

The background of the slide is a grayscale photograph of a server room. In the foreground, there are several computer monitors and keyboards on desks. In the background, there are tall server racks filled with equipment, with many cables hanging from them. The overall scene is a typical data center or server room environment.

Linux Day Torino
24 ottobre 2026

Oltre lo SCSI
Enterprise Storage moderno con
Linux
da iSCSI a Fibre Channel

Massimo Nuvoli

Mi presento

- Architetto di Sistemi
- Lavoro per me stesso, Progetto Archivio SRL, Dicobit
- Trainer certificato in ambiente tecnico

e ...

- Co-fondatore di Adenda SRL (CTO)

Provider di infrastrutture di rete, con il primo datacenter innovativo a Torino



Mm



ma anche...



Mm

Il caso PHP2Shell

- Due RCE critiche nel runtime PHP
- CVE-2026-109618 e CVE-2026-109619, CVSS 9.2/10
- RCE non autenticata, coi privilegi del worker web
- Colpiti PHP-FPM, mod_php, php-cgi, FrankenPHP, LiteSpeed
- Scoperte da Calif, OpenAI e altri ricercatori

Il piano: divulgazione coordinata

- Scoperta sotto embargo, release prevista per il 22/10
- 16/10: Chainguard informata, aiuta a testare le fix
- 17/10: primi exploit in the wild (4 giorni prima)
- Attore mai identificato, IP da rete di anonimizzazione
- «Tutto secondo i piani» solo fino a 2 giorni prima

Il leak e la corsa

- 19/10: cominciano a circolare voci e leak
- 20/10: CVE ID in rete, score sbagliato (10.0 invece di 9.2)
- 20/10: video PoC pubblicato per errore → release anticipata
- 21/10 07:41: Chainguard rompe l'embargo (6h prima)
- 21/10 14:00: patch 8.2.35, 8.3.36, 8.4.27, 8.5.12

Cosa è andato storto

- Leak con info errate → panico, non chiarezza
- Embargo rotto: la fix in binario esposta a tutti
- Nessuna mitigazione completa: serviva solo aggiornare
- 23/10: terza RCE (CVE-2026-115560), ancora senza patch
- La lezione: disciplina nel patching e nella divulgazione
- Già ad APRILE si sapeva del problema ma la discussione sulla soluzione è stata un pelino lunga.
- I repository sono un elemento della catena di sicurezza che deve essere maggiormente curato.

E noi, cosa abbiamo fatto?

- In team, con colleghi nella stessa situazione penosa, da soli non si va molto lontano!
- A parte non dormire...
- Messe offline le VM più complicate
- Le meno complicate sistemate appena pronti i repository
- Molte applicazioni con semplici aggiornamenti si sono rotte, non è accettabile
- Sotto stress i repository sono stato un altro anello debole che ha rallentato la soluzione

Cosa abbiamo capito

- Troppe applicazioni vecchie, non gestibili
- Clienti che non colgono quanto sia difficile (o impossibile) oggi mantenere in sicurezza server e applicazioni abbandonati a se stessi
- Il mondo sta cambiando in fretta → IA
- L'IA sta stravolgendo la sicurezza, in modo irreversibile
- Dobbiamo usare IA per combattere IA (serio ?)

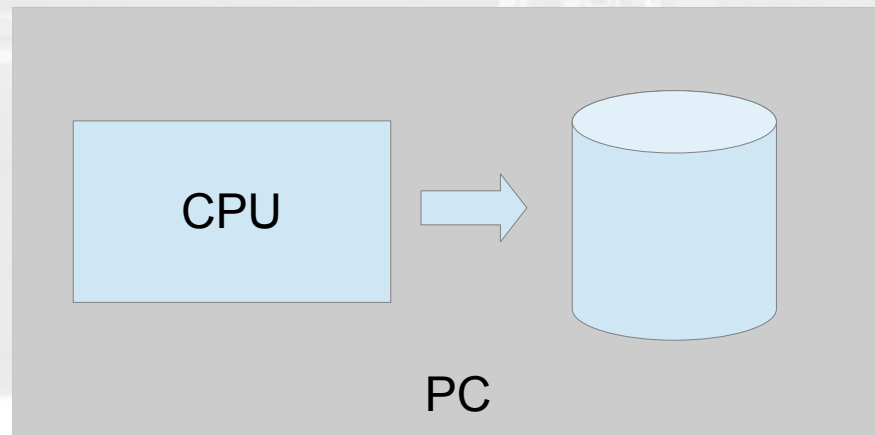
Di cosa parliamo

- La storia: dal bus SCSI al Fibre Channel
- iSCSI: ascesa e perché ha perso
- Velocità, latenza e costi
- Fibre Channel con Linux (pratica)
- Oltre lo SCSI: NVMe-oF
- Conclusione e domande

