Esaminare la progettazione della soluzione

BeeGFS on NetApp with E-Series Storage

NetApp
March 21, 2024

This PDF was generated from https://docs.netapp.com/it-it/beegfs/beegfs-design-overview.html on March 21, 2024. Always check docs.netapp.com for the latest.

Sommario

Esaminare la progettazione della soluzione

Panoramica del design

Per supportare la soluzione BeeGFS su NetApp, che combina il file system parallelo BeeGFS con i sistemi storage NetApp EF600, sono necessarie apparecchiature, cablaggi e configurazioni specifiche.

Scopri di più:

- "Configurazione dell'hardware"
- "Configurazione del software"
- "Verifica del progetto"
- "Linee guida per il dimensionamento"
- "Tuning delle performance"

Architetture derivate con variazioni di design e performance:

- "Building Block ad alta capacità"

Configurazione dell'hardware

La configurazione hardware per BeeGFS su NetApp include nodi di file e cablaggio di rete.

Configurazione del nodo del file

I file node hanno due socket CPU configurati come zone NUMA separate, che includono l'accesso locale a un numero uguale di slot PCIe e memoria.

Gli adattatori InfiniBand devono essere inseriti nei riser o negli slot PCI appropriati, in modo da bilanciare il carico di lavoro sulle corsie PCIe e sui canali di memoria disponibili. Il carico di lavoro viene bilanciato isolando completamente il lavoro per i singoli servizi BeeGFS in un nodo NUMA specifico. L'obiettivo è quello di ottenere performance simili da ogni nodo di file come se si trattasse di due server a socket singolo indipendenti.

La figura seguente mostra la configurazione NUMA del nodo del file.

[]

I processi BeeGFS vengono bloccati in una particolare zona NUMA per garantire che le interfacce utilizzate si trovino nella stessa zona. Questa configurazione evita la necessità di un accesso remoto sulla connessione tra socket. La connessione tra socket è talvolta nota come collegamento QPI o GMI2; anche nelle moderne architetture di processore, può essere un collo di bottiglia quando si utilizza una rete ad alta velocità come HDR InfiniBand.

Configurazione del cablaggio di rete

All'interno di un building block, ogni nodo di file è connesso a due nodi a blocchi utilizzando un totale di quattro

connessioni InfiniBand ridondanti. Inoltre, ciascun nodo di file dispone di quattro connessioni ridondanti alla rete di storage InfiniBand.

Nella figura seguente, notare che:

- Tutte le porte dei nodi di file delineate in verde vengono utilizzate per la connessione al fabric di storage; tutte le altre porte dei nodi di file sono le connessioni dirette ai nodi di blocco.
- Due porte InfiniBand in una zona NUMA specifica si collegano ai controller A e B dello stesso nodo a blocchi.
- Le porte nel nodo NUMA 0 si connettono sempre al primo nodo a blocchi.
- Le porte nel nodo NUMA 1 si connettono al secondo nodo a blocchi.

[]

> Per le reti di storage con switch ridondanti, le porte evidenziate in verde chiaro devono essere collegate a uno switch e le porte in verde scuro a un altro switch.

La configurazione del cablaggio illustrata nella figura consente a ciascun servizio BeeGFS di:

- Viene eseguito nella stessa zona NUMA indipendentemente dal nodo del file che esegue il servizio BeeGFS.
- Disporre di percorsi secondari ottimali per la rete di storage front-end e per i nodi a blocchi back-end, indipendentemente da dove si verifica un guasto.
- Ridurre al minimo gli effetti delle performance se un nodo di file o un controller in un nodo a blocchi richiede manutenzione.

Cablaggio per sfruttare la larghezza di banda

Per sfruttare l'intera larghezza di banda bidirezionale PCIe, assicurarsi che una porta di ciascun adattatore InfiniBand si colleghi al fabric di storage e l'altra porta si colleghi a un nodo a blocchi. La velocità massima teorica di una porta HDR InfiniBand è di 25 Gbps (senza tenere conto della segnalazione e di altri overhead). La larghezza di banda massima single-direction di uno slot PCIe 4.0 x16 è di 32 Gbps, creando un potenziale collo di bottiglia durante l'implementazione di nodi di file che incorporano adattatori InfiniBand a doppia porta in grado teoricamente di gestire 50 GBps di larghezza di banda.

La figura seguente mostra il design del cablaggio utilizzato per sfruttare l'intera larghezza di banda bidirezionale PCIe.

