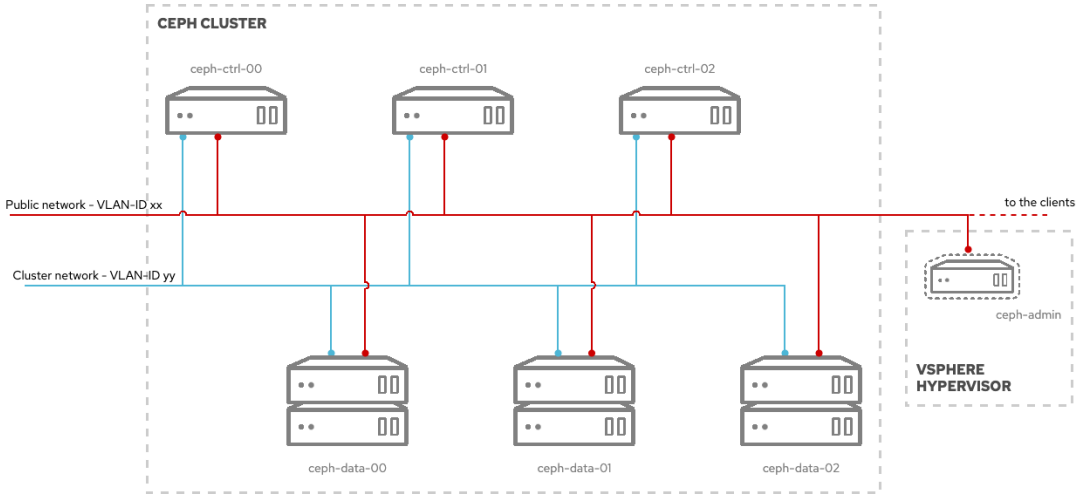

Figura 4: Ceph - schema di deploy.

Nel caso specifico si sono utilizzate 2 network distinte:

- **Public network**: ospita il traffico da e per i client, tutti i processi e i nodi devono essere raggiungibili dai client attraverso questa network.
- **Cluster network**: supporta il traffico inter-cluster, ovvero tutta la comunicazione che intercorre tra i vari processi interni al cluster, come redistribuzione dei dati, replicazione, e in generale tutto ciò che non riguarda l'interazione con i client.

La separazione del traffico tra le due network è consigliata[3] per evitare un possibile aumento di traffico generato dalle eventuali operazioni di ripristino possa pregiudicare in maniera seria le prestazioni del servizio erogato ai client sulla rete pubblica.
Nella figura 5, è mostrato lo schema di interconnessione degli host che compongono la controlplane.

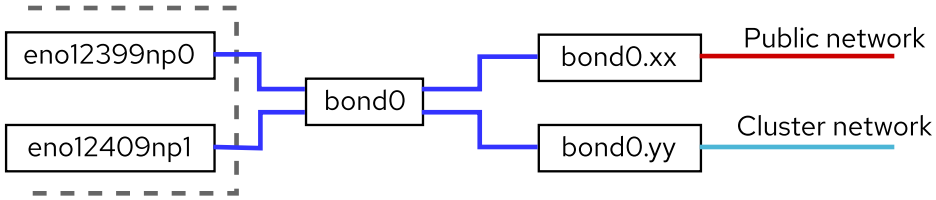

Figura 5: Interconnessione controlplane host.

Avendo a disposizione una singola scheda con due separate NIC, si è deciso di utilizzare le 2 interfacce in bonding per poter avere tolleranza al guasto della singola porta e usufruire di una connessione con banda passante aggregata durante la normale operatività.
Nella fig.6, è mostrato lo schema di interconnessione degli host ospitanti gli OSD.

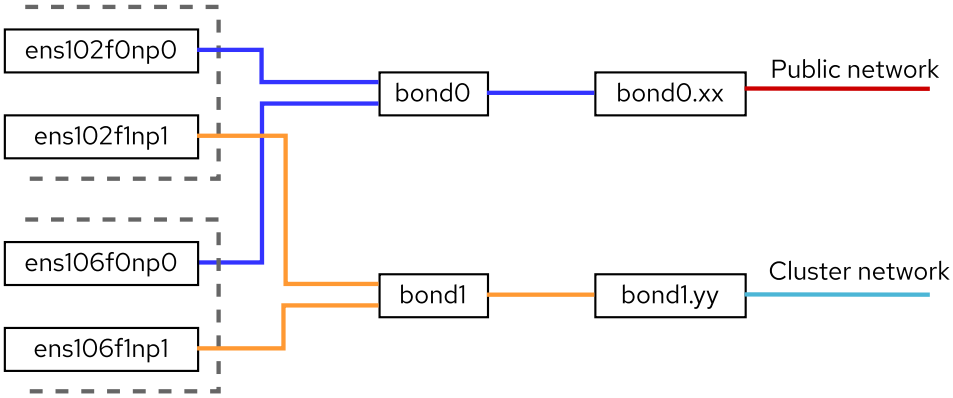

Figura 6: Interconnessione OSD host.