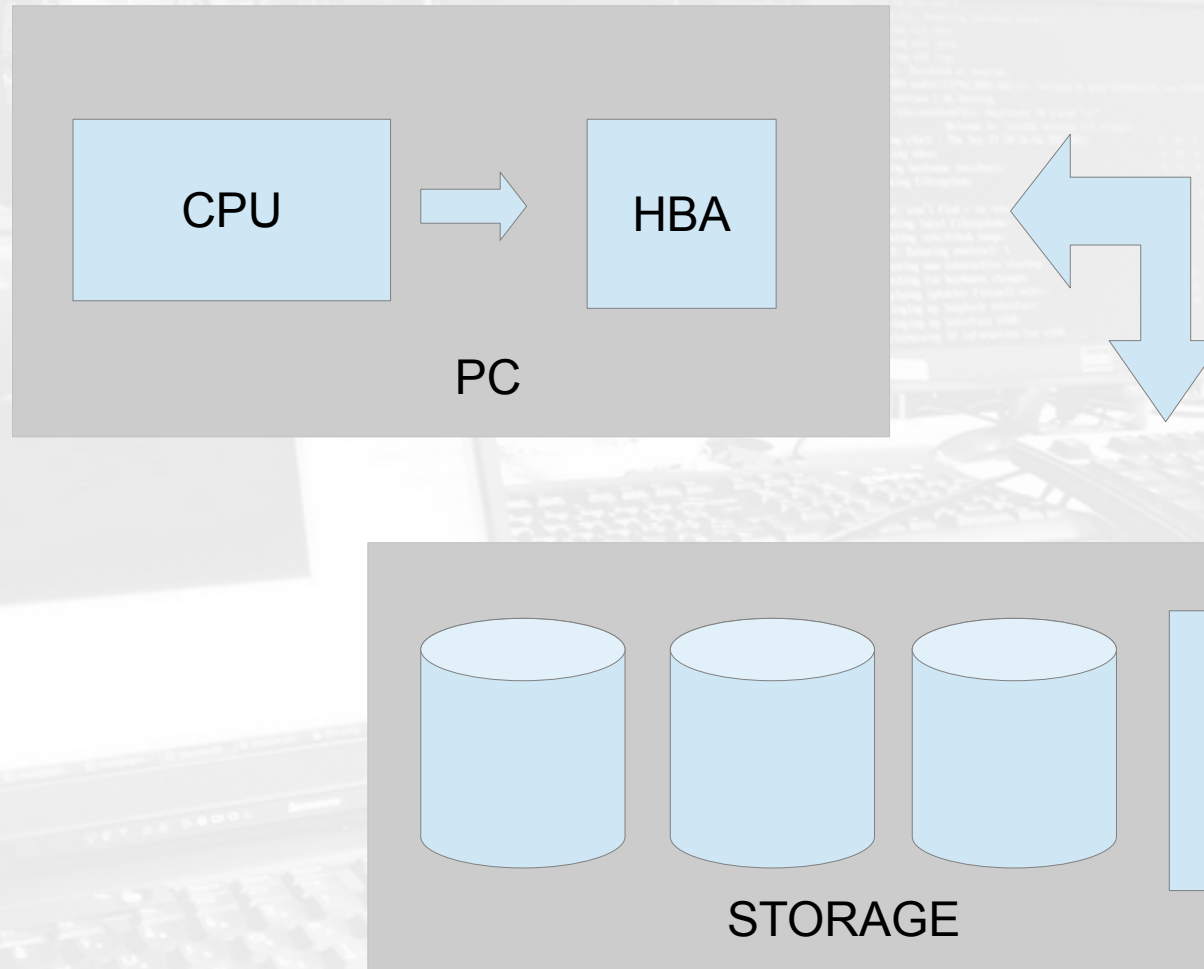
SAN Storage Area Network

- L'idea di base è di NON avere lo storage a bordo delle macchine che “producono”
- Le dimensioni (geografiche) non sono limitate
- Non esiste una tecnologia “universale”, il concetto e gli standard vengono dalla notte dei tempi, ma la tecnologia negli ultimi 40 anni (!) è cambiata molto rapidamente
- “allungare” i collegamenti tra la CPU e un dispositivo che deve essere veloce e a bassa latenza è una sfida molto impegnativa.