[]

Per ogni servizio BeeGFS, utilizzare lo stesso adattatore per connettere la porta preferita utilizzata per il traffico client con il percorso al controller dei nodi a blocchi che è il principale proprietario di tali volumi di servizi. Per ulteriori informazioni, vedere "Configurazione del software".

Configurazione del software

La configurazione software per BeeGFS su NetApp include componenti di rete BeeGFS, nodi a blocchi EF600, nodi di file BeeGFS, gruppi di risorse e servizi BeeGFS.

Configurazione di rete BeeGFS

La configurazione di rete di BeeGFS è costituita dai seguenti componenti.

- **IP mobili** gli IP mobili sono un tipo di indirizzo IP virtuale che può essere instradato dinamicamente a qualsiasi server della stessa rete. Più server possono avere lo stesso indirizzo IP mobile, ma possono essere attivi solo su un server alla volta.

  Ciascun servizio del server BeeGFS dispone di un proprio indirizzo IP che può spostarsi tra i nodi di file in base alla posizione di esecuzione del servizio del server BeeGFS. Questa configurazione IP mobile consente a ciascun servizio di eseguire il failover in modo indipendente sull'altro nodo di file. Il client deve semplicemente conoscere l'indirizzo IP di un particolare servizio BeeGFS; non deve sapere quale nodo di file sta eseguendo quel servizio.

- **Configurazione multi-homing del server BeeGFS** per aumentare la densità della soluzione, ciascun nodo di file dispone di più interfacce di storage con IP configurati nella stessa subnet IP.

  È necessaria un'ulteriore configurazione per garantire che questa configurazione funzioni come previsto con lo stack di rete Linux, perché per impostazione predefinita, le richieste a un'interfaccia possono essere risposte su un'interfaccia diversa se i relativi IP si trovano nella stessa subnet. Oltre ad altri inconvenienti, questo comportamento predefinito rende impossibile stabilire o mantenere correttamente le connessioni RDMA.

  L'implementazione basata su Ansible gestisce il rafforzamento del comportamento del percorso inverso (RP) e del protocollo di risoluzione degli indirizzi (ARP), oltre a garantire che quando gli IP mobili vengono avviati e arrestati; i corrispondenti percorsi e le regole IP vengono creati dinamicamente per consentire alla configurazione di rete multihomed di funzionare correttamente.

- **La configurazione multi-rail del client BeeGFS** *Multi-rail* si riferisce alla capacità di un'applicazione di utilizzare più "guide" di rete indipendenti per aumentare le performance.

  Sebbene BeeGFS possa utilizzare RDMA per la connettività, BeeGFS utilizza IPoIB per semplificare il rilevamento e la creazione di connessioni RDMA. Per consentire ai client BeeGFS di utilizzare più interfacce InfiniBand, è possibile configurare ciascun client con un indirizzo IP situato in una subnet diversa e quindi configurare le interfacce preferite per metà dei servizi del server BeeGFS in ciascuna subnet.

  Nel diagramma seguente, le interfacce evidenziate in verde chiaro si trovano in una subnet IP (ad esempio, `100.127.0.0/16`) e le interfacce di colore verde scuro si trovano in un'altra subnet (ad esempio, `100.128.0.0/16`).

  La figura seguente mostra il bilanciamento del traffico tra più interfacce client BeeGFS.

  []

  Poiché ogni file in BeeGFS viene in genere sottoposto a striping su più servizi di storage, la configurazione multi-rail consente al client di ottenere un throughput superiore a quello possibile con una singola porta InfiniBand. Ad esempio, il seguente esempio di codice mostra una configurazione di striping comune dei file che consente al client di bilanciare il traffico tra entrambe le interfacce:

```
root@ictad21h01:/mnt/beegfs# beegfs-ctl --getentryinfo myfile
Entry type: file
EntryID: 11D-624759A9-65
Metadata node: meta_01_tgt_0101 [ID: 101]
Stripe pattern details:
+ Type: RAID0
+ Chunksize: 1M
+ Number of storage targets: desired: 4; actual: 4
+ Storage targets:
  + 101 @ stor_01_tgt_0101 [ID: 101]
  + 102 @ stor_01_tgt_0101 [ID: 101]
  + 201 @ stor_02_tgt_0201 [ID: 201]
  + 202 @ stor_02_tgt_0201 [ID: 201]
```

L'utilizzo di due subnet IPoIB è una distinzione logica. Se lo si desidera, è possibile utilizzare una singola subnet fisica InfiniBand (rete di storage).