In questo caso, avendo a disposizione 2 schede da 2 porte ognuna, si è optato per una configurazione in bonding ”distribuendo” le porte slave per ogni bonding su entrambe le schede. In questo modo si raggiunge un aumento di banda passante aggregata e tolleranza ai guasti della singola port o della singola scheda di rete, anche se con possibile degrado delle prestazioni.

[3] `https://docs.ceph.com/en/latest/rados/configuration/network-config-ref/`

6.3 Configurazione OSD

Nell'architettura di Ceph, la gestione a basso livello dell'hardware per quanto riguarda i supporti di memorizzazione di massa (hard disk, solid state drive, dischi NVMe) è demandata ai processi OSD (object store daemon). Questi processi utilizzano l'hardware a disposizione organizzando lo spazio disponibile suddividendolo in 3 parti distinte di un medesimo *data store* chiamato Bluestore:

- **Data Block (Data)** In questi spazio Ceph memorizza gli oggetti direttamente come blocchi su un device senza l'utilizzo di filesystem, al fine di eliminare sovrastrutture e ottenere maggiori performance.
- **Block Database (DB)** Garantisce la consistenza dei dati immagazzinando, per ogni oggetto, indirizzi dei blocchi nel data block, placement group e metadati. Queste informazioni sono contenute e organizzate in un database chiave/valore. Può risiedere sullo stesso device del data block, oppure su un device più performante, come SSD e dispositivi NVMe, al fine di aumentare le prestazioni.
- **Write-ahead Log (WAL)** Gestisce le operazioni di accesso ai dati in maniera atomica, mantenendo un journal delle operazioni. Come per il block database, può risiedere in un device diverso da quello che ospita il data block.

Per un incremento prestazionale dell'intero data store, è consigliato[4] destinare *WAL* o *DB* su device meglio performanti, e lasciare la parte *Data* su device con minor rapporto costo/capacità. Secondo le best practices [5], se è dispoinibile uno spazio su device non rotazionali (SSD, NVMe) dell'ordine di almeno il 1-4% rispetto alle dimensioni del data block, è suggerito utilizzarlo per ospitare il DB (senza limite superiore alle dimensioni ammesse). Se lo spazio "veloce" disponibile è inferiore, è consigliato destinarvi il WAL. Si noti che nel caso il DB venga locato su un device differente dal data block, il WAL viene implicitamente spostato anch'esso sullo stesso device.

Nel nostro specifico caso, l'hardware mette a disposizione 16 dischi rotativi da 12 TB e 8 SSD da 1.92 TB su ognuno degli *OSD host*, pertanto lo schema di configurazione per gli OSD è quello rappresentato in figura 7, dove i 16 OSD utilizzano ognuno un device rotazionale e la metà dello spazio disponbile su ogni SSD.

Si noti che la scelta di condividere un dispositivo performante per ospitare DB o WAL rappresenta un compromesso tra il possibile incremento di prestazioni ad un costo contenuto e la possibile riduzione del grado di tolleranza ai guasti. Questo perchè condividere lo spazio sul medesimo disco prestazionale riduce chiaramente la necessità di disporre di un numero più alto di dispositivi, ma fa anche si che il fault di un singolo

[4] `https://docs.ceph.com/en/latest/rados/configuration/bluestore-config-ref/#devices`

[5] `https://docs.ceph.com/en/latest/rados/configuration/bluestore-config-ref/#sizing`

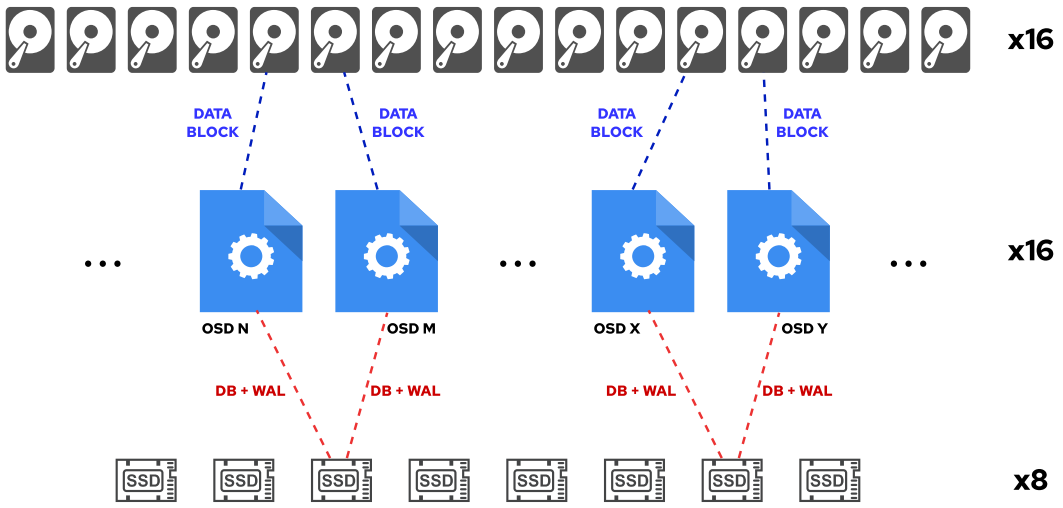

Figura 7: Configurazione OSD host.

disco performante pregiudichi completamente la dispoinbilità di tutti gli OSD coinvolti. Inoltre, in caso di fault e perdita dei dati di un DB o WAL localizzati su un device separato, non sarà possibile ripristinare il funzionamento dell'OSD sostituendo il device guasto, ma sarà invece necessario riconfigurare completamente un nuovo OSD anche sul data block device.

