SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria

Corso di laurea in Ingegneria delle Telecomunicazioni

Tesi di laurea specialistica

Analisi e ottimizzazione della rete InfiniBand del sistema HPC CRESCO di ENEA

Relatore:

Prof. Marco LISTANTI

Correlatore:

Dott. Giovanni BRACCO

Candidato:

Andrea PETRICCA

Anno accademico 2008-2009

Ringraziamenti

Al Dott. Giovanni Bracco va
il mio doveroso e sentito ringraziamento, per la disponibilità,
per i suoi insegnamenti
e per i consigli utili che mi sono stati dati.
Al Dott. Silvio Migliori, al Dott. Antonio Perozziello e al Dott. Andrea Santoro
per avermi accompagnato e seguito in questa esperienza.

A tutti i miei parenti più cari che mi hanno sostenuto con tutto il loro amore
In particolare alle mie zie Silvia, Cinzia e Carla
uniche per l'aiuto morale che hanno saputo trasmettermi.

A mia nonna Edda, la mia seconda mamma
l'abbraccio più dolce che esiste

A mia madre e a mio padre
dove le parole non bastano, dove le paure diventano sorrisi
dove le difficoltà si trasformano in puro coraggio, dove
un esame difficile si supera ad occhi chiusi
dove basta uno sguardo per capirsi
dove l'amore è infinito.
A mia madre e a mio padre
Manuela e Giovanni
semplicemente
grazie.

Andrea

Ahahah!!! Credevate che avessi finito???

Come potevo dimenticarmi delle persone che “veramente” hanno contribuito alla fine di questa divertente agonia…i miei AMICI. Oh, mi viene già da piange.

Ciao Givvi, non basterebbe una tesi intera per descrivere quanto sei stato importante per me; non sei stato solo un amico, ma un fratello. Riusciamo a vedere, pensare e dire la stessa cosa nello stesso momento, ti porterò dentro di me fino a quando non guarderai i fiori dalle radici e ti verrò disegnare i baffi (e le basette) sulla foto. Insostituibile.

Per tutti Marco, per me Bati. La nostra storia d’amore è iniziata con delle lettere scritte in un periodo dove anche un semplice bacio era una conquista. La musica poi ci ha unito e fatto crescere, lasciando ricordi indimenticabili. La magggica Roma ci ha fatto sognare, incazzare, piangere, gridare, discutere, abbracciare, scavalcare ed emozionare.

L’importante è stato farlo insieme. Grazie Marco.

Ora è il turno di una persona che in modo silente e intenso dimostra quotidianamente quanto mi vuole bene. Vale, tra di noi le parole non servono, con uno sguardo riusciamo a dire tutto, e anche se sei della Lazie…rimani il mio cugino preferito.

Lo sostengo da molto tempo, e negli anni ho avuto la continua conferma. Ho avuto la fortuna di avere al mio fianco “il gruppo universitario” più bello del mondo.

Ed è a voi tutti che dedico il lavoro di questa tesi.

Non sarei mai riuscito a finire questo percorso senza ognuno di voi. Grazie a tutti.

Marco, Luco, Melo, Achille, Rachele, Christian, Willy, Faber, Valentocchia, Fabiola, Valerio coccia, mister Aldo, Lorenzo, Claudio, il presidente Aldo, Nino, Adriano e tutti quelli che in questo momento per stanchezza(sono le 3 di notte) non ho scritto, ma che hanno il permesso di aggiungere a penna il proprio nome su ogni copia (lascio lo spazio apposta).

…

Grazie ragazzi, non dimenticherò mai quello che siete stati per me.

Andrea

Ahahaha…ci siete cascati ancora??? La prossima tesi la faccio solo di ringraziamenti!!!

Diciamo che manca solo una persona.
Una persona che ho conosciuto in questi ultimi 5 mesi.
Un collega, adepto insieme a me del buon Filini, malato immaginario,
compagno di pausa caffè-sigaretta, confidente di emozioni,
rivale in amore di Bubba…ma soprattutto un nuovo amico.
È stato un colpo di fulmine tra di noi, simpatia da subito
stima affetto e complicità poco dopo.
Alessio Rocchi, per me Ale o cazzone, dire che sei stato fondamentale è riduttivo.
Ho la seria impressione che la nostra amicizia durerà per molto tempo.
Grazie Dott. Ing. Comm. Avv. Sen. Sir. Ale Rocchi.

Si ho finito, ed è anche ora.
Le persone care non nominate non si devono sentire escluse
dai ringraziamenti di questa tesi.
Sarebbe stato riduttivo ringraziarle solo per avermi aiutato in questi anni universitari.
Vi porto nel cuore a prescindere da tutto.

Andrea

Sommario

Indice delle figure e delle tavole

Introduzione

Questo elaborato rappresenta la conclusione di un lavoro di tesi specialistica svolta all'interno del centro di ricerca ENEA[1] di Portici nell'ambito del progetto CRESCO (Centro di RicErca computazionale sui Sistemi COmplessi)[2].

Con High Performance Computing (HPC) s'intendono tutte le tecnologie hardware e software che permettono la risoluzione di problemi che necessitano di ingenti risorse computazionali e di capacità di gestire grandi quantità di dati. Le sue applicazioni spaziano dalla ricerca scientifica all'industria: dall'automotive all'analisi finanziaria, fino alla biochimica e all'aerospazio, i sistemi di calcolo ad alta prestazione ricoprono un'importanza strategica assolutamente trasversale.
All'interno dei sistemi HPC ricoprono un ruolo sempre più importante le tecnologie d'interconnessione. Le prestazioni complessive sono legate non solo alla velocità di computazione dei singoli nodi, ma anche alla quantità d'informazioni che i nodi riescono a trasmettere per unità di tempo e al tempo impiegato da un messaggio per arrivare al nodo di destinazione, quindi l'alta velocità e la bassa latenza di trasferimento dati concorrono al raggiungimento di alte prestazioni nello svolgere applicazioni HPC.
La tecnologia di rete Infiniband[3] applicata a un sistema HPC ha una complessità elevatissima a livello di hardware e di software utilizzati. Una descrizione in dettaglio di protocolli, architetture ed elementi fisici sono da considerarsi oltre gli scopi della tesi. Sarà fornita in seguito una descrizione globale dell'architettura utilizzata e informazioni particolareggiate saranno fornite solo quando la trattazione degli argomenti lo renderà indispensabile.
Lo studio è stato condotto tanto su problematiche architetturali quanto su quelle che si riferiscono all'interconnessione fisica degli apparati all'interno dell'infrastruttura di calcolo CRESCO di ENEA, con lo scopo di compiere una analisi ad ampio spettro dell'intera rete e di formulare considerazioni ed ipotesi di *assessment* riguardo ai fattori evolutivi e di miglioramento delle performance globali.

Il primo capitolo presenta una panoramica sugli aspetti dell'HPC: saranno presentate le tecnologie più diffuse per implementarlo e s'introdurranno considerazioni in merito all'importanza degli investimenti *corporate* in questo ambito, al fine di ottimizzare e incrementare lo sviluppo aziendale.

Il secondo capitolo descrive InfiniBand, una particolare tecnologia molto usata in sistemi HPC, che ne permette non solo il funzionamento, ma ha anche il merito di incrementare le prestazioni di calcolo distribuito proprie del sistema stesso. Nella medesima sede si parlerà diffusamente della struttura e delle caratteristiche della rete analizzata (CRESCO).
Inoltre, si descriverà l'ambiente del sistema HPC CRESCO, i dispositivi che lo compongono, l'architettura di rete e gli aspetti software e prestazionali. Si spiegherà il funzionamento dei vari livelli che compongono il sistema, con un occhio di riguardo verso la parte switch che è la chiave delle prestazioni di un sistema InfiniBand.