> Il supporto multi-rail è stato aggiunto in BeeGFS 7.3.0 per consentire l'utilizzo di più interfacce IB in una singola subnet IPoIB. La progettazione della soluzione BeeGFS su NetApp è stata sviluppata prima della disponibilità generale di BeeGFS 7.3.0, dimostrando così l'utilizzo di due subnet IP per utilizzare due interfacce IB sui client BeeGFS. Uno dei vantaggi dell'approccio a più subnet IP consiste nell'eliminare la necessità di configurare il multihoming sui nodi client BeeGFS (per ulteriori informazioni, vedere "Supporto di BeeGFS RDMA").

Configurazione del nodo a blocchi EF600

I nodi a blocchi sono costituiti da due controller RAID attivi/attivi con accesso condiviso allo stesso set di dischi. In genere, ciascun controller possiede la metà dei volumi configurati sul sistema, ma può sostituire l'altro controller in base alle necessità.

Il software multipathing sui nodi di file determina il percorso attivo e ottimizzato per ciascun volume e si sposta automaticamente sul percorso alternativo in caso di guasto di un cavo, di un adattatore o di un controller.

Il seguente diagramma mostra il layout del controller nei nodi a blocchi EF600.

[]

Per facilitare la soluzione ha con disco condiviso, i volumi vengono mappati su entrambi i nodi di file in modo che possano assumere il controllo reciproco in base alle necessità. Il seguente diagramma mostra un esempio di configurazione del servizio BeeGFS e della proprietà del volume preferita per ottenere le massime prestazioni. L'interfaccia a sinistra di ciascun servizio BeeGFS indica l'interfaccia preferita che i client e gli altri servizi utilizzano per contattarlo.

[]

Nell'esempio precedente, client e servizi server preferiscono comunicare con il servizio di storage 1 utilizzando l'interfaccia i1b. Il servizio di storage 1 utilizza l'interfaccia i1a come percorso preferito per comunicare con i suoi volumi (storage_tgt_101, 102) sul controller A del primo nodo a blocchi. Questa configurazione utilizza l'intera larghezza di banda PCIe bidirezionale disponibile per l'adattatore InfiniBand e ottiene prestazioni

migliori da un adattatore HDR InfiniBand a due porte rispetto a quanto sarebbe altrimenti possibile con PCIe 4.0.

Configurazione del nodo del file BeeGFS

I nodi di file BeeGFS sono configurati in un cluster ad alta disponibilità (ha) per facilitare il failover dei servizi BeeGFS tra più nodi di file.

La progettazione del cluster ha si basa su due progetti Linux ha ampiamente utilizzati: Corosync per l'appartenenza al cluster e Pacemaker per la gestione delle risorse del cluster. Per ulteriori informazioni, vedere "Training Red Hat per add-on ad alta disponibilità".

NetApp ha creato ed esteso diversi agenti di risorse OCF (Open Cluster Framework) per consentire al cluster di avviare e monitorare in modo intelligente le risorse BeeGFS.

Cluster BeeGFS ha

In genere, quando si avvia un servizio BeeGFS (con o senza ha), è necessario disporre di alcune risorse:

- Indirizzi IP raggiungibili dal servizio, generalmente configurati da Network Manager.
- File system sottostanti utilizzati come destinazione per BeeGFS per l'archiviazione dei dati.

  Questi sono in genere definiti in `/etc/fstab` E montato da Systemd.

- Un servizio Systemd responsabile dell'avvio dei processi BeeGFS quando le altre risorse sono pronte.

  Senza software aggiuntivo, queste risorse vengono avviate solo su un singolo nodo di file. Pertanto, se il nodo del file non è in linea, una parte del file system BeeGFS non è accessibile.

Dato che più nodi possono avviare ciascun servizio BeeGFS, Pacemaker deve assicurarsi che ogni servizio e le risorse dipendenti siano in esecuzione su un solo nodo alla volta. Ad esempio, se due nodi tentano di avviare lo stesso servizio BeeGFS, esiste il rischio di danneggiamento dei dati se entrambi tentano di scrivere negli stessi file sulla destinazione sottostante. Per evitare questo scenario, Pacemaker si affida a Corosync per mantenere in modo affidabile lo stato del cluster complessivo in sincronia tra tutti i nodi e stabilire il quorum.

Se si verifica un errore nel cluster, Pacemaker reagisce e riavvia le risorse BeeGFS su un altro nodo. In alcuni scenari, Pacemaker potrebbe non essere in grado di comunicare con il nodo guasto originale per confermare che le risorse sono state interrotte. Per verificare che il nodo sia inattivo prima di riavviare le risorse BeeGFS altrove, Pacemaker interrompe il nodo difettoso, idealmente rimuovendo l'alimentazione.