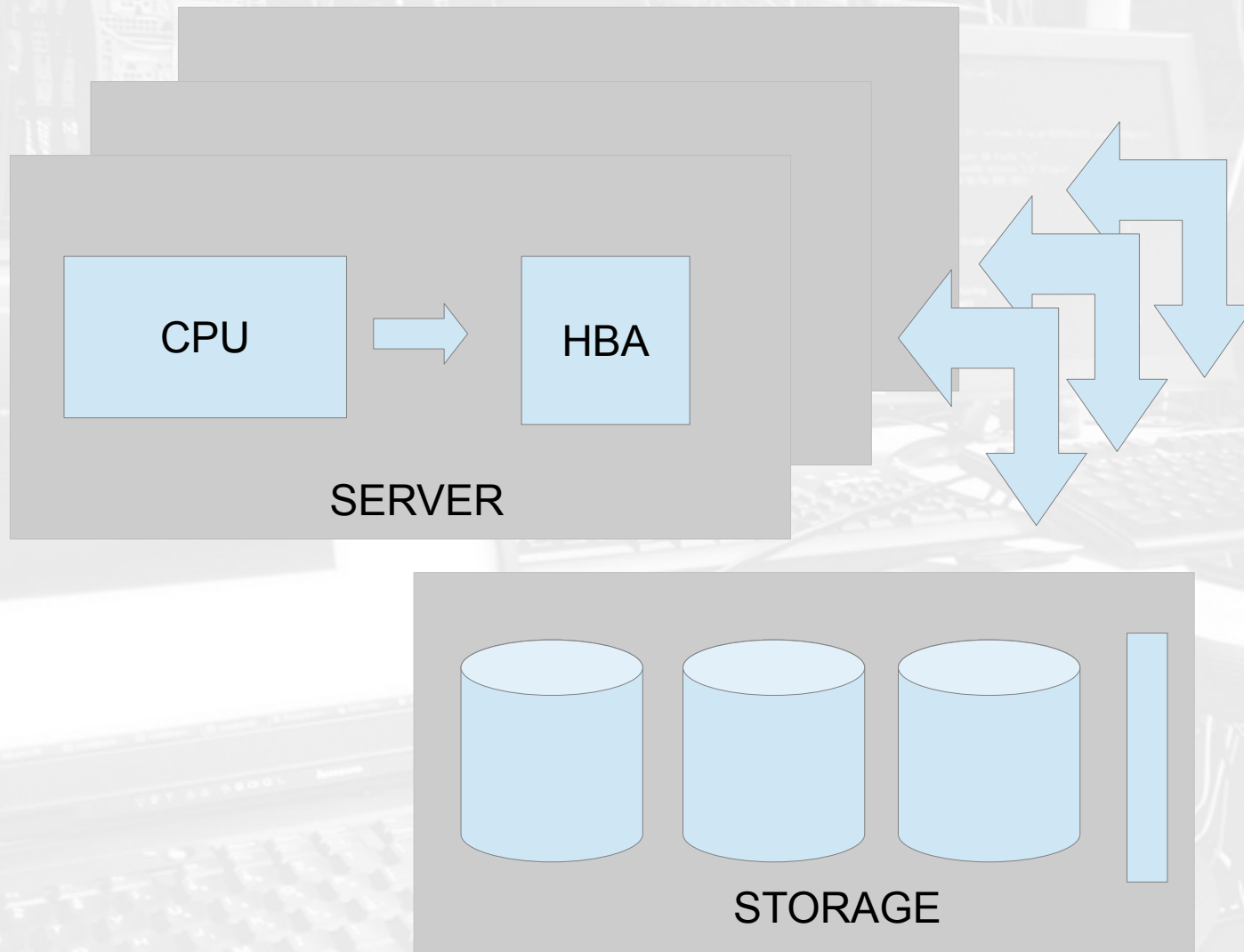
Partiamo dalla normalità



.. il disco fuori dal PC



... molti “server” uno storage

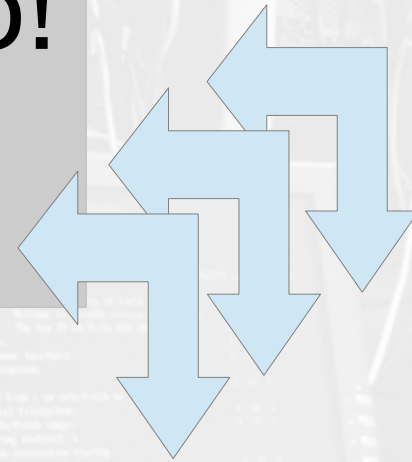


... molto tutto!