6.3.1 Replica

Lo spazio disponibile in un cluster Ceph può essere suddiviso in diverse porzioni chiamate *pool*. Ogni pool può essere utilizzato per una specifica applicazione, e può essere configurato opportunamente nelle modalità con le quali i dati vengono organizzati.

Come menzionato, Ceph è in grado di scalare la sua implementazione sia in aumento che in diminuzione delle risorse (in particolare rispetto al numero di nodi OSD), oppure quando alcune di esse subiscono un guasto e quindi diventano indisponibili. Questo implica che gli oggetti contenuti nelle risorse in fault vengano spostati, in modo da mantenere un certo livello di bilanciamento all'interno della topologia fisica del cluster. La gestione del singolo oggetto sarebbe particolarmente ardua da assicurare in ambienti dove questi raggiungono numerosità elevata, quindi i vari oggetti all'interno del singolo pool vengono suddivisi in *placement group*, di seguito PG. I placement group sono quindi considerati l'unità atomica nel collocamento dei dati, e il ribilanciamento viene organizzato sulla base della distribuzione dei PG. Ogni pool definisce anche le impostazioni sulle tecniche e algoritmi di ridondanza, come replica (e relativo fattore), EC (e relativi parametri), etc. In figura 8 [6] è mostrato come i vari oggetti siano organizzati in PG nel caso di un pool replicato con fattore 2:

[6] `https://access.redhat.com/webassets/avalon/d/Red_Hat_Ceph_Storage-6-Architecture_Guide-en-US/images/08af4a1fab18995fda3aad1c3ede873e/arc-04.png`

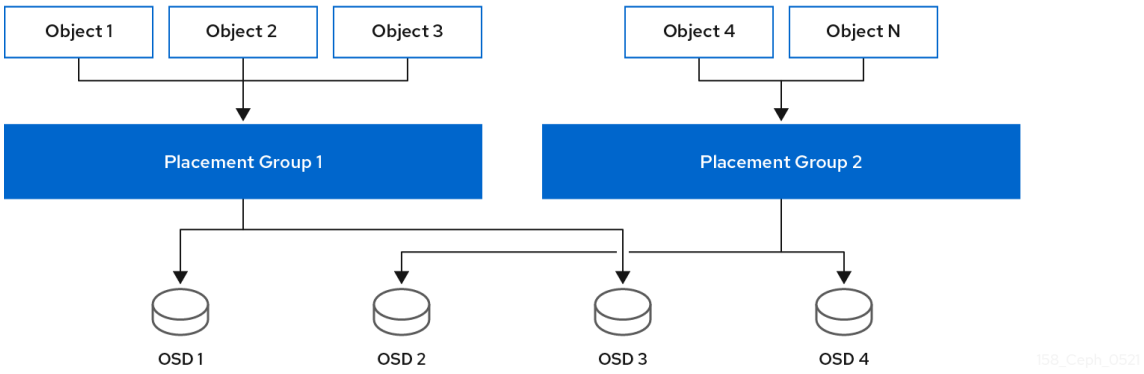

Figura 8: Placement group.

Quindi è facile intuire che il numero di PG nei quali è suddiviso un pool possa influire sulle prestazioni dello stesso, poichè pochi PG troppo popolosi o di dimensioni ridotte ma in numero elevato rispetto al numero di oggetti possono portare a delle inefficienze anche gravi. Le indicazioni fornite dalla documentazione [7] indicano come desiderata un numero di PG pari a 150 per singolo OSD, da qui, conoscendo numero di OSD coinvolti e stimando numero e dimensione media degli oggetti, è possibile ottenere in numero di PG per pool. Nelle recenti versioni di Ceph è stato incluso un'apposita funzionalità di *PG autoscaling* che si occupa di riconfigurare dinamicamente il numero di PG per il singolo pool in base alle variazioni del numero di oggetti contenuti, perciò è consigliato impostare manualmente il numero di PG solo se è necessario popolare in maniera massiva un pool conoscendo in maniera dettagliata l'intera mole di dati.

Lo schema di ridondanza dei dati scelto nella maggior parte dei pool definiti sul cluster in oggetto è quello basato su replica, con un fattore 3. Questo implica che lo stesso dato sia presente 3 volte all'interno del cluster, e dato che la CRUSH map considera come fault domain il singolo host, il dato sarà presente su 3 host diversi. Durante la scrittura, il dato viene quindi effettivamente memorizzato su 3 OSD (considerati primario, secondario e terziario per lo specifico PG) prima che al client venga notificata l'avvenuta scrittura, come schematizzato in figura 9 [8].