Sono disponibili molti agenti di scherma open-source che consentono a Pacemaker di recinzione di un nodo con un'unità di distribuzione dell'alimentazione (PDU) o utilizzando il server Baseboard Management Controller (BMC) con API come Redfish.

Quando BeeGFS viene eseguito in un cluster ha, tutti i servizi BeeGFS e le risorse sottostanti vengono gestiti da Pacemaker in gruppi di risorse. Ogni servizio BeeGFS e le risorse da cui dipende sono configurati in un gruppo di risorse, che garantisce che le risorse vengano avviate e interrotte nell'ordine corretto e collocate sullo stesso nodo.

Per ciascun gruppo di risorse BeeGFS, Pacemaker esegue una risorsa di monitoraggio BeeGFS personalizzata responsabile del rilevamento delle condizioni di guasto e dell'attivazione intelligente dei failover quando un servizio BeeGFS non è più accessibile su un nodo specifico.

La figura seguente mostra i servizi e le dipendenze BeeGFS controllati da pacemaker.

[]

Per avviare più servizi BeeGFS dello stesso tipo sullo stesso nodo, Pacemaker è configurato per avviare i servizi BeeGFS utilizzando il metodo di configurazione Multi Mode. Per ulteriori informazioni, consultare "Documentazione BeeGFS su Multi Mode".

Poiché i servizi BeeGFS devono essere in grado di avviarsi su più nodi, il file di configurazione per ciascun servizio (normalmente situato in `/etc/beegfs`) Viene memorizzato in uno dei volumi e-Series utilizzati come destinazione BeeGFS per quel servizio. In questo modo, la configurazione e i dati di un particolare servizio BeeGFS sono accessibili a tutti i nodi che potrebbero aver bisogno di eseguire il servizio.

```
# tree stor_01_tgt_0101/ -L 2
stor_01_tgt_0101/
├── data
│   ├── benchmark
│   ├── buddymir
│   ├── chunks
│   ├── format.conf
│   ├── lock.pid
│   ├── nodeID
│   ├── nodeNumID
│   ├── originalNodeID
│   ├── targetID
│   └── targetNumID
└── storage_config
    ├── beegfs-storage.conf
    ├── connInterfacesFile.conf
    └── connNetFilterFile.conf
```

Verifica del progetto

La progettazione di seconda generazione per la soluzione BeeGFS su NetApp è stata verificata utilizzando tre profili di configurazione a blocchi.

I profili di configurazione includono quanto segue:

- Un singolo building block di base, che include gestione BeeGFS, metadati e servizi di storage.
- Metadati BeeGFS più un building block per lo storage.
- Un building block BeeGFS solo per lo storage.

I blocchi costitutivi erano collegati a due switch Mellanox Quantum InfiniBand (MQM8700). Dieci client BeeGFS sono stati collegati anche agli switch InfiniBand e utilizzati per eseguire utility di benchmark sintetiche.

La figura seguente mostra la configurazione BeeGFS utilizzata per validare la soluzione BeeGFS su NetApp.

[]

Striping del file BeeGFS

Un vantaggio dei file system paralleli è la capacità di eseguire lo striping di singoli file su più destinazioni di storage, che potrebbero rappresentare volumi sullo stesso sistema di storage sottostante o su sistemi di storage diversi.

In BeeGFS, è possibile configurare lo striping per directory e per file per controllare il numero di destinazioni utilizzate per ogni file e per controllare la dimensione del blocco (o dimensione del blocco) utilizzata per ogni stripe di file. Questa configurazione consente al file system di supportare diversi tipi di carichi di lavoro e profili i/o senza dover riconfigurare o riavviare i servizi. È possibile applicare le impostazioni di stripe utilizzando `beegfs-ctl` Tool della riga di comando o con applicazioni che utilizzano l'API di striping. Per ulteriori informazioni, consultare la documentazione di BeeGFS per "Striping" e. "API di striping".

Per ottenere le migliori prestazioni, i modelli di stripe sono stati regolati durante l'intero test e vengono annotati i parametri utilizzati per ciascun test.

Test della larghezza di banda IOR: Client multipli

I test della larghezza di banda IOR hanno utilizzato OpenMPI per eseguire lavori paralleli dello strumento IOR (Synthetic i/o Generator Tool), disponibile presso "HPC GitHub") Su tutti i 10 nodi client a uno o più blocchi di base BeeGFS. Se non diversamente specificato:

- Tutti i test utilizzavano l'i/o diretto con una dimensione di trasferimento di 1 MiB.
- Lo striping dei file BeeGFS è stato impostato su una dimensione di blocco di 1 MB e su una destinazione per file.