CPU

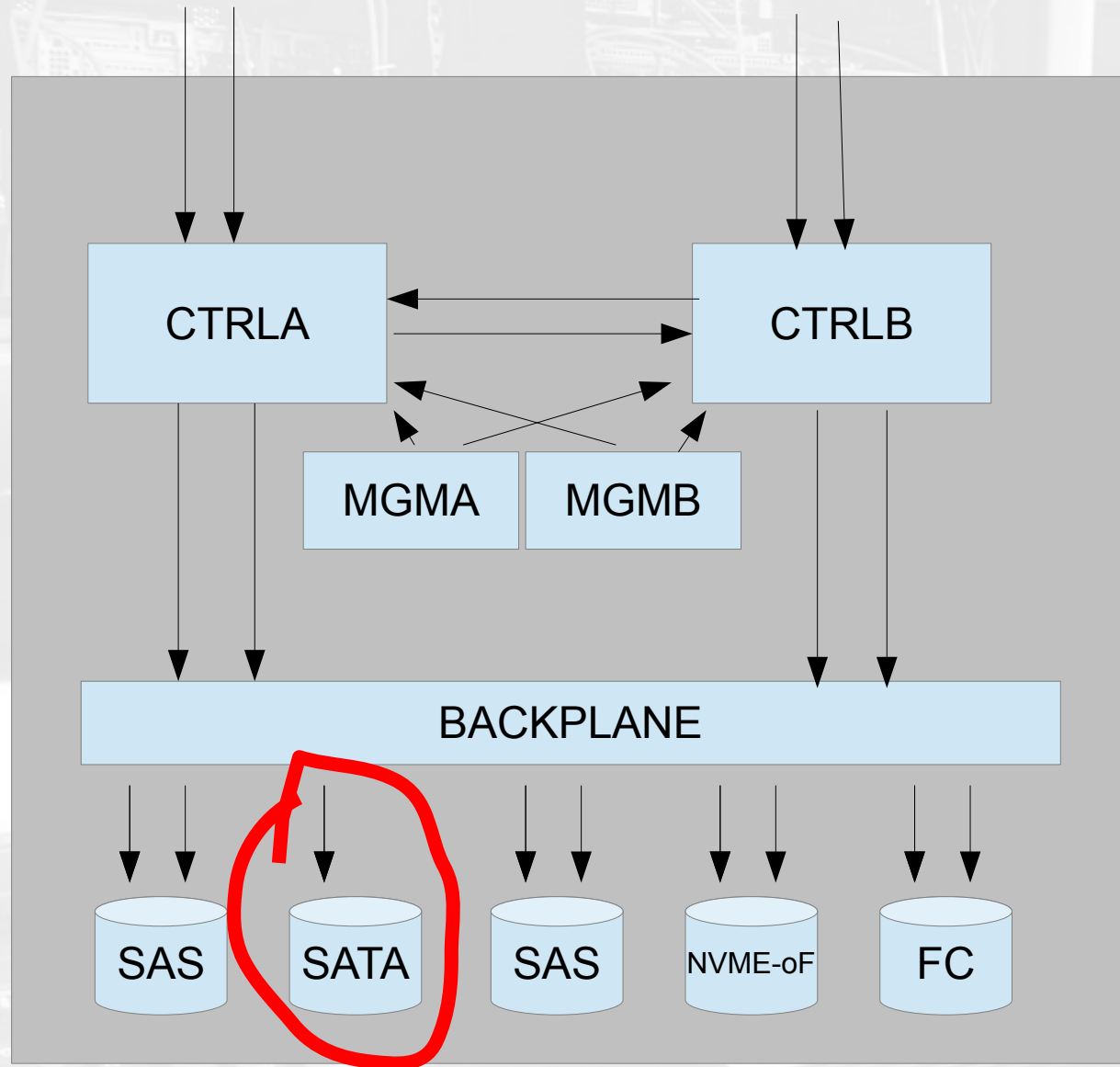
HBA

SERVER



STORAGE

Gli storage “seri”



una curiosità: quale filesystem?

- In caso di accesso “esclusivo” i classici FS vanno benissimo: ext4, NTFS, etc. etc.
- In caso di due device che accedono allo stesso dispositivo di storage servono filesystem particolari
 - Oracle Cluster File System (OCFS)
 - Red Hat Global File System (GFS2)
 - E altri (VMFS, CSV, QFS etc. etc.)
- Molto complessi... oltre a gestire le normali funzioni devono in qualche modo “coordinare” l'accesso ai file in modo da garantire la “coerenza” del dato

La storia...

- Il concetto di SAN nasce durante l'epoca dei primi grandi Mainframe, quando iniziano ad esserci più processori e più necessità di IO verso il disco, e, soprattutto, i dischi sono di grandi dimensioni, impossibile tenerli “vicino” alla CPU
- Nasce negli anni 80 lo standard HIPPI "High Performance Parallel Interface", la prima versione parallela era portata a max 25mt tramite 50 cavi doppi ritorti, era point to point e aveva caratteristiche molto innovative
- Lo standard HIPPI è il primo standard di comunicazione standard ANSI ad arrivare quasi al gigabit/secondo (0,8) nel 1999

gli anni 90 e il Fibre Channel

- Negli anni 90 nasce e si sviluppano due tecnologie che in breve tempo eclissano HIPPI, il gigabit ethernet e soprattutto Fibre Channel
- Nasce come Fiber Channel, poi viene sviluppato anche su rame e si preferisce il nome “British” Fibre per renderlo unico e distinguibile