[7] https://docs.ceph.com/en/reef/rados/operations/placement-groups/

[8] https://access.redhat.com/webassets/avalon/d/Red_Hat_Ceph_Storage-6-Architecture_Guide-en-US/images/ef4e58d7ba3d62ac6add9981d35ef63a/arc-05.png

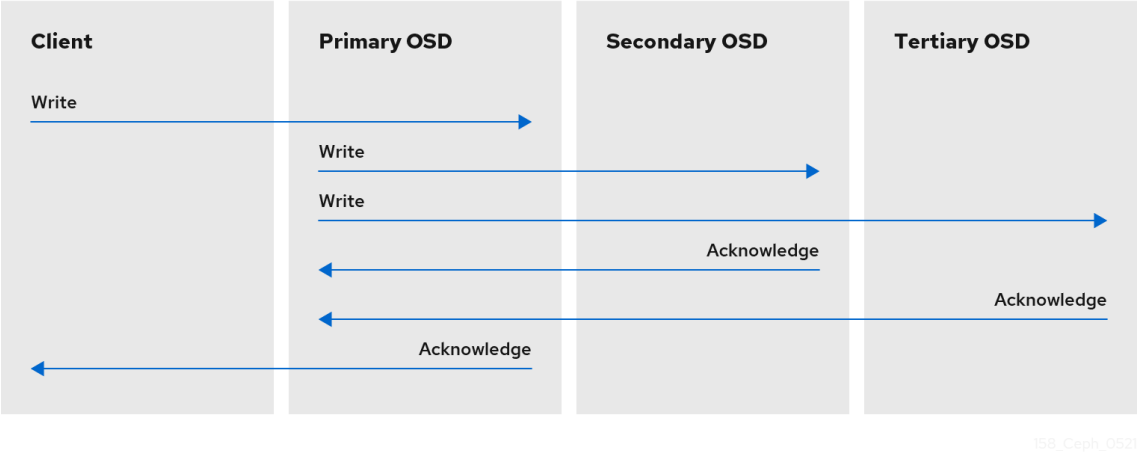

Figura 9: Scrittura repliche.

Per ulteriori dettagli si veda [1].

7 Benchmark

Per valutare la configurazione definita in fase di implementazione e l'adeguatezza in caso di integrazione con altri sistemi, è stato eseguita una sessione di benchmark e, una volta raccolti i dati, sono stati prodotti alcuni grafici, riportati di seguito.

Per poter valutare obbiettivamente le prestazioni del cluster, è importante individuare quindi quali sono le prestazioni dei device sui quali sono state definite architettura e configurazione, per questo nei grafici che seguono sono sempre presenti anche i dati rilevati per i singoli device rispetto al modello di benchmark corrispondente.

Le modalità di accesso allo storage oggetto del benchmark sono le seguenti:

- **sequential writes - 1 MB** scrittura di un file di grosse dimensioni (es. backup).
- **sequential reads - 1 MB** lettura di un file di grosse dimensioni (es. backup).
- **random writes - 64 kB** generico workload in scrittura.
- **random reads - 64 kB** genrico workload in lettura.
- **random writes - 8 kB** modalità di scrittura di un DBMS.
- **random reads - 8 kB** modalità di lettura di un DBMS.
- **single thread random writes - 4 kB** scrittura di un RDBMS (con immediato flush delle modifiche).
- **single thread random reads - 4 kB** simula la lettura di un RDBMS.

Per ogni run, e quindi per ogni modalità di accesso, sono riportati i valori misurati di bandwidth (aggregata per gli 8 processi concorrenti) relativi ai device o servizi:

- **ssd** prestazioni baseline per il singolo disco SSD.
- **hdd** prestazioni baseline per il singolo disco HDD.
- **CephFS** volume CephFS basato su un pool ospitato su dischi rotativi e DB su SSD.
- **CephFS HDD only** volume CephFS basato su un pool ospitato su soli dischi rotativi.
- **CephFS SSD only** volume CephFS basato su un pool ospitato su soli SSD.
- **RADOS** pool RADOS ospitato su dischi rotativi e DB su SSD.
- **RADOS HDD only** pool RADOS ospitato su soli dischi rotativi.
- **RADOS SSD only** pool RADOS ospitato su soli SSD.
- **RBD** immagine RBD basata su un pool ospitato su dischi rotativi e DB su SSD.
- **RBD HDD only** immagine RBD basata su un pool ospitato su soli dischi rotativi.
- **RBD SSD only** immagine RBD basata su un pool ospitato su soli SSD.

Fondamentalmente, durante la fase di benchmark sono state valutate le prestazioni di tre differenti configurazioni dell'intero cluster; in particolare, oltre alla configurazione descritta nella sezione 6.3 (data block su dischi rotativi e DB su SSD), sono state analizzate le configurazioni dove DB e WAL risiedono completamente su dischi rotativi o SSD.

Per tutti i workload è stato lanciato un run con 8 processi paralleli, equivalenti alla simulazione di 8 client che accedono contemporaneamente allo storage, con dimensioni della coda di I/O variabile. Per i run basati su single thread la lunghezza della coda di I/O è pari a 1.