I seguenti parametri sono stati utilizzati per IOR con il conteggio dei segmenti regolato per mantenere la dimensione del file aggregato a 5TiB per un building block e 40TiB per tre building block.

```
mpirun --allow-run-as-root --mca btl tcp -np 48 -map-by node -hostfile
10xnodes ior -b 1024k --posix.odirect -e -t 1024k -s 54613 -z -C -F -E -k
```

Un building block di base BeeGFS (gestione, metadati e storage)

La figura seguente mostra i risultati del test IOR con un singolo building block di base BeeGFS (gestione, metadati e storage).

[]

Metadati BeeGFS + building block dello storage

La figura seguente mostra i risultati del test IOR con un singolo metadata BeeGFS + building block dello storage.

[]

Building block BeeGFS solo per lo storage

La figura seguente mostra i risultati del test IOR con un singolo building block BeeGFS solo per lo storage.

[]

Tre blocchi di base BeeGFS

La figura seguente mostra i risultati del test IOR con tre blocchi di base BeeGFS.

[]

Come previsto, la differenza di performance tra il building block di base e i successivi metadati + building block di storage è trascurabile. Il confronto tra i metadati e il building block di storage e un building block di solo storage mostra un leggero aumento delle performance di lettura dovuto ai dischi aggiuntivi utilizzati come target di storage. Tuttavia, non vi è alcuna differenza significativa nelle prestazioni di scrittura. Per ottenere performance più elevate, è possibile aggiungere più elementi di base insieme per scalare le performance in modo lineare.

Test della larghezza di banda IOR: Singolo client

Il test della larghezza di banda IOR ha utilizzato OpenMPI per eseguire più processi IOR utilizzando un singolo server GPU dalle performance elevate per esplorare le performance ottenibili da un singolo client.

Questo test confronta anche il comportamento di rilettura e le performance di BeeGFS quando il client è configurato per utilizzare la paging-cache del kernel Linux (`tuneFileCacheType = native`) rispetto al valore predefinito `buffered` impostazione.

La modalità di caching nativa utilizza la paging-cache del kernel Linux sul client, consentendo alle operazioni di rilettura di provenire dalla memoria locale invece di essere ritrasmesse sulla rete.

Il diagramma seguente mostra i risultati del test IOR con tre blocchi di base BeeGFS e un singolo client.

[]

Lo striping di BeeGFS per questi test è stato impostato su una dimensione del blocco di 1 MB con otto destinazioni per file.

Sebbene le performance di scrittura e lettura iniziale siano più elevate utilizzando la modalità buffer predefinita, per i carichi di lavoro che rileggono gli stessi dati più volte, la modalità caching nativa offre un significativo miglioramento delle performance. Questo miglioramento delle performance di rilettura è importante per i carichi di lavoro come il deep learning che rileggono lo stesso set di dati più volte in diverse epoche.

Test delle performance dei metadati

I test delle prestazioni dei metadati hanno utilizzato lo strumento MDTest (incluso come parte di IOR) per misurare le prestazioni dei metadati di BeeGFS. I test hanno utilizzato OpenMPI per eseguire lavori paralleli su tutti e dieci i nodi client.

I seguenti parametri sono stati utilizzati per eseguire il test di benchmark con il numero totale di processi scalati da 10 a 320 in incrementi di 2x e con una dimensione del file di 4k.

```
mpirun -h 10xnodes –map-by node np $processes mdtest -e 4k –w 4k -i 3 -I
16 -z 3 -b 8 -u
```

Le performance dei metadati sono state misurate prima con uno e due metadati + building block di storage per mostrare come le performance si ridimensionano aggiungendo ulteriori building block.

Un solo metadata BeeGFS + building block di storage

Il seguente diagramma mostra i risultati di MDTest con un solo metadata BeeGFS + blocchi di base dello storage.

[]

Due metadati BeeGFS + blocchi di base per lo storage

Il seguente diagramma mostra i risultati di MDTest con due metadati BeeGFS + blocchi di base dello storage.

[]

Validazione funzionale

Nell'ambito della convalida di questa architettura, NetApp ha eseguito diversi test funzionali, tra cui:

- Errore di una singola porta InfiniBand client disattivando la porta dello switch.
- Errore di una porta InfiniBand di un singolo server disattivando la porta dello switch.
- Attivazione dello spegnimento immediato del server mediante BMC.
- Posizionamento corretto di un nodo in standby e failover del servizio su un altro nodo.
- Posizionamento corretto di un nodo di nuovo online e fallimento dei servizi di back nel nodo originale.
- Spegnere uno degli switch InfiniBand utilizzando la PDU. Tutti i test sono stati eseguiti mentre era in corso il test di stress con `sysSessionChecksEnabled: false` Set di parametri sui client BeeGFS. Non sono stati osservati errori o interruzioni dell'i/O.