- Il primo standard in realtà è del 1988, ma viene alla luce ufficialmente nel 1994, durante il periodo di “definizione” esplode la sua popolarità, e vengono sviluppati da diversi vendor (ad esempio SUN e IBM) soluzioni anche incompatibili tra di loro (ma chi avrebbe mai collegato allora uno storage SUN ad un mainframe IBM?)

Perché Fibre Channel

- Nasce per la fibra ottica, trasporta HIPPI, ESCON e SCSI, il segreto del suo successo è il fatto di essere seriale, nell'era dei cavi SCSI (50 fili) e HIPPI (100 fili) è un fattore non da poco
- Viene usato sia il cavo di rame che la fibra ottica multimodale, l'uso di questo standard favorisce la diffusione

ed ecco iSCSI

- Tutti pensano che la i di iSCSI sia “IP” invece sta per “Internet”
- L'idea è di portare lo standard SCSI “Small Computer Systems Interface” su reti di tipo TCP/IP
- Viene definito da IBM e CISCO dal 1998 e diventa una bozza di standard nel 2000
- Supporto nei sistemi operativi:
 - 2002 Solaris AIX NetBSD
 - 2003 Windows Netware HP-UX
 - 2005 Linux (kernel)
 - etc. etc.

iSCSI prende piede

- L'evoluzione della rete ethernet e della commutazione, con la nascita dei primi switch full wire speed, spingono verso la soluzione iSCSI
- Lo sviluppo delle reti IP a livello geografico, la crescita di Internet fanno la loro parte.
- La velocità di FC e ethernet cresce di pari passo ma la diffusione delle reti locali spinge verso il basso il costo degli apparati
- FC diventa sempre più di nicchia, iSCSI si espande e sembra diventare il sistema del futuro, ma...