Il terzo capitolo descrive le analisi e i tentativi di ottimizzare una rete di per sé molto complessa, per mezzo anche di strumenti automatici di monitoring (uno dei quali realizzato ad-hoc) atti a rilevare i flussi dati e gli errori che occorrono sul sistema.
Nel capitolo troveranno menzione anche le ipotesi e le intuizioni formulate ma ancora non validate da test opportuni (e che sono pianificate per la sperimentazione al sopraggiungere di opportune condizioni al contorno).
Va sottolineato, a questo proposito, che alcuni dei problemi identificati sono in corso di valutazione da parte degli specialisti InfiniBand IBM e Cisco, che collaborano con ENEA per la gestione del sistema.
In questo contesto, è importante notare le difficoltà oggettive in cui s'imbattono coloro che cerchino di reperire informazioni riguardanti gli aspetti tecnici di applicazione dei vari dispositivi impiegati. CRESCO è un cluster ad alte prestazioni in cui la connettività è implementata tramite hardware Cisco; la manualistica disponibile non fornisce adeguata documentazione riguardo ai *factory defaults* e alle modalità con cui eventuali cambi di configurazione (ad es. nei *port settings*, nel routing, ecc.) possono essere svolti in modo da garantire conformità con le specifiche protocollari.

Come vedremo, questa carenza ha implicato la necessità di procedere per approssimazioni successive, nella verifica e validazione delle ipotesi sperimentali.

Capitolo 1. High Performance Computing

Dai primi anni ’90 le tecnologie di rete si sono ampiamente diffuse soprattutto in ambito accademico. Questo ha favorito, all’interno di istituzioni accademiche, ditte e istituti di ricerca, una graduale sostituzione degli ambienti computazionali basati su grandi *mainframe*[I], che avevano caratterizzato gli anni ’70, con stazioni di lavoro interconnesse ad alta velocità. Con l’avvento di questa distribuzione delle risorse computazionali è stato coniato il termine Cluster.

Il cluster risponde all’esigenza, di utenti e amministratori di sistema, di assemblare insieme più macchine per garantire prestazioni, capacità, disponibilità, scalabilità ad un buon rapporto prezzo/prestazioni ed allo stesso tempo ridurre il più possibile il costo di gestione di tale sistema distribuito. Di cluster si parla sin dalla ‘preistoria’ dell’informatica (sistemi VAX della Digital -1983-, sistemi IBM JES3 per mainframe 360/370) ma spesso se n’è parlato male, confondendo le problematiche hardware con quelle software.

Infatti, per cluster si può intendere un modello architetturale all’interno del quale si possono ritrovare tutti i paradigmi di parallelismo riconosciuti, ovvero una sorta di modello generale, da non mettere in contrapposizione con i vari modelli in esso contenuti.

L’*High Performance Computing* (HPC) comprende tutte le tecnologie hardware e software che permettono la risoluzione di problemi che necessitano di ingenti risorse computazionali e di capacità di gestire grandi quantità di dati. All’interno del capitolo si analizzano le varie applicazioni dell’HPC all’interno del mondo accademico, nella ricerca scientifica e nell’industria.

[I] Mainframe: è un computer di alto costo e complessità, usato principalmente per applicazioni critiche, come grandi quantità di dati da processare.

1.1 Applicazioni

Le applicazioni dell'HPC sono le più disparate e spaziano dai database all'elaborazione di immagini, dall'industria aereo-spaziale a quella farmaceutica, dall'elettronica alle telecomunicazioni, dai trasporti alla chimica, dai service provider alle assicurazioni.

1.1.1 Applicazioni nella ricerca scientifica

L'HPC ha un profondo impatto sul modo in cui si fa ricerca scientifica. In particolare esiste una nuova disciplina, la *computational science[6]*, che si pone a complemento delle metodologie tradizionali di ricerca, l'approccio teorico e quello sperimentale, attraverso la costruzione e la simulazione di modelli matematici. Il *data mining*, una serie di strumenti che permettono di analizzare le complesse relazioni all'interno di insiemi di dati, costituisce il collegamento tra la teoria e la sperimentazione. L'assimilazione dei dati lega la simulazione all'osservazione, mentre il calcolo a elevate prestazioni è il legame tra teoria e simulazione.

L'HPC fornisce la capacità di costruire modelli fisici, chimici e biologici sempre più realistici e complessi e di studiarne il comportamento più in profondità.

Le simulazioni al calcolatore permettono la comprensione di fenomeni troppo complessi da trattare per via analitica oppure troppo costosi o pericolosi per essere studiati per via sperimentale. Esperimenti che erano fatti nei laboratori o sul campo, ora vengono sostituiti da simulazioni numeriche. Per esempio, alcuni studi come l'integrità di depositi di scorie nucleari o i cambiamenti climatici globali comportano scale temporali che escludono la sperimentazione. La scienza computazionale è cruciale in molte aree della ricerca scientifica e la capacità di fornire simulazioni gioca un ruolo importante nella progettazione e nella produzione in molte industrie (figura 1).

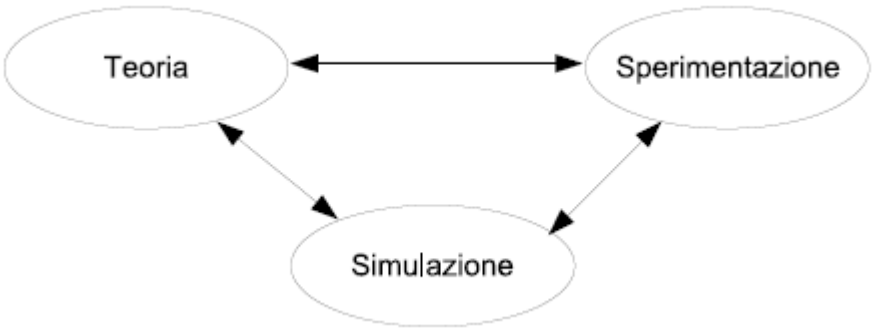

Figura 1 - Legame tra teoria, sperimentazione e simulazione

1.1.2 Applicazioni nell'industria

L'HPC è una tecnologia chiave che fornisce nuova competitività in aree che vanno dallo sviluppo dei prodotti, alla produzione vera e propria, al servizio clienti e al supporto. La costruzione di modelli per la simulazione dei progetti porta a un aumento della qualità dei prodotti e quindi un aumento della competitività. L'HPC permette di ridurre significativamente i costi e il time-to-market[II] nello sviluppo di nuovi prodotti. Ad esempio, il modello di un nuovo prodotto può essere usato per simularne il comportamento durante il processo di produzione. Questi strumenti permettono di migliorare la capacità produttiva mediante processi di ottimizzazione su larga scala, capaci di adattarsi in maniera rapida ed efficace a una situazione in evoluzione. L'HPC ha la capacità di introdurre nuove soluzioni, prodotti, processi dove prima questo non era possibile. Dall'elenco TOP500[7] (la lista dei 500 calcolatori più potenti al mondo) si nota come all'interno dell'industria, l'HPC, sia molto utilizzato nella modellizzazione e simulazione, nell'analisi, manipolazione e visualizzazione di grandi quantità di dati. Con l'HPC si possono studiare sistemi fisici o chimici estremamente complessi come ad esempio modelli meteorologici o di funzioni del corpo umano (come cuore o cervello). Inoltre l'HPC trova applicazione in campo finanziario e bancario, dove permette la creazione di previsioni finanziarie molto più accurate e affidabili.

Disponendo di maggiori risorse di calcolo e di memorizzazione è possibile eliminare quelle ipotesi semplificative necessarie per rendere il problema trattabile su sistemi di taglia inferiore.

[II] Time to market: tempo che intercorre dall'ideazione di un prodotto alla sua effettiva realizzazione.