> Si è verificato un problema noto (vedere "Changelog") Quando le connessioni RDMA client/server BeeGFS vengono interrompute inaspettatamente, a causa della perdita dell'interfaccia primaria (come definito nella `connInterfacesFile`) O un server BeeGFS non funzionante; l'i/o client attivo può bloccarsi per un massimo di dieci minuti prima della ripresa. Questo problema non si verifica quando i nodi BeeGFS vengono posizionati correttamente in standby o non sono in standby per la manutenzione pianificata o se TCP è in uso.

Convalida NVIDIA DGX A100 SuperPOD e BasePOD

NetApp ha validato una soluzione di storage per NVDIA DGX A100 SuperPOD utilizzando un file system BeeGFS simile costituito da tre blocchi di base con i metadati e il profilo di configurazione dello storage applicato. Il lavoro di qualificazione ha comportato il test della soluzione descritta da questo NVA con venti server GPU DGX A100 che eseguono una varietà di storage, machine learning e benchmark di deep learning. Tutto lo storage certificato per l'utilizzo in NVIDIA DGX A100 SuperPOD è certificato automaticamente per l'utilizzo anche nelle architetture NVIDIA BasePOD.

Per ulteriori informazioni, vedere "NVIDIA DGX SuperPOD con NetApp" e. "NVIDIA DGX BasePOD".

Linee guida per il dimensionamento

La soluzione BeeGFS include consigli per il dimensionamento delle performance e della capacità basati su test di verifica.

L'obiettivo di un'architettura building-block è creare una soluzione semplice da dimensionare aggiungendo più building block per soddisfare i requisiti di un particolare sistema BeeGFS. Utilizzando le linee guida riportate di seguito, è possibile stimare la quantità e i tipi di blocchi di base BeeGFS necessari per soddisfare i requisiti del proprio ambiente.

Tenere presente che queste stime sono le migliori performance del caso. Le applicazioni di benchmarking sintetico vengono scritte e utilizzate per ottimizzare l'utilizzo dei file system sottostanti in modi diversi dalle

applicazioni reali.

Dimensionamento delle performance

La seguente tabella fornisce il dimensionamento delle performance consigliato.

Profilo di configurazione	1 MIB letture	1MiB scrive
Metadati + storage	62GiBps	21 GiBps
Solo storage	64 GiBps	21 GiBps

Le stime di dimensionamento della capacità dei metadati si basano sulla "regola generale" secondo cui 500 GB di capacità sono sufficienti per circa 150 milioni di file in BeeGFS. Per ulteriori informazioni, consultare la documentazione di BeeGFS per "Requisiti di sistema".)

L'utilizzo di funzionalità come gli elenchi di controllo degli accessi e il numero di directory e file per directory influisce anche sulla velocità di utilizzo dello spazio dei metadati. Le stime della capacità dello storage tengono conto della capacità utilizzabile del disco insieme all'overhead di RAID 6 e XFS.

Dimensionamento della capacità per metadati + building block dello storage

La seguente tabella fornisce il dimensionamento della capacità consigliato per i metadati e gli elementi di base dello storage.

Dimensioni del disco (2+2 RAID 1) gruppi di volumi di metadati	Capacità dei metadati (numero di file)	Dimensioni del disco (8+2 RAID 6) gruppi di volumi di storage	Capacità dello storage (contenuto del file)
1,92 TB	1,938,577,200	1,92 TB	51,77 TB
3,84 TB	3,880,388,400	3,84 TB	103,55 TB
7,68 TB	8,125,278,000	7,68 TB	216,74 TB
15,3 TB	17,269,854,000	15,3 TB	460,60 TB

Quando si ridimensionano i metadati e gli elementi di base dello storage, è possibile ridurre i costi utilizzando unità più piccole per i gruppi di volumi di metadati rispetto ai gruppi di volumi di storage.

Dimensionamento della capacità per gli elementi di base solo per lo storage

La seguente tabella fornisce il dimensionamento della capacità a regola d'uso per gli elementi di base solo per lo storage.