il ritorno di FC

- Ethernet cresce rapidamente: 10G → 25/100G di serie, 400G nei fabric
- FC tiene il passo: 32GFC → 64GFC (Gen 7) diffuso, 128GFC (Gen 8) in arrivo
- iSCSI non viene abbandonato ma con le dovute limitazioni viene sempre più relegato all'ambito “geografico”
- nascono e si sviluppano i tentativi FCoE ovvero di passare i frame FC su connessioni di tipo ethernet, e altre cose molto creative come ATAoE e simili, destinati a fallire
- Il futuro prevede una coesistenza tra le soluzioni FC, quelle iSCSI e FCoE

Ma perché iSCSI ha “perso”

- Si appoggia su TCP/IP L4, gran parte del lavoro viene fatto da software
- Negli anni hanno iniziato ad implementare quello che si chiama “hardware offload” ovvero a portare in hardware le varie operazioni svolte dal software
- Il kernel viene coinvolto in modo troppo pesante
- Infatti nasce FCoE che resta di fatto a livello L2 (non è così ma si può dire..)
- FC è e resta più ottimizzato in hardware, ottimizzato per le funzionalità che deve svolgere

LA PILA OSI



quanto è veloce iSCSI?

- la capacità di trasporto di iSCSI è condizionata da una quantità di fattori
- velocità della connessione (10Gbit 25Gbit 100Gbit 400Gbit)
- dimensioni dell'MTU (Maximum Transmission Unit)

Le reti “normali” hanno MTU a 1500

Grazie proprio ad iSCSI nasce e viene sviluppato il concetto di “jumbo frame” ovvero MTU a 9000 che sono 6 volte tanto il normale

- Perché impatta così tanto nelle prestazioni? aumenta la dimensione del payload, riduce il numero di cambi di contesto per gestire i dati

quanto è veloce iSCSI?

- la teoria? 10Gbit 1250Mbyte/sec, 100Gbit 12500Mbyte/sec
- la pratica? con MTU 1500 ~1000Mbyte/sec, con jumbo frame (9000) fino a 1250Mbyte/sec
- il problema è il carico della CPU... per gestire il 10Gbit si può arrivare anche al 20/30% della CPU impegnata per iSCSI
- Lo stesso carico (più o meno) si crea dal lato opposto sullo storage
- a 100Gbit il carico CPU resta il collo di bottiglia: è qui che NVMe-oF fa la differenza

Due conti per fare chiarezza

- Immaginiamo di avere un file da 1TB
- 700M pacchetti da 1,5K
- 116M pacchetti da 9K
- Immaginiamo di fare “girare” in rete 700M pacchetti al secondo, anche 116M
- Il problema degli switch non è più la velocità ma il numero di pacchetti al secondo che sono in grado di gestire
- lo stesso problema c'è a livello di host, soprattutto se una parte del lavoro lo fa la CPU

e la latenza?

- a parità di dischi e al crescere della velocità la latenza è uno dei fattori che riduce la velocità dell'insieme
- al crescere della CPU soprattutto sugli host cresce la latenza dovuto al carico dei processori
- su iSCSI gigabit la latenza “indotta” può arrivare a 30/40ms
- su FC stessi dischi e stessi processori portano a meno del 50% di latenza, al limite della latenza dei dischi e dello storage

Ridondanza del percorso

- Uno dei fattori chiave nel mondo delle SAN è la ridondanza
- Sia con iSCSI che con FC l'implementazione della ridondanza è fatta tramite sistemi che operano a livello di blocco e non trasporto
- Si tende ad evitare qualsiasi meccanismo a livello di rete (LACP/BONDING) e a privilegiare la logica di gestione della presenza del disco e della connessione a livello di protocollo iSCSI o FC
- ALUA (Asymmetric Logical Unit Assignment)
- Su GNU/Linux Multipath!

hardware offload missing

- sarebbe la soluzione, ovvero avere hardware che ottimizza TCP/IP e “scarica” il lavoro della CPU
- ci stanno provando da anni ma le implementazioni sono lacunose, coprono una parte del processo di elaborazione, non tutto
- molto spesso compare la dicitura “se hai problemi disattiva l'hardware offload” perché nonostante tutto le implementazioni sono anche non perfette, e non sono facilmente adattabili
- esistono anche “schede” speciali per iSCSI... ma oltre al costo c'è un problema di compatibilità

vendor missing

- i grandi player del mercato (uso il plurale ma è uno solo) dopo aver investito molti soldi su iSCSI (e altre amenità) come tutti ha lasciato lì la questione
- i produttori di storage hanno a catalogo iSCSI ma non lo considerano strategico né lo sviluppano
- i produttori dal basso, quelli che fanno storage economici, pian piano vanno verso soluzioni FC... ma nel loro ambito iSCSI è ancora obiettivamente consigliato
- il mercato ha “ucciso” la concorrenza, i produttori di chipset e altri componenti sono due/tre in tutto il mondo. Se da un lato il prezzo sale, la qualità non ne risente

ma il costo?