Tutti i run del benchmark sono stati effettuati con FIO [9].

7.1 Risultati ottenuti

Per sistemi di storage con architetture complesse come quello in esame risulta complicato individuare con esattezza una configurazione che risponda a diversi casi d'uso in maniera efficace. Perciò, al di la dell'osservanza delle best practices, l'unico modo per valutare accuratamente le prestazioni e definire le configurazioni che possano sostenere il workload atteso è utilizzare degli specifici tool di benchmarking ed analizzare i dati ottenuti per validare o riformulare le proprie scelte.

[9] `https://fio.readthedocs.io/en/latest/fio_doc.html`

Di seguito sono riportati i risultati misurati nella sessione di benchmark, separati per tipologia di workload. In ogni tabella sono riportati i valori aggregati (rispetto ad gruppo di 8 processi concorrenti) medi di bandwidth in lettura e scrittura. Per ogni gruppo di dati è poi stato disegnato un grafico per meglio comprendere quali sono le differenze prestazionali tra le differenti configurazioni, e come si discostano dalle performance erogate dai singoli HDD e SSD.

7.1.1 Accesso sequenziale

Tabella 4: Sequential access - 1MB block - iodepth 8 - 8 jobs

	AVG Write BW (MB/s)	AVG Read BW (MB/s)
SSD	440.00	562.00
HDD	248.00	224.00
CephFS	1171.00	732.00
CephFS HDD only	642.00	681.00
CephFS SSD only	1128.00	1172.00
RBD	536.00	1470.00
RBD HDD only	432.00	1610.00
RBD SSD only	1037.00	1483.00
RADOS	1172.00	66.80
RADOS HDD only	1173.00	70.80
RADOS SSD only	1172.00	71.30

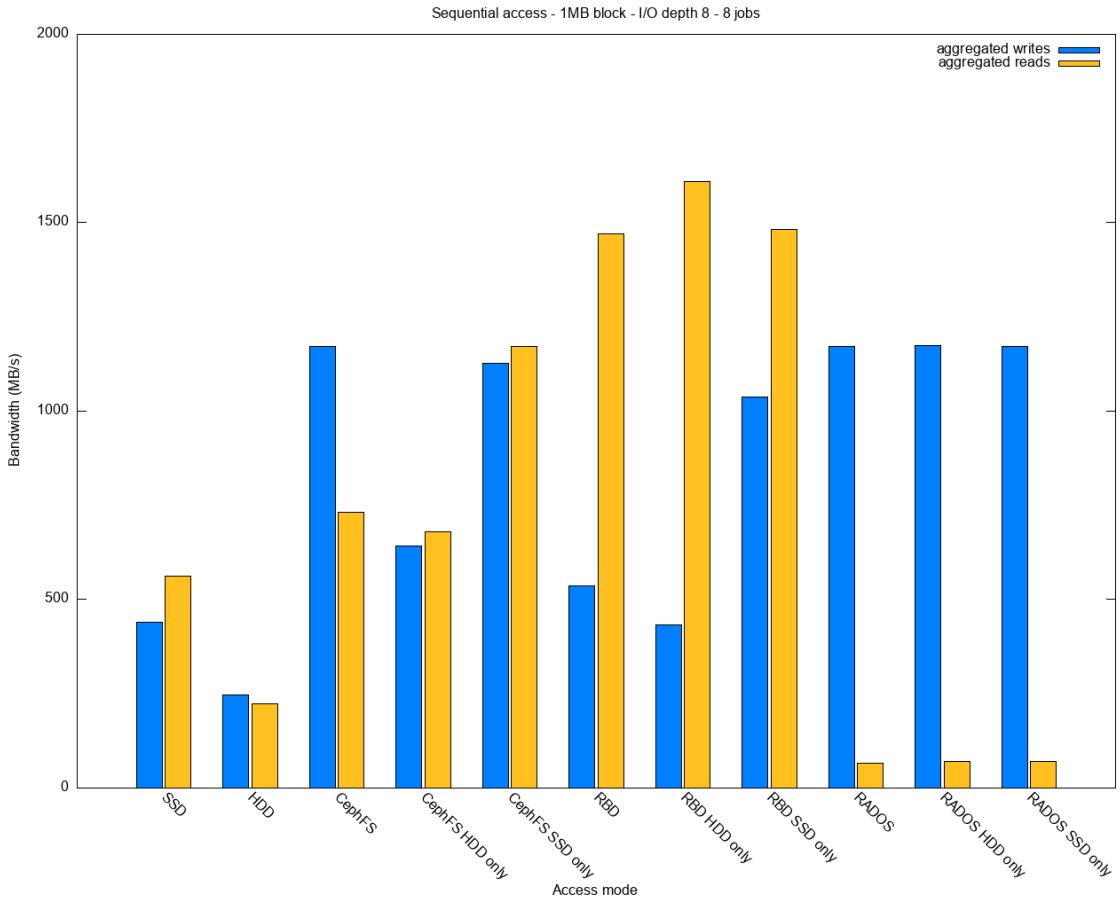

Figura 10: Sequential access - 1M block - iodepth 8.