Dimensioni del disco (10+2 RAID 6) gruppi di volumi di storage	Capacità dello storage (contenuto del file)
1,92 TB	59,89 TB
3,84 TB	119,80 TB
7,68 TB	251,89 TB
15,3 TB	538,55 TB

L'overhead di performance e capacità dell'inclusione del servizio di gestione nel building block di base (primo) è minimo, a meno che non sia attivato il blocco globale dei file.

Tuning delle performance

La soluzione BeeGFS include consigli per l'ottimizzazione delle performance basati su test di verifica.

Sebbene BeeGFS fornisca performance ragionevoli, NetApp ha sviluppato una serie di parametri di tuning consigliati per massimizzare le performance. Questi parametri tengono conto delle funzionalità dei nodi a blocchi e-Series sottostanti e di eventuali requisiti speciali necessari per eseguire BeeGFS in un'architettura ha a disco condiviso.

Ottimizzazione delle performance per i file node

I parametri di tuning disponibili che è possibile configurare includono:

1. **Impostazioni di sistema in UEFI/BIOS dei nodi di file.** per massimizzare le prestazioni, si consiglia di configurare le impostazioni di sistema sul modello di server utilizzato come nodi di file. È possibile configurare le impostazioni di sistema quando si impostano i nodi di file utilizzando il programma di configurazione del sistema (UEFI/BIOS) o le API Redfish fornite dal controller di gestione della baseboard (BMC).

   Le impostazioni di sistema variano a seconda del modello di server utilizzato come nodo di file. Le impostazioni devono essere configurate manualmente in base al modello di server in uso. Per informazioni su come configurare le impostazioni di sistema per i nodi di file Lenovo SR665 validati, vedere "Ottimizzare le impostazioni del sistema del nodo di file per le performance".

2. **Impostazioni predefinite per i parametri di configurazione richiesti.** i parametri di configurazione richiesti influiscono sulla modalità di configurazione dei servizi BeeGFS e sulla modalità di formattazione e montaggio dei volumi e-Series (dispositivi a blocchi) da parte di Pacemaker. Questi parametri di configurazione richiesti includono:
   - Parametri di configurazione del servizio BeeGFS

     È possibile ignorare le impostazioni predefinite per i parametri di configurazione in base alle esigenze. Per i parametri che è possibile regolare per i carichi di lavoro o i casi di utilizzo specifici, vedere "Parametri di configurazione del servizio BeeGFS".

   - I parametri di montaggio e formattazione del volume sono impostati sui valori predefiniti consigliati e devono essere regolati solo in caso di utilizzo avanzato. I valori predefiniti sono i seguenti:
     - Ottimizza la formattazione iniziale del volume in base al tipo di destinazione (ad esempio gestione, metadati o storage), oltre alla configurazione RAID e alle dimensioni dei segmenti del volume sottostante.
     - Regolare il modo in cui pacemaker monta ciascun volume per assicurarsi che le modifiche vengano immediatamente applicate ai nodi a blocchi e-series. In questo modo si evita la perdita di dati quando i nodi di file non funzionano con scritture attive in corso.

       Per i parametri che è possibile regolare per i carichi di lavoro o i casi di utilizzo specifici, vedere "formattazione dei volumi e parametri di configurazione del montaggio".

3. **Impostazioni di sistema nel sistema operativo Linux installato sui nodi di file.** è possibile ignorare le impostazioni predefinite del sistema operativo Linux quando si crea l'inventario Ansible nel passaggio 4 di

"Creare l'inventario Ansible".

Le impostazioni predefinite sono state utilizzate per validare la soluzione BeeGFS su NetApp, ma è possibile modificarle per adattarle ai carichi di lavoro o ai casi di utilizzo specifici. Di seguito sono riportati alcuni esempi delle impostazioni di sistema del sistema operativo Linux che è possibile modificare:

- Code i/o su dispositivi a blocchi e-Series.

  È possibile configurare le code i/o sui dispositivi a blocchi e-Series utilizzati come destinazioni BeeGFS per:

  - Regolare l'algoritmo di scheduling in base al tipo di dispositivo (NVMe, HDD e così via).
  - Aumentare il numero di richieste in sospeso.
  - Regolare le dimensioni della richiesta.
  - Ottimizza il comportamento di Read ahead.
- Impostazioni della memoria virtuale.

  È possibile regolare le impostazioni della memoria virtuale per ottenere performance di streaming ottimali e costanti.
- Impostazioni della CPU.

  È possibile regolare il regolatore di frequenza della CPU e altre configurazioni della CPU per ottenere le massime prestazioni.
- Dimensione richiesta di lettura.

  È possibile aumentare la dimensione massima della richiesta di lettura per gli HCA Mellanox.