- La larga diffusione della fibra ottica, sta di fatto abbattendo uno dei costi del FC
- FC a 16 Gbit → 2Gbyte al secondo ha dei costi per controller, switch, cavi, molto più contenuti di FC 32 e FC 64
- Il ridotto consumo di CPU negli host che fanno virtualizzazione impatta altrettanto, ogni 3 host con iSCSI ne serve uno in più per avere pari capacità di calcolo
- Non secondario: tutto l'hardware FC dismesso a favore di iSCSI sta ritornando nel mercato come usato

ma il costo?

- Uno storage Infortrend/HP con 6 SSD da 800GB e 6HDD da 8TB, tipico per l'uso aziendale medio, vale tra i 16 e i 18K euro in FC
- Lo stesso prodotto in iSCSI costa circa 3K in meno
- Ma un host per la virtualizzazione (ipotizzando 4 nodi) vale ben più di 3K
- La variabilità del carico delle CPU a seconda del carico dell'iSCSI inoltre rende la struttura molto più “impattata dai sovraccarichi di IO”

ma il costo?

- Gli switch FC sono scesi di prezzo in modo evidente, ad oggi con 6K euro si compera una coppia di apparati validi, e con 8K euro si costruisce l'infrastruttura in pieno ridondata.
- Gli switch 10Gbit “seri” non costano carissimi, ma tra cavi e schede di rete con hardware offload il rischio è di superare di gran lunga il budget degli 8K euro, senza contare la necessità di separare totalmente il traffico iSCSI dal resto, quindi servono comunque più switch e più schede di rete a bordo degli host.

il costo dei chip (e la carenza di SSD)

- La domanda AI assorbe la capacità produttiva NAND/flash
- NAND +55-60% QoQ nel Q1 2026; SSD enterprise +53-58%
- Costi ripesati 200-400% rispetto al 2024
- Un SSD enterprise da 30TB oggi costa ~18× un HDD equivalente
- >95% della produzione in 5 fabbricanti: nuovi impianti = anni e miliardi
- Carenza attesa fino al 2027: usare bene lo storage è d'obbligo

L'impatto energetico

- In termini strettamente energetici lo storage “condiviso” non è ottimale, almeno al crescere della struttura.
- Non è semplice da calcolare ma semplificando:
 - dato il 33% di consumo dello storage “a bordo” o “iscsi”, ogni tre host più o meno ne dobbiamo aggiungere uno
 - uno storage consuma più o meno come un host
 - da quattro host in su lo storage condiviso inizia a rendere dal punto di vista energetico

L'impatto economico

- Lo storage (e gli switch) costano... ma anche i dischi a bordo delle macchine
- Non semplice da valutare ma, senza valutazioni nel lungo periodo il costo dello storage distribuito tende ad andare in pari rispetto al costo degli host risparmiati dai 6 host in su
- Ma... se la vita dei nodi è indicativamente dai 3 ai 4 anni lo storage può vivere anche 10 anni
- L'investimento quindi rende, nel lungo termine, da 2 nodi in su!

Cosa cambia se passiamo a FC?

- Può essere un ottimo passaggio per ottimizzare le prestazioni, pensiamo al 30% perso con altri metodi
- Su una struttura nativa con dischi a bordo significa perdere il singolo punto di rottura
- La struttura può essere evoluta utilizzando questo dispositivo come punto di passaggio, senza impattare le prestazioni
- Ridurre gli host di circa un terzo ottimizzando i consumi

ma sotto GNU/Linux?