I grafici riportati per le scritture e letture sequenziali mostrano un risultato abbastanza equilibrato: le varie modalità di accesso hanno prestazioni tutto sommato confrontabili. Sono particolarmente indicati per il supporto a questo tipo di workload CephFS in caso di frequenti scritture e RBD in caso prevalgano le operazioni di lettura.

7.1.2 *Accesso casuale*

Tabella 5: Random access - 64kB block - iodepth 16 - 8 jobs

	AVG Write BW (MB/s)	AVG Read BW (MB/s)
SSD	431.00	405.00
HDD	25.20	26.30
CephFS	1060.00	183.00
CephFS HDD only	115.00	184.00
CephFS SSD only	1151.00	1121.00
RBD	526.00	129.00
RBD HDD only	83.30	173.00
RBD SSD only	1160.00	1158.00
RADOS	1166.00	5140.00
RADOS HDD only	1167.00	5274.00
RADOS SSD only	1166.00	5245.00

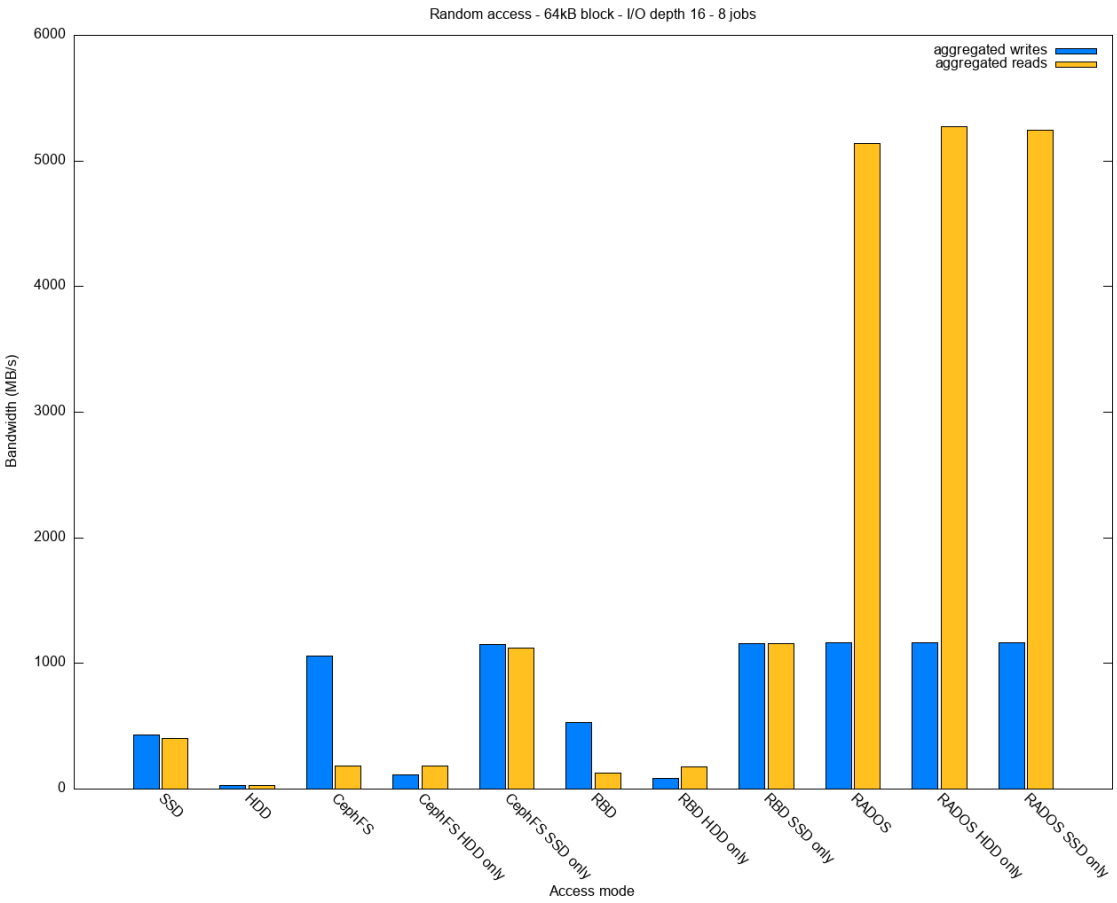

Figura 11: Random access - 64k block - iodepth 16.