L'HPC permette di elaborare immagini di sempre maggiore complessità in maniera più rapida ed efficiente. In particolare trova applicazione campo cinematografico e multimediale; inoltre rende possibile fare dei tour virtuali all'interno di edifici in via di progettazione. La meccanica computazionale consiste nello studio di materiali, fluidi e gas attraverso l'uso del calcolatore e comprende molte delle applicazioni dell'HPC nell'industria. Principalmente i calcolatori sono usati per lo studio di fluidi in movimento (fluidodinamica computazionale) o per lo studio di materiali sotto stress (analisi a elementi finiti).

1.2 Uso di cluster in HPC

Con il termine "Cluster" si intende quindi una particolare configurazione hardware e software avente per obiettivo quello di fornire all'utente un insieme di risorse computazionali estremamente potenti ed affidabili.
Con l'avvento dei servizi di rete internet e del web in particolare, i cluster sono cresciuti sempre più in popolarità, proponendosi come sistemi altamente scalabili ed efficienti.
Quando siti web o server che offrono servizi subiscono accessi da molti utenti, si può prevedere l'utilizzo di un cluster al fine di distribuire il carico complessivo sui diversi computer (i "nodi" del cluster) o per garantire la continuità del servizio a fronte dell'indisponibilità di uno dei nodi.
In generale questo tipo di organizzazione è assolutamente trasparente all'utente finale.
In alcuni ambienti, in cui l'esigenza di risorse computazionali è molto sentita, rivestono un ruolo molto importante i servizi di *Batch Queueing*, i quali consistono in sostanza in un gestore della distribuzione del carico di lavoro su un pool di macchine, anche eterogenee.
Il sistema rende quindi disponibile all'utente un insieme di code in grado di accettare applicazioni in esecuzioni in modo da consentire il bilanciamento del carico con l'esecuzione di job[III], sulla base di specifiche dettate dall'utente e dalla disponibilità dei diversi sistemi.
Il sistema decide inoltre quando si verificano le condizioni per eseguire il programma, e infine, al completamento dello stesso, si occupa della restituzione dei risultati sul richiesto flusso di output. Esistono dei prodotti evoluti che permettono il bilanciamento del carico

[III] Job: nome che indica la richiesta da parte di un utente di svolgere un determinato processo o calcolo.

all'interno dei pool di macchine, permettendo quindi il checkpoint dello stato di avanzamento della computazione e la migrazione da un nodo all'altro, qualora il nodo su cui viene lanciato il job dovesse essere spento.
La nuova frontiera va ben oltre questo approccio, fornendo dei meccanismi di migrazione automatica di processi e/o dati per garantire una maggiore omogeneità di distribuzione dei carichi di lavoro. Si può pertanto parlare propriamente di cluster quando un insieme di computer completi ed interconnessi ha le seguenti proprietà:

- i vari computer appaiono all'utente finale una singola risorsa computazionale;
- le varie componenti sono risorse dedicate al funzionamento dell'insieme.

1.2.1 Tipologie di Cluster

Possiamo trovare diverse tipologie di cluster, in base alle prestazioni richieste dal gestore e alle caratteristiche dei problemi analitici da risolvere.
Tra queste troviamo:

- High Performance Computing;
- High Availability;
- Load Balancing;
- Grid Computing.

Il primo, che sarà analizzato in questo elaborato, è un sistema che permette l'esecuzione distribuita di processi sui nodi del cluster. Processi che hanno la possibilità di dialogare tra di loro e quindi permettere anche il calcolo parallelo. Le connessioni tra i nodi sono di alta banda e bassa latenza come InfiniBand[3], Myrinet[4] e QsNet[5], trattate in base alle necessità nel 2° capitolo.
Un cluster High Availability garantisce una certa continuità dei servizi e dei sistemi, cercando di raggiungere il 100% di disponibilità.
La terza tipologia, che si avvale anche di caratteristiche di high availability è la Load Balancing, dove il sistema si preoccupa di bilanciare dinamicamente il carico sui vari server.

Ultima in ordine ma non per importanza è il Grid Computing, dove le risorse di molti computer, potenzialmente anche distribuiti in area geografica (WAN[IV] o Internet), collaborano al fine di risolvere problemi analitici di qualsiasi tipologia. Svantaggio principale della griglia computazionale è la bassa velocità con la quale i dispositivi collaborano tra loro.

Il sistema CRESCO può essere inquadrato nelle tipologie di HPC e di Grid-Computing, in quanto permette calcoli paralleli con altissima efficienza ed è la fonte di calcolo predominante di Grid-ENEA, ossia il sistema di Grid-Computing di ENEA.

1.3 Classificazione

Le architetture di HPC possono essere classificate sulla base di vari parametri [8]:

- Flusso di istruzioni e dati (Classificazione di Flynn)
- Architettura della memoria
- Interconnessione

1.3.1 Classificazione di Flynn

La classificazione di Flynn si basa sulla presenza di flussi singoli o multipli di dati e istruzioni. Le combinazioni possibili sono 4: SISD, SIMD, MISD, MIMD.

SISD (Single Istruction Single Data)

In questa categoria ricade l'architettura di Von Neumann[V][9]. L'unità di controllo preleva un'istruzione alla volta dalla memoria, la fornisce al processore che la esegue insieme al dato cui l'istruzione fa riferimento e scrive il risultato ottenuto in memoria. Il registro *Program Counter* viene aggiornato puntando all'istruzione successiva che deve essere prelevata per l'esecuzione (figura 2).

[IV] WAN: Wide Area Network, una rete informatica usata per connettere insieme più reti locali (LAN) in modo che un utente di una rete possa comunicare con utenti di un' altra rete.

[V] Architettura di Von Neumann: si riferisce allo schema di progettazione di calcolatori elettronici (CPU, unità di memoria, input, output e bus dati).

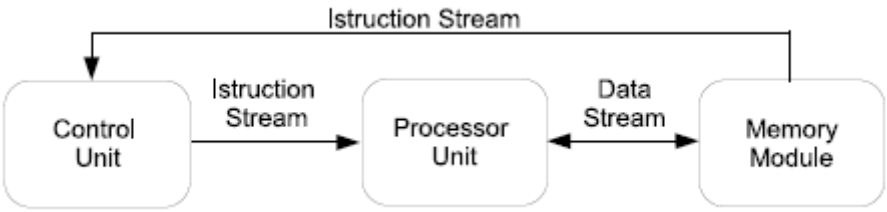

Figura 2 - Architettura SISD

I processori moderni dei Personal Computer non rientrano perfettamente in questa categoria, anche se sono comunemente inseriti in essa, in quanto usano tecniche derivate dal calcolo parallelo per aumentare le prestazioni, come ad esempio le pipeline.

SIMD (Single Istruction Multiple Data)

In questa categoria rientrano i calcolatori paralleli detti *Array Processor*. Sono costituiti da un grande numero di elementi di elaborazione (dell'ordine di centinaia o migliaia) molto semplici. Ciascun elemento ha una memoria locale e comunica con gli altri elementi mediante particolari istruzioni (figura 3).

Le istruzioni vengono eseguite su flussi di dati differenti. Un'unica unità di controllo preleva un'istruzione e la fornisce a tutti gli elementi di elaborazione. Questi ultimi prelevano i dati a cui l'istruzione fa riferimento, la eseguono e scrivono il risultato in memoria. Le unità di elaborazione lavorano in maniera sincrona condividendo un unico segnale di clock ed eseguendo contemporaneamente le stesse istruzioni. Questo tipo di architettura è utilizzata per problemi caratterizzati da un elevata regolarità, come ad esempio l'elaborazione di immagini.