- il supporto come “client” è nativo, tutte le schede FC permettono di collegarsi ad uno storage
- multipath per gestire i percorsi multipli verso i device
- il supporto lato “server” è limitato a due tipologie di schede:
 - QLOGIC/EMULEX
- alcuni supporti ufficiali tra cui IET, SCST e LIO
- in alcuni esiste anche il supporto per Infiniband e FCoE

I target iSCSI su Linux: IET, SCST, LIO

- IET (iSCSI Enterprise Target): abbandonato dal 2010
- SCST: stabile ma fuori dal kernel, config a file
- LIO: nel kernel, supporta FC, iSCSI e NVMe-oF
- Si configura con targetcli, basato su sysfs

Fibre Channel: i protagonisti

- WWNN identifica il nodo, WWPN la porta
- HBA: adattatore host (QLogic, Broadcom/Emulex)
- N_Port host/storage, F_Port switch, E_Port ISL
- Velocità: 32, 64, 128 GFC

Entrare nel fabric: i login

- FLOGI: la porta si registra sul fabric
- PLOGI: due porte stabiliscono la sessione
- PRLI: negoziazione del protocollo (SCSI o NVMe)
- Solo dopo i login i LUN diventano visibili

Topologie del fabric

- Point-to-point: host → storage diretto
- Arbitrated loop: storico, ormai obsoleto
- Switched fabric: lo standard enterprise
- ISL (inter-switch link) per fabric ridondanti

Zoning e LUN masking

- Zoning: chi può parlare con chi (lato switch)
- Soft zoning per WWPN, hard zoning per porta
- LUN masking: quali LUN vede l'initiator (lato array)
- Servono entrambi: isolamento + autorizzazione

Linux: HBA e scoperta dei LUN

- `lspci | grep -i fibre`
- `systool -c fc_host -v`
- `cat /sys/class/fc_host/host*/port_state`
- `lsscsi / sg_inq` per interrogare i device
- `echo '- - -' > /sys/class/scsi_host/hostX/scan`

Multipath e tuning

- DM-Multipath: ridondanza e bilanciamento
- multipath -ll per verificare i path
- Queue depth: HBA, device e array d'accordo
- I/O scheduler none/mq-deadline su SAN
- IRQ affinity sulle CPU del NUMA node

SCSI vs NVMe: il cambio di paradigma

- SCSI: una coda, un comando alla volta
- NVMe: migliaia di code, una per core
- SCSI: overhead di traduzione tra i layer
- NVMe: comandi nativi, meno CPU e interrupt
- Da migliaia a milioni di IOPS

Oltre lo SCSI: NVMe

- SCSI: un comando alla volta, un collo di bottiglia
- NVMe nasce per gli SSD: code per core, non una coda sola
- Migliaia di code, ognuna profonda migliaia di comandi
- Nessuna traduzione: il device parla NVMe nativo
- Latenze dell'ordine dei microsecondi

NVMe-oF: i trasporti

- FC-NVMe: NVMe sopra il fabric FC esistente
- NVMe/TCP: Ethernet standard, il più semplice da adottare
- NVMe/RoCEv2: RDMA su Ethernet lossless (PFC, ECN)
- NVMe/InfiniBand: nicchie HPC
- Stesso set di comandi, trasporti diversi

FC-NVMe: il fabric continua

- Riutilizza switch, HBA e zoning già in campo
- Il login PRLI negozia NVMe invece di SCSI
- Discovery tramite NQN, non più solo WWPN
- Transizione graduale da SCSI a NVMe sullo stesso fabric

NVMe-oF su Linux

- `nvme list` e `nvme list-subsys`
- `nvme discover -t tcp -a IP -s PORTA`
- `nvme connect -t tcp -n NQN -a IP -s PORTA`
- FC-NVMe: dipende da driver HBA e supporto array
- `nvme-cli` unifica tutti i trasporti

Conclusione

- iSCSI, mi piace molto l'affermazione non mia “implementazioni di iSCSI su reti gigabit sono da considerare al massimo test di pre produzione”
- FC, complesso sì ma non troppo, per le prestazioni che garantisce va ripescato dal cilindro
- Usare il metodo di connessione giusto nel posto giusto e non scelte solo di tipo “economico” e “pratico”
- E oltre lo SCSI: NVMe-oF è il prossimo passo, sullo stesso fabric



Un po' di pratica!



DOMANDE?



per contattarmi:

maxnuv@linux.it