Ottimizzazione delle performance per i nodi a blocchi

In base ai profili di configurazione applicati a un particolare building block BeeGFS, i gruppi di volumi configurati sui nodi a blocchi cambiano leggermente. Ad esempio, con un nodo a blocchi EF600 a 24 dischi:

- Per il singolo building block di base, inclusi i servizi di gestione, metadati e storage BeeGFS:
  - 1 gruppo di volumi RAID 10 2+2 per la gestione di BeeGFS e i servizi di metadati
  - 2 gruppi di volumi 8+2 RAID 6 per i servizi di storage BeeGFS
- Per i metadati BeeGFS + building block di storage:
  - 1 gruppo di volumi RAID 10 2+2 per i servizi di metadati BeeGFS
  - 2 gruppi di volumi 8+2 RAID 6 per i servizi di storage BeeGFS
- Solo per lo storage BeeGFS building block:
  - 2 gruppi di volumi RAID 6 da 10+2 per i servizi di storage BeeGFS

> Poiché BeeGFS ha bisogno di una quantità di spazio di storage significativamente inferiore per la gestione e i metadati rispetto allo storage, un'opzione è quella di utilizzare dischi più piccoli per i gruppi di volumi RAID 10. I dischi più piccoli devono essere inseriti negli slot più esterni. Per ulteriori informazioni, consultare "istruzioni per l'implementazione".

Questi sono tutti configurati dall'implementazione basata su Ansible, insieme a diverse altre impostazioni

generalmente consigliate per ottimizzare performance/comportamento, tra cui:

- Regolare le dimensioni del blocco della cache globale a 32 KiB e regolare il vampate della cache basato sulla domanda al 80%.
- Disattivazione del bilanciamento del carico automatico (per garantire che le assegnazioni dei volumi dei controller rimvengano come previsto).
- Abilitare il caching in lettura e disabilitare il caching Read-ahead.
- Abilitare il caching in scrittura con mirroring e richiedere il backup della batteria, in modo che le cache persistano in caso di guasto di un controller di nodi a blocchi.
- Specifica dell'ordine in cui i dischi vengono assegnati ai gruppi di volumi, bilanciando i/o tra i canali di disco disponibili.

Building block ad alta capacità

Il design della soluzione BeeGFS standard è costruito tenendo in considerazione i carichi di lavoro dalle performance elevate. I clienti che cercano casi di utilizzo ad alta capacità devono osservare le variazioni nelle caratteristiche di progettazione e performance qui delineate.

Configurazione hardware e software

La configurazione hardware e software per l'building block ad alta capacità è standard, ad eccezione del fatto che i controller EF600 devono essere sostituiti con controller EF300 con un'opzione per il collegamento tra 1 e 7 tray di espansione IOM con 60 unità ciascuno per ciascun array di storage, totale da 2 a 14 vassoi di espansione per building block.

I clienti che implementano una progettazione di building block ad alta capacità probabilmente utilizzeranno solo la configurazione di base basata su building block, costituita da servizi di gestione, metadati e storage BeeGFS per ciascun nodo. Per un'efficienza dei costi, i nodi di storage ad alta capacità devono eseguire il provisioning dei volumi di metadati sui dischi NVMe nell'enclosure di controller EF300 e fornire i volumi di storage ai dischi NL-SAS nei vassoi di espansione.

[]

Linee guida per il dimensionamento

Queste linee guida sul dimensionamento presuppongono che gli building block ad alta capacità siano configurati con un gruppo di volumi 2+2 NVMe SSD per i metadati nell'enclosure EF300 di base e 6 gruppi di volumi 8+2 NL-SAS per tray di espansione IOM per lo storage.

Dimensioni del disco (HDD con capacità)	Capacità per BB (1 vassoio)	Capacità per BB (2 tray)	Capacità per BB (3 tray)	Capacità per BB (4 tray)
4 TB	439 TB	878 TB	1317 TB	1756 TB
8 TB	878 TB	1756 TB	2634 TB	3512 TB
10 TB	1097 TB	2195 TB	3292 TB	4390 TB
12 TB	1317 TB	2634 TB	3951 TB	5268 TB

Dimensioni del disco (HDD con capacità)	Capacità per BB (1 vassoio)	Capacità per BB (2 tray)	Capacità per BB (3 tray)	Capacità per BB (4 tray)
16 TB	1756 TB	3512 TB	5268 TB	7024 TB
18 TB	1975 TB	3951 TB	5927 TB	7902 TB

Informazioni sul copyright

Copyright © 2024 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEQUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina http://www.netapp.com/TM sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.