Tabella 6: Random access - 8k block - iodepth 32 - 8 jobs

	AVG Write BW (MB/s)	AVG Read BW (MB/s)
SSD	373.00	328.00
HDD	3.65	3.69
CephFS	21.50	23.90
CephFS HDD only	19.60	23.60
CephFS SSD only	347.00	434.00
RBD	71.40	76.60
RBD HDD only	77.90	35.60
RBD SSD only	128.00	655.00
RADOS	523.00	773.00
RADOS HDD only	515.00	735.00
RADOS SSD only	524.00	707.00

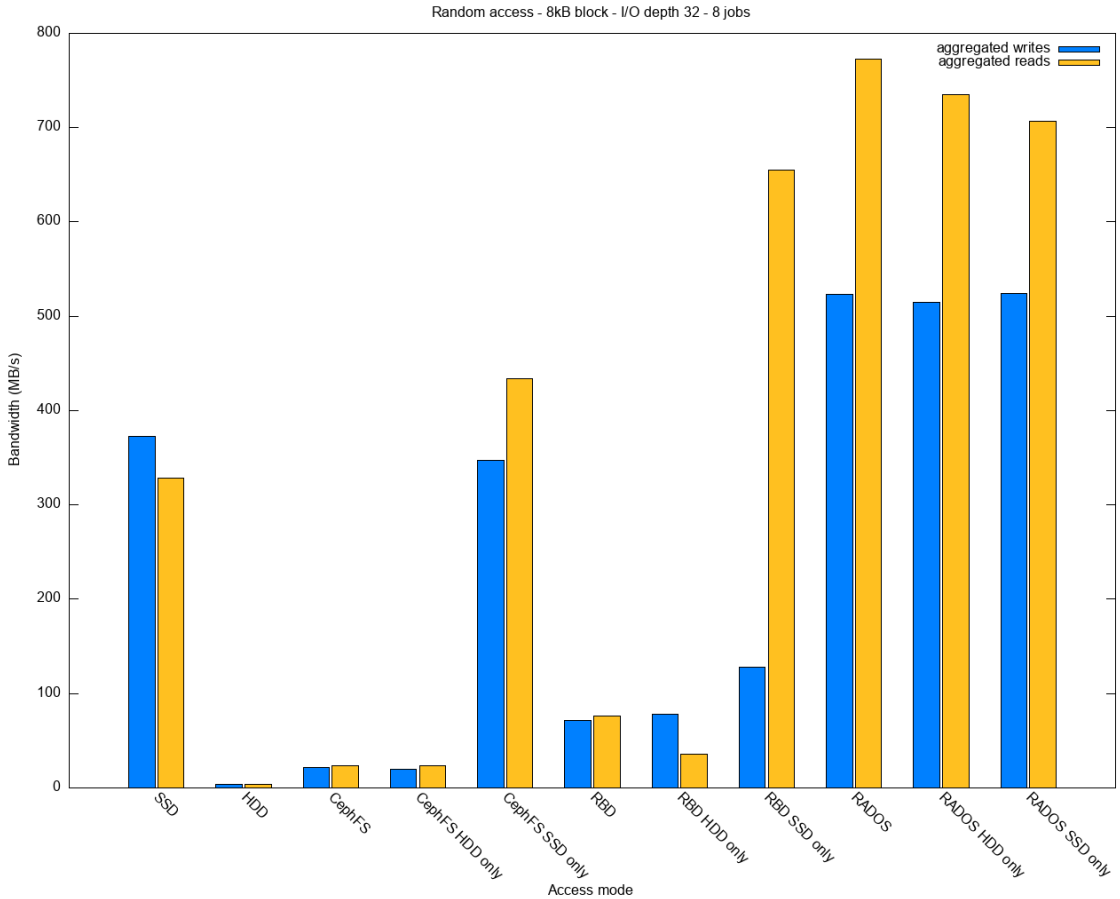

Figura 12: Random access - 8k block - iodepth 32.

Le misurazioni rilevate danno l'idea di quanto RADOS (che si noti è lo strato di storage sul quale si basano tutti gli altri servizi) sia efficace in workload basati su accesso casuale. Non essendo però adatto a rispondere ad ogni caso d'uso (ad esempio quando non è pos-

sibile utilizzare direttamente degli storage ad oggetti) dovranno essere scelti altri servizi adatti ma con prestazioni più ridotte.

7.1.3 Accesso sequenziale single thread

Tabella 7: Single thread random access - 4kB block - iodepth 1 - 8 jobs

	AVG Write BW (MB/s)	AVG Read BW (MB/s)
SSD	266.00	216.00
HDD	1.88	1.47
CephFS	10.90	4.50
CephFS HDD only	10.00	4.46
CephFS SSD only	34.30	68.90
RBD	4.65	270.00
RBD HDD only	0.63	1754.00
RBD SSD only	4.14	1241.00
RADOS	41.80	87.40
RADOS HDD only	27.70	86.50
RADOS SSD only	39.70	86.10