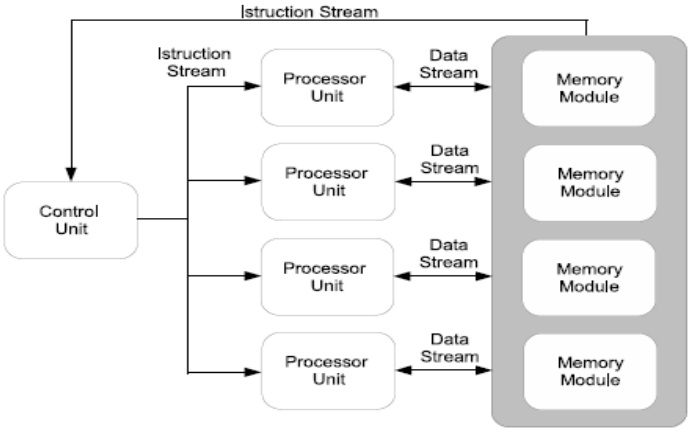

Figura 3 - Architettura SIMD

MISD (Multiple Istruction Single Data)

In tale architettura, istruzioni diverse vengono eseguite sullo stesso insieme di dati.

A seconda di come si interpreta questa definizione esistono calcolatori appartenenti a questa categoria oppure no. Se si considera MISD un calcolatore in cui distinte unità di elaborazione eseguono operazioni distinte sugli stessi dati, allora non esistono esempi di questa architettura (figura 4). Se invece si considera SIMD un calcolatore in cui il flusso di dati attraversa una serie di unità di elaborazione, allora ricadono in questa categoria *Systolic Array* e le architetture con pipeline.

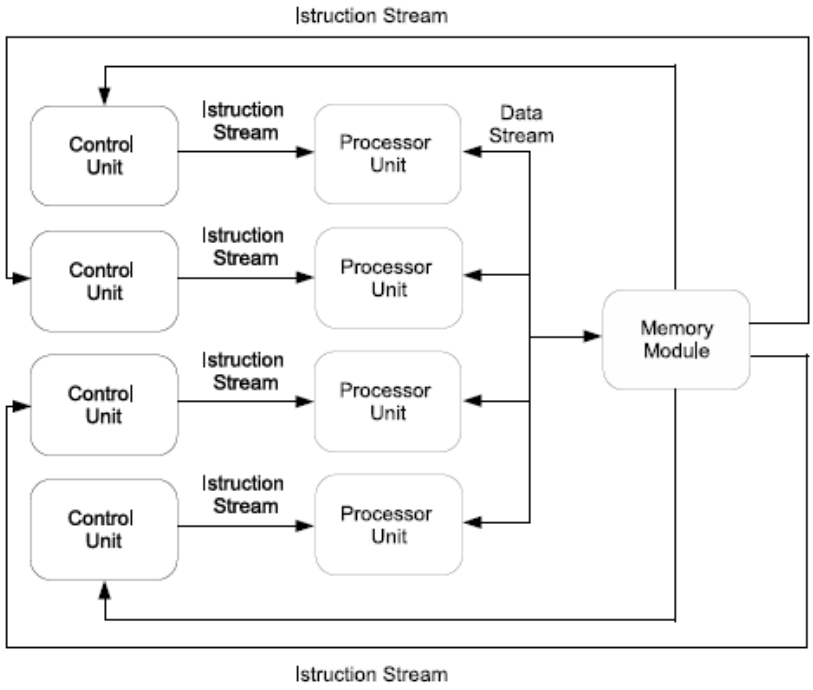

Figura 4 - Architettura MISD

MIMD (Multiple Istruction Multiple Data)

In questo tipo di calcolatore più unità di elaborazione eseguono istruzioni distinte su flussi di dati indipendenti (figura 5). Spesso ciascuna unità lavora su una parte del problema da risolvere in modo da ridurre i tempi di esecuzione. La sincronizzazione avviene mediante hardware specializzato oppure via software. In questa classificazione ricadono la maggior parte dei calcolatori usati per l'HPC.

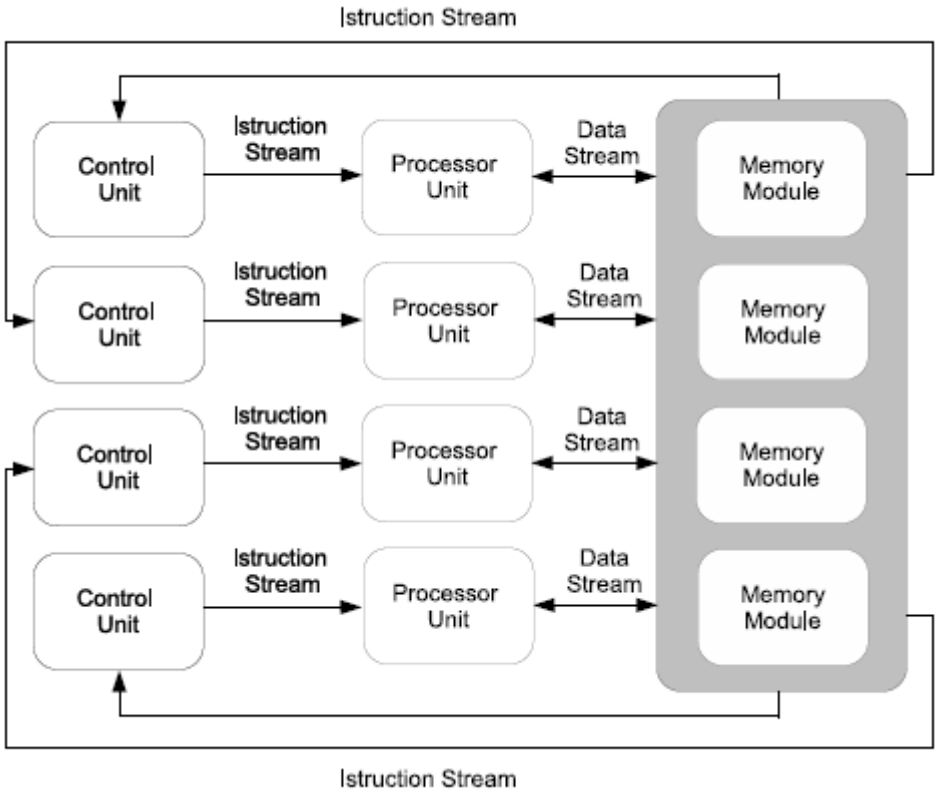

Figura 5 - Architettura MIMD

1.3.2 Classificazione per architettura della memoria

Questa classificazione suddivide i calcolatori in sistemi a memoria condivisa (*Shared memory systems*) e sistemi a memoria distribuita (*Distributed memory systems*) (figura 6).

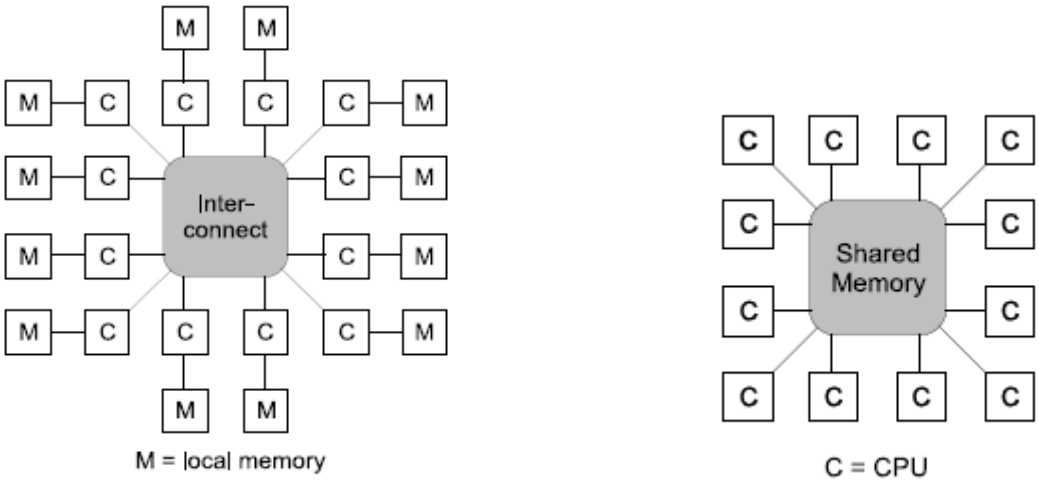

Figura 6 - Sistema a memoria condivisa e sistema a memoria distribuita

Nei sistemi a memoria condivisa, più CPU condividono lo stesso spazio di indirizzamento. Questo significa che non è necessario sapere dove sono memorizzati i dati, in quanto tutte le CPU possono accedere a qualunque locazione di memoria allo stesso modo.

I sistemi a memoria condivisa possono essere SIMD o MIMD. La comunicazione e la sincronizzazione avviene implicitamente attraverso variabili condivise.

Meccanismi di sincronizzazione come i semafori sono usati per regolare l'accesso in scrittura delle unità di elaborazione alle stesse locazioni di memoria. I vantaggi di questo tipo di architettura sono la flessibilità, la semplicità di programmazione e la facilità con cui possono essere adattati sistemi operativi esistenti per architetture SISD. La condivisione dei dati tra le unità di elaborazione è semplice e veloce.

Gli svantaggi sono la necessità di sincronizzazione, la limitata scalabilità (l'aumento del numero delle CPU porta a contese nell'accesso alla memoria), l'aumento progressivo dei costi a causa di hardware specializzato. L'affidabilità è ridotta, infatti se una CPU o un modulo di memoria si guasta questo può portare al blocco dell'intero sistema.

Nei sistemi a memoria distribuita ciascuna CPU è dotata di propria memoria.

Le CPU sono connesse tra loro e possono scambiare dati tra le rispettive memorie.
In tale architettura è necessario sapere dove si trovano i dati, in quanto solo la CPU relativa ad un blocco di memoria può accedervi direttamente. Anche questi sistemi possono essere SIMD o MIMD. I vantaggi sono la scalabilità rispetto al numero di CPU e alla dimensione della memoria, i costi ridotti e l'elevata affidabilità in quanto un guasto ad una CPU non blocca l'intero sistema. Gli svantaggi sono la difficoltà di programmazione (tramite scambio di messaggi), la necessità di decomporre le strutture dati e tempi di accesso alla memoria non uniformi.

CRESCO è un cluster di tipo HPC formato da un insieme di nodi a memoria condivisa e multi-core, interconnessi tra di loro tramite una rete a bassa latenza InfiniBand.

1.3.3 Classificazione per tipo di interconnessione

Le interconnessioni tra le unità di elaborazione costituiscono un elemento fondamentale dei calcolatori e ne determinano le prestazioni complessive. Per questo motivo, dedicheremo il prossimo capitolo per analizzare la tecnologia Infiniband che permette l'interconnessione tra i dispositivi di memoria e di calcolo del sistema CRESCO in particolare.

1.4 Analisi delle prestazioni

La legge di Amdahl [10] viene usata per trovare il miglioramento atteso massimo in una architettura di calcolatori o in un sistema informatico quando vengono migliorate solo alcune parti del sistema, la cui forma più generale può essere espressa dalla frase:
"*Il miglioramento che si può ottenere su una certa parte del sistema è limitato dalla frazione di tempo in cui tale attività ha luogo*".
Aumentando le risorse a disposizione, ad esempio raddoppiando il numero di unità di elaborazione, ci si aspetta un aumento analogo nelle prestazioni. Questo è possibile, ma non è quello che accade normalmente, a causa di una serie di inefficienze introdotte dall'implementazione parallela dell'algoritmo, in particolare dovuto alla comunicazione

tra i processi, all'inattività di alcune unità di elaborazione e alla maggiore complessità computazionale rispetto alla versione sequenziale.

A parte alcuni casi particolari (detti *embarrassingly parallel*), le unità di elaborazione devono comunicare tra loro, ad esempio per lo scambio dei risultati intermedi. Il tempo impiegato nella comunicazione è una delle maggiori fonti di inefficienza nel calcolo parallelo.

Le unità di elaborazione possono essere inattive a causa di scarso bilanciamento del carico di lavoro, dell'attesa di sincronizzazione con le altre unità oppure a causa di codice sequenziale che deve essere eseguito solo su una unità. Se il miglior algoritmo sequenziale è difficilmente parallelizzabile oppure se non è possibile farlo, allora se ne utilizza uno con prestazioni minori, ma che può essere facilmente ripartito su più unità di elaborazione, a scapito di una maggiore complessità. L'analisi delle prestazioni è importante per determinare il migliore tra vari algoritmi, la velocità e l'efficienza di una piattaforma hardware e i benefici dovuti all'implementazione parallela. Le tre grandezze principalmente usate sono lo speedup, l'efficienza e il costo.

1.4.1 Speedup

Nella valutazione dei calcolatori paralleli si è interessati all'aumento di prestazioni che si può ottenere con l'implementazione parallela di un algoritmo. Lo speedup (assoluto) S è definito come il rapporto tra il tempo di esecuzione del miglior algoritmo sequenziale e il tempo di esecuzione dell'algoritmo parallelo su N identiche unità di elaborazione . Lo speedup relativo S_r è definito come il rapporto tra il tempo di esecuzione dell'algoritmo parallelo su una sola unità di elaborazione e il tempo di esecuzione su N identiche unità di elaborazione.

Ora che è stato definito lo speedup (relativo) è interessante stabilire qual' è il valore che si ottiene con un certo algoritmo e qual' è il valore massimo ottenibile.

Lo speedup è limitato dalla porzione di programma che deve essere eseguita sequenzialmente, che non è mai nulla, e dal numero di nodi che non può aumentare arbitrariamente a causa della scalabilità dell'hardware. Richiamando la legge di Amdahl possiamo ricavare il massimo aumento di prestazioni di un sistema. Se s è il tempo di esecuzione della parte di algoritmo eseguita sequenzialmente e p è il tempo di esecuzione della parte ese-

guita in parallelo (con s + p = 1 per semplicità), allora con N unità di elaborazione lo speedup relativo è

$$S_r = \frac{s+p}{s+\frac{p}{N}} = \frac{1}{s+(1-s)/N} \leq \frac{1}{s}$$

quindi il limite è dato dal reciproco della frazione sequenziale dell'algoritmo, valore ottenibile con un numero infinito di unità di elaborazione. In realtà si possono ottener speedup più elevati. Questo non è in contraddizione con la legge di Amdahl. Infatti in essa si assume implicitamente che la parte che può essere eseguita in modo parallelo p sia costante. Questo però non accade in pratica. Quando si aumenta il numero di unità di elaborazione, si aumenta anche la dimensione del problema in modo da sfruttare le nuove risorse disponibili. Per esempio, nel caso di un'integrazione numerica, si diminuisce il passo in integrazione, aumentando la complessità del problema. In generale, fissato un tempo di esecuzione ragionevole, si vuole massimizzare la dimensione del problema. Non è la dimensione del problema ad essere costante, ma il tempo di esecuzione. La parte sequenziale dell'algoritmo generalmente non aumenta all'aumentare della dimensione del problema, ma rimane costante, basti pensare al tempo di caricamento del programma. Se ora si suppone che la parte dell'algoritmo eseguita in parallelo aumenti linearmente con il numero di elementi di elaborazione, allora con una notazione analoga a quella introdotta in precedenza, si ha che il tempo di esecuzione dell'algoritmo su un'unica unità di elaborazione è $s' + p'N$. Lo speedup relativo è quindi:

$$S_r = \frac{s' + p'N}{s' + p'} = s' + p'N = N + (1-N)s'$$

Questo dimostra che è possibile ottenere speedup elevati anche con un grande numero di unità di elaborazione (dell'ordine delle migliaia).

Da un punto di vista teorico il massimo speedup assoluto ottenibile è pari a N, cioè uno speedup lineare. Infatti se il miglior algoritmo sequenziale impiega un tempo t_s per essere eseguito su una sola unità di elaborazione, allora uno speedup pari a N può essere ottenuto se nessuna unità di elaborazione impiega un tempo superiore a t_s/N. Se si suppone

possibile ottenere uno speedup superiore a N (speedup sopralineare), allora ciascuna unità di elaborazione impiega un tempo inferiore a t_s/N.

Se un'unica unità di elaborazione esegue il lavoro svolto dalle N unità, impiega un tempo inferiore a t_s, ma questo contraddice il fatto che il tempo t_s sia ottenuto con il miglior algoritmo sequenziale. In pratica speedup superiori ad N sono possibili quando il lavoro compiuto dalla versione sequenziale è maggiore di quello della versione parallela (per esempio nell'esplorazione di un albero, si può trovare la soluzione senza esplorarlo tutto), oppure a causa di accorgimenti hardware che avvantaggiano la versione parallela (per esempio, dividendo il problema su più unità, questo riesce ad essere contenuto nelle rispettive cache). Nelle relazioni e nelle considerazioni scritte sopra è stato tralasciato l'overhead dovuto alla comunicazione e alle funzioni del sistema operativo (creazione/distruzione processi, allocazione di memoria, ecc.)

1.4.2 Efficienza

L'efficienza è la frazione del tempo di esecuzione in cui l'unità di elaborazione viene utilizzata. Essa è definita come il rapporto tra lo speedup S_r ed il numero di unità N

$$E = \frac{S_r}{N}$$

Idealmente lo speedup è pari ad N e quindi l'efficienza `e 1. Nella realtà l'efficienza è compresa tra 0 e 1 a seconda di quanto vengono utilizzate le unità di elaborazione.

1.4.3 Costo

Si definisce costo dell'esecuzione di un'algoritmo su N unità di elaborazione, il prodotto tra il numero di unità di elaborazione e il tempo di esecuzione

$$C = N\, t_p$$

Esso rappresenta la somma dei tempi di esecuzione di ciascuna unità.

1.5 Metodo di misura delle prestazioni

La misura delle prestazioni avviene tramite applicazioni di riferimento (*benchmark*[VI]) che permettono di confrontare tra loro diverse architetture e di capire quale è la più appropriata per gli algoritmi che verranno utilizzati. Poiché solitamente sono disponibili i risultati dei benchmark di vari calcolatori, è possibile confrontare il proprio sistema con altri. I benchmark possono testare un sotto sistema specifico (memoria, I/O) oppure il calcolatore nel suo complesso. Scegliendo un benchmark che sia simile all'applicazione che verrà effettivamente usata sul calcolatore, si possono ottenere i risultati più significativi. Si descrivono brevemente i benchmark più utilizzati.

I moderni processori includono una *floating point unit* (FPU), componente specializzata nel calcolo delle operazioni in virgola mobile. Quindi il FLOPS[VII] è un'unità di misura delle prestazioni della FPU. Lo studio delle prestazioni di un calcolatore attraverso la misura dei FLOPS non fornisce indicazioni dettagliate sulle reali capacità di elaborazione della CPU poiché non sono considerati fattori quali il carico del microprocessore e il tipo di operazione in virgola mobile, quindi è bene specificare che il calcolo delle prestazioni attraverso il FLOPS può essere indicativo solo negli ambiti in cui il calcolo in virgola mobile sia la caratteristica predominante del calcolo stesso.

La valutazione dell'effettiva potenza di calcolo tramite FLOPS deve essere effettuata in relazione ad un benchmark standard, che consenta di comparare i valori ottenuti con quelli di altri elaboratori. Un riferimento in questo senso sono il Linpack e il Lapack[11], forse con la più nota versione portabile HPL[13] tra l'altro utilizzata per testare le prestazioni di CRESCO.

[VI] Benchmark: si intende un insieme di test software volti a fornire una misura delle prestazioni di un computer per quanto riguarda diverse operazioni.

[VII] FLOPS: Floating Point Operations Per Second, indica il numero di operazioni in virgola mobile eseguite in un secondo dalla CPU.

1.6 Sistemi Operativi tipici dei cluster HPC

Il sistema operativo di un sistema distribuito deve essere tale da consentire e favorire le operazione tipiche di un cluster. Allo stato attuale vengono individuate due tipologie di sistemi operativi utilizzati all'interno di cluster HPC:

- “Network Operating System”;
- “Distributed Operating System”.

La differenza principale tra queste due tipologie risiede principalmente nel tipo di visione del sistema che il sistema stesso è in grado di dare all'utente.
Nei sistemi denominati “Network Operating System” l'utente ha un percezione precisa dei nodi, delle risorse occupate e può scegliere dove avviare la propria elaborazione.
Nei sistemi denominati “Distributed Operating System” l'utente ha solo la percezione d'insieme e sarà il sistema stesso a determinare dove e come verrà eseguita la computazione.
Da un punto di vista progettuale i sistemi operativi di tipo “Distributed Operating System” hanno un livello di complessità decisamente più marcato rispetto ai “Network Operating System”.
Il sistema stesso, infatti, deve gestire problematiche legate ai guasti, alla ridondanza dei componenti critici, alla sicurezza ed alle performance, mantenendo livelli di perfomance e scalabilità elevati. Nei sistemi operativi di tipo “Network Operating System” è l'utente stesso, in particolare l'amministratore, che deve intervenire in caso di guasto di parte del sistema ed è altresì l'utente che è demandato a decidere su quale nodo avviare l'elaborazione delle informazioni, oltre al fatto che la maggior parte dei meccanismi che rendono il sistema operativo non sono celati dal sistema stesso e pertanto la complessità d'insieme risulta notevolmente ridotta.
Il sistema operativo scelto per CRESCO è LINUX RedHat EL 5.1[14] montato su tutti i nodi di calcolo.
Il sistema HCO CRESCO è inserito nella infrastruttura ENEA-GRID i cui componenti principali sono il gestore delle risorse LSF[15] e il file system distribuito OpenAFS[16] .

1.7 File System parallelo: GPFS e LUSTRE

In un sistema HCP è importante il ruolo di un file system parallelo, e due delle implementazioni più diffuse sono GPFS e LUSTRE.

GPFS (*General Parallel File Syste*m)[17] facilita il sistema ad avere alte performance permettendo ai dati di essere accessibili contemporaneamente da più processori. I già esistenti file system sono costruiti per un ambiente a singolo server, ed aggiungere più file server non migliora le prestazioni.

GPFS fornisce alte performance di I/O tramite lo "striping", ossia il distribuire l'informazione su più dischi. Permettendo la lettura e scrittura in parallelo su questi dischi, dove l'informazione è ripartita, si ha un incremento sostanziale delle prestazioni. Altre caratteristiche importanti di GPFS sono l'high avaliability, supporto a cluster eterogenei, disaster recovery[VIII] e sicurezza.

LUSTRE[18] è un file system di rete usato da sistemi cluster ad alte prestazioni. Il nome deriva da una fusione tra le parole Linux e cluster.

LUSTRE tende ad ottimizzare la scalabilità, la sicurezza, la robustezza a guasti e l'alta disponibilità di risorse. Creato e gestito dalla Sun Microsystems, Inc.

Il principale obiettivo è stato quello di creare un file system in grado di gestire decine di migliaia di nodi, storage con ordini di grandezza dei petabyte inserito in una infrastruttura sicura che arrivi ad un data rate di 100 GBps.

Nel sistema HPC CRESCO è utilizzato il file system parallelo GPFS e il traffico dati avviene sulla rete InfiniBand oggetto di questa tesi.

[VIII] Disaster recovery: si intende l'insieme di misure tecnologiche e processi organizzativi atti a ripristinare sistemi, dati e infrastrutture necessarie all'erogazione di servizi di business a fronte di gravi emergenze.

Capitolo 2. Interconnessioni InfiniBand su CRESCO

Nel corso degli ultimi anni l'evoluzione tecnologica e il conseguente aumento di prestazioni che ha caratterizzato i componenti di un personal computer hanno incrementato esponenzialmente la richiesta di banda passante tra i suoi diversi sottosistemi. Sono state proposte numerose alternative per superare i limiti dell'ormai limitato bus Pci e si è creata una certa confusione circa il posizionamento di queste nuove tecnologie e su come esse possano collaborare.

Un buon punto di partenza per capire e analizzare il presente è dare un'occhiata al passato: il bus Pci (*Peripheral Component Interconnect*) [20], introdotto nei primi anni Novanta, è stato originariamente concepito come interfaccia di connessione tra i chip presenti sulle schede madri ma si è presto evoluto costituendo prima un rimpiazzo per la tecnologia Isa (lo standard fino a quel momento dedicato alla gestione delle schede di espansione) e quindi il principale sistema di Input/Output su server e piattaforme di comunicazione. Il bus Pci nella sua prima versione opera alla frequenza di 33 MHz e comunica tramite un canale a 32 bit; queste due specifiche consentivano di raggiungere una banda passante di 133 MByte/s che doveva però essere condivisa da tutte le unità agganciate al bus. Nel corso degli anni la banda è stata portata a 533 MByte/s attraverso il raddoppio della frequenza di clock e dei bit trasmessi per ciclo: tali evoluzioni sono indirizzate principalmente ad ambienti server che richiedono un'elevata capacità di comunicazione.

Con l'ingresso nell'era di internet le richieste in termini di prestazioni, affidabilità e scalabilità avevano rapidamente trasformato il bus Pci in uno dei principali colli di bottiglia nei sistemi di comunicazione di oggi: i server devono rendere disponibili grandi quantità di dati a milioni di utenti non solo appartenenti alla rete locale ma anche connessi tramite Internet. Per esaudire queste richieste, i dipartimenti delle aziende, hanno cercato di concentrare i dati e le applicazioni su pochi server robusti e dotati di schede Gigabit Ethernet o moduli per i collegamenti in fibra ottica.

Le ultime evoluzione dello standard PCI, ossia con la versione PCI Express 2.0, portano l'ampiezza di banda a 20 GBps con connessione 16X.

La scelta fatta per interconnessioni di CRESCO è caduta su InfiniBand, che garantisce alta banda, bassa latenza e flessibilità, supportata da HCA con tecnologia PCI Express

2.1 Specifiche InfiniBand™ Architecture.

L'associazione di industrie denominata "*Infiniband Trade Association* (*IBTA*)"[3] ha messo mano alla definizione delle specifiche della omonima architettura di input/output, con l'obiettivo di sviluppare uno standard aperto per far fronte alle necessità di calcolo emergente a livello di impresa e fornire una soluzione atta a connettere e condividere dispositivi distribuiti in cluster, in reti storage o in reti IP.

L'associazione è stata fondata da Compaq, Dell, HP, IBM, Intel, Microsoft e Sun. A queste si sono successivamente aggiunte: 3Com, Agilent, Adaptec, Brocade, Cisco, Fujitsu-Siemens, Hitachi, Lucent, NEC e Nortel Networks.

Con la terminologia InfiniBand Architecture (IBA) intendiamo l'architettura che costituisce InfiniBand, ossia lo stack protocollare che caratterizza InfiniBand (IB). Le specifiche dell'architettura IB descrivono in primo luogo l'interagire dei dispositivi di calcolo (processor nodes) e dei dispositivi di interfaccia con l'esterno (I/O nodes) atti a formare una rete ad alta velocità il cui scopo principale è il trasferimento di dati sia tra sistemi di computer che elementi di storage. Nel caso in cui l'utilizzo per lo storage sia quello dominante la rete viene chiamata SAN[IX]. Il modo in cui si progetta una architettura InfiniBand è indipendente dal tipo di sistema operativo utilizzato e dalle piattaforme su cui si poggerà l'IBA. Vedremo nei capitoli successivi, per avere una concezione più generale dell'argomento, il tipo di piattaforma sulla quale si basa la rete InfiniBand studiata.

[IX] SAN: Storage Area Network: Una rete SAN consiste in un'infrastruttura di comunicazione, che fornisce connessioni fisiche e in un livello di gestione, che organizza connessioni, elementi di storage e sistemi di computer in modo da garantire un trasferimento di dati sicuro e robusto.

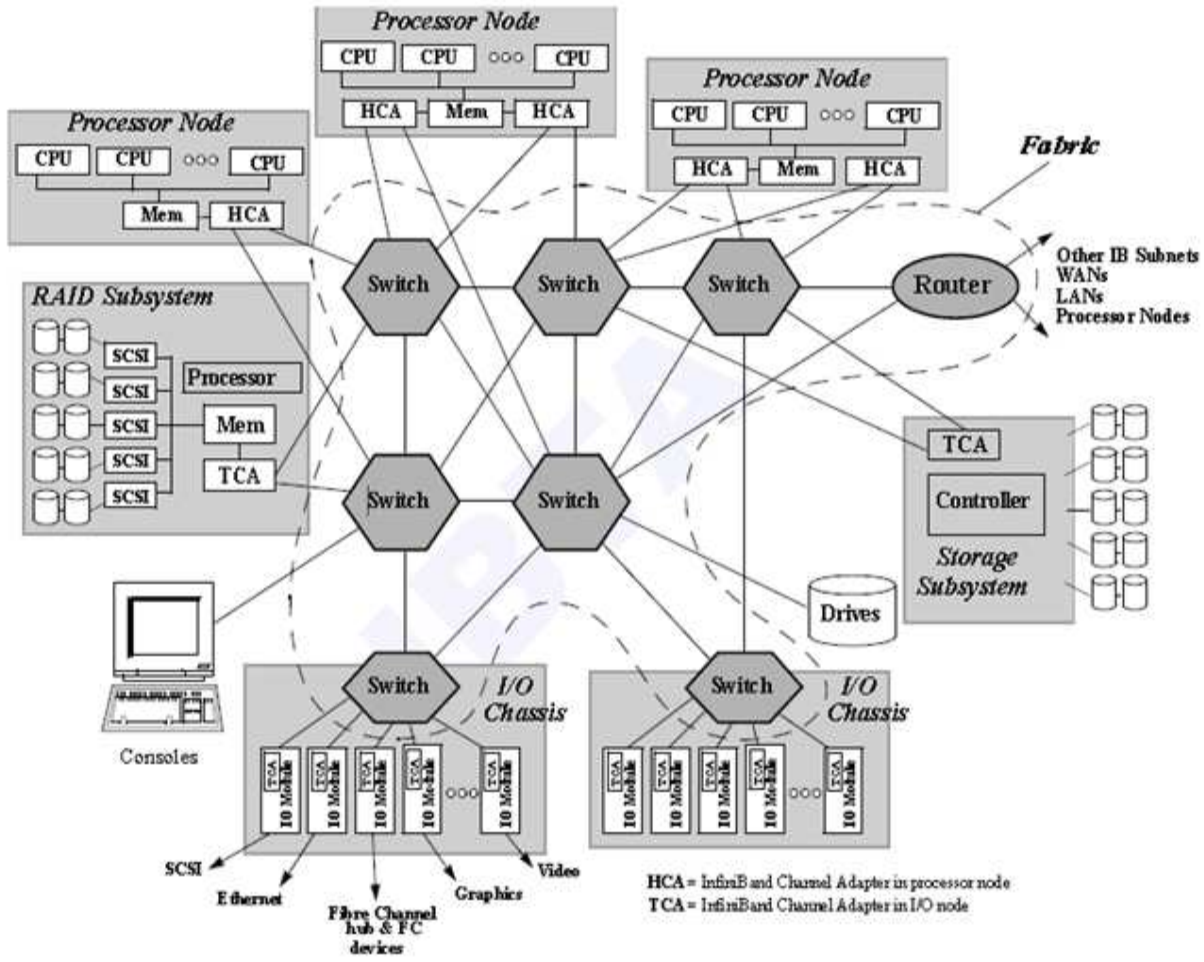

Figura 7 - Schema organizzativo di un possibile cluster HCP

Come si evince da figura 7, l'architettura IB è improntata su tecnologia point-to-point, switched I/O fabric, mediante la quale i dispositivi di end-node sono interconnessi con una serie di switch in cascata che costituiscono poi la sottorete. Le proprietà fisiche delle interconnessioni IB supportano due ambienti predominanti, con larghezza di banda, distanze e costi di ottimizzazioni appropriati per questi ambienti:

- Module-to-module, tipico di un computer system che supporta l'inserimento in slot di dispositivi I/O.
- Chassis-to-chassis, tipico di un sistema interconnecting computers con dispositivi esterni di storage in ambiente data-center.

IBA supporta differenti tipologie strutturali, da sistemi di singoli computer alla possibilità di espandere l'ambiente includendo: repliche di componenti per incrementare l'affidabilità, I/O aggiuntivo per ottimizzare le performance e la scalabilità, host node aggiuntivi per aumentare le prestazioni di calcolo. IBA è una architettura rivoluzionaria che permette di stare al passo con l'incessante incremento di richieste da parte dei consumatori in quanto per le caratteristiche su dette permette di incrementare la scalabilità e la larghezza di banda, decrementare l'utilizzazione spinta delle CPU, massimizzare availability[X] e isolation[XI], e sostenere la tecnologia internet.

2.2 Infiniband e DDR

La tecnologia InfiniBand basa il suo funzionamento sulle caratteristiche teoria delle memorie RAM, in particolare per il caso del sistema CRESCO, sulla tecnologia DDR applicata alle memorie, di cui daremo una descrizione specifica per la comprensione dei capitoli successivi.

La DDR SDRAM, acronimo di Double Data Rate Synchronous Dynamic Random Access Memory (in italiano "memoria dinamica ad accesso casuale sincrona a doppia velocità")[21], è un tipo di memoria RAM su circuiti integrati usati nei personal computer.

Possiede una larghezza di banda maggiore rispetto alla SDRAM poiché trasmette i dati sia sul fronte di salita che sul fronte di discesa del ciclo di clock. Questa tecnica consente di raddoppiare la velocità di trasferimento senza aumentare la frequenza del bus di memoria. Quindi un sistema DDR ha un clock effettivo doppio rispetto a quello di uno SDRAM.

[X] Availability: misura l'attitudine di un'entità ad essere in grado di svolgere una funzione richiesta in determinate condizioni ad un dato istante, o durante un dato intervallo, supponendo che siano assicurati i mezzi esterni eventualmente necessari.

[XI] Isolation:è una proprietà che definisce come e quando le modifiche di una operazione possono diventare visibili alle altre operazioni concorrenti.

Poiché i dati sono trasferiti 8 byte per volta (il bus è sempre a 64 bit) una RAM DDR dà una velocità di trasferimento VT di:

$VT = BM \cdot 2 \cdot 8$

dove BM è la velocità di clock del bus di memoria, 2 è il numero di invii per ciclo di clock, 8 è il numero di byte trasferiti ad ogni invio. Quindi con una frequenza di clock di 100MHz, una DDR SDRAM dà una velocità massima di trasferimento di quasi 1526 MBps.

Nella realtà la situazione è più complessa, poiché la velocità di trasferimento è notevolmente influenzata dai fenomeni di latenza, che si verificano durante le operazioni di lettura/scrittura e che dipendono strettamente dal tipo e dalla qualità del chip, nonché dalla frequenza di funzionamento.

2.2.1 Coesistenza SDR, DDR, e QDR

Per comprendere alcune analisi descritte nel terzo capitolo è fondamentale descrivere la capacità di InfiniBand di far cooperare tecnologie SDR, DDR e QDR nello stesso sistema di rete.

InfiniBand infatti, supporta queste tre tecnology, e gli apparati di rete che operano con interconnessioni IB permettono la trasmissione su link 4X SDR, 4X DDR e 4X QDR ossia rispettivamente 10, 20 o 40 Gbps.

Questo permette l'incremento del throughput dei link fisici e inoltre riduce il ritardo di serializzazione dei dati. Lo scheduling negli apparati IB è notevolmente dinamico e veloce, in quanto gli switch IB effettuano il forwarding dei pacchetti analizzando solo l'external header aggiunto sul pacchetto, permettendo all'intera rete sia di sfruttare al meglio le velocità raggiunte a livello dei link, sia decrementare notevolmente i valori di latenza.

La cooperazione delle tre capacità di trasferimento dati è permessa grazie alla politica di gestione dei dati da parte degli switch presenti in rete:

Se un pacchetto è inviato con un data rate (SDR) e poi deve essere commutato su link con larghezza di banda maggiore (DDR o QDR), la tecnica di inoltro del pacchetto da parte dello switch deve essere del tipo *Store-and-forward* (traducibile come "immagazzina e rinvia").

Questa è una tecnica nella quale un'informazione (suddivisa in pacchetti), nel suo percorso tra le singole stazioni (o nodi) della rete, deve essere totalmente ricevuta, prima di poter essere ritrasmessa nel collegamento in uscita.

In pratica, il pacchetto viene ricevuto e verificato (la verifica può avvenire ad esempio controllando la correttezza del CRC[XII]), e solo allora viene ritrasmesso; ciò implica quindi lo svantaggio che se tra due singoli nodi vi è un ritardo, questo verrà moltiplicato per tutti i nodi che l'informazione dovrà attraversare per giungere a destinazione, aumentando notevolmente la latenza di trasferimento. La possibilità di far cooperare SDR, DDR e QDR quindi genera un compromesso tra convivenza in rete di apparati con caratteristiche differenti e una latenza non propriamente ottimizzata.

[XII] CRC: Cyclic redundancy check (ovvero Controllo a ridondanza ciclica) è un metodo per il calcolo di somme di controllo (*checksum* - sequenza di bit che viene utilizzata per verificare l'integrità di un dato o di un messaggio che può subire alterazioni). Utile per l'individuazione degli errori casuali, il CRC non è invece affidabile per verificare la completa correttezza dei dati contro tentativi intenzionali di manomissione.

2.3 L'architettura InfiniBand

InfiniBand è fondato su un'architettura punto-punto basata su canali di comunicazione seriali. La rete a commutazione può avere come nodi server, canali in fibra, router e sistemi di storage.

Ogni nodo può comunicare con gli altri in una configurazione a più collegamenti e le ridondanze possono essere utilizzate per incrementare la robustezza del sistema.

Dal punto di vista fisico, un singolo canale InfiniBand ha una struttura molto simile a quello dell'architettura Pci Express[23]: vi sono due contatti per direzione che sono in grado di fornire una banda di 2,5 Gbps (250 MByte al secondo utilizzando un metodo di codifica a 8/10 bit simile a quello di Pci Express). I canali possono essere aggregati lungo un collegamento in configurazioni da 1, 4 o 12 coppie di segnali (figura 8).

La banda massima raggiungibile è di poco più di 90 Gbps per direzione, se si utilizza la modalità 12x. Sono supportati collegamenti sia in fibra sia in rame. La distanza massima tra due nodi connessi tramite fibra è di oltre 300 metri e ogni sottorete può gestire fino a 64.000 nodi.