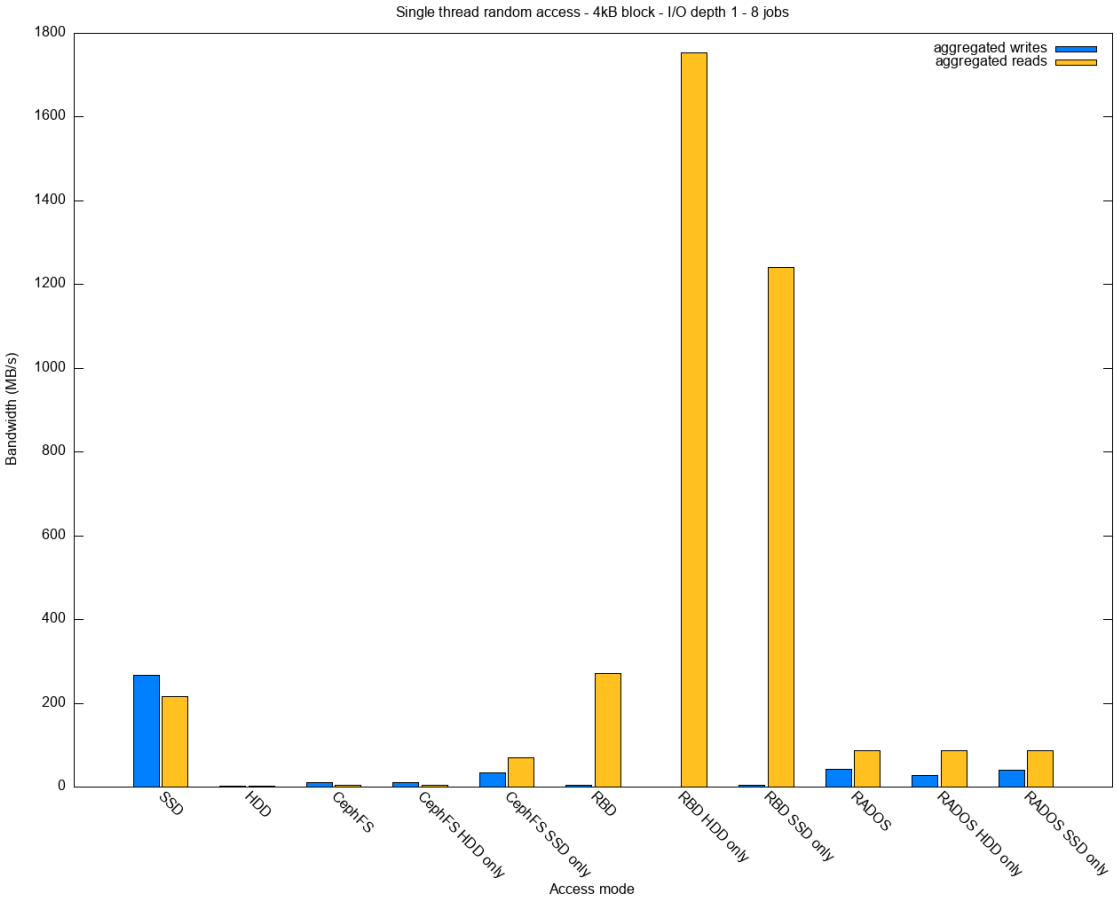

Figura 13: Single thread random access - 4k block - iodepth 1.

Per questo workload, il servizio RBD ha mostrato prestazioni molto interessanti, soprattuto per quanto riguarda le operazioni di lettura. Il modello di workload è molto specifico, ed è spesso un caso critico per la maggior parte dei sistemi di storage, dove scritture di questa natura hanno un impatto consistente sulle performance.

8 Conclusioni

Le misurazioni effettuate hanno dimostrato come, rispetto alle baseline, la configurazione scelta abbia prestazioni apprezzabili.

Un altro aspetto da considerare è che il confronto con una configurazione basata puramente su device a stato solido indica un livello prestazionale generalmente superiore ma non così eclatante da giustificare un costo per unità di capacità che risulterebbe abbastanza elevato. Quindi, nello scegliere una possibile configurazione, è altrettanto importante ottenere un trade-off tra le prestazioni e il costo delle configurazioni, se lo specifico caso d'uso lo consente.

Un'altro vantaggio nell'analisi attraverso i benchmark per un sistema di storage multi-protocollo come Ceph è quello di poter individuare, per ogni specifico workload, il servizio che offrirebbe le prestazioni ottimali. Ad esempio, rifacendosi ai risultati ottenuti, potremmo scegliere di utilizzare CephFS per ospitare dei contenuti di dimensioni significative, oppure optare per RBD nel supportare l'esecuzione di un RDBMS con frequenti letture attese.

In definitiva, si ritiene che la configurazione presentata alla sezione 6.3 possa essere un'opzione bilanciata per ospitare una serie di informazioni di diversa natura con prestazioni più che accettabili. La stessa consente inoltre di ottenere un buon bilanciamento tra capacità effettiva e costi. Inoltre lo schema prescelto consente anche di minimizzare l'impatto di eventuali guasti, riducendo l'interdipendenza tra i componenti hardware in gioco.

Per ottenere ulteriori avanzamenti nelle prestazioni erogabili e nell'ottica di aumentare ulteriormente la resilienza ai guasti, il cluster potrà essere esteso con l'introduzione di hardware aggiuntivo attraverso delle procedure automatizzate che non richiederebbero tempi di fermo.

Riferimenti bibliografici

[1] Sage A. Weil *et al*, Rados: a scalable, reliable storage service for petabyte-scale storage clusters, in: Proceedings of the 2nd international workshop on Petascale data storage: held in conjunction with Supercomputing'07, **1**, 35-44 (2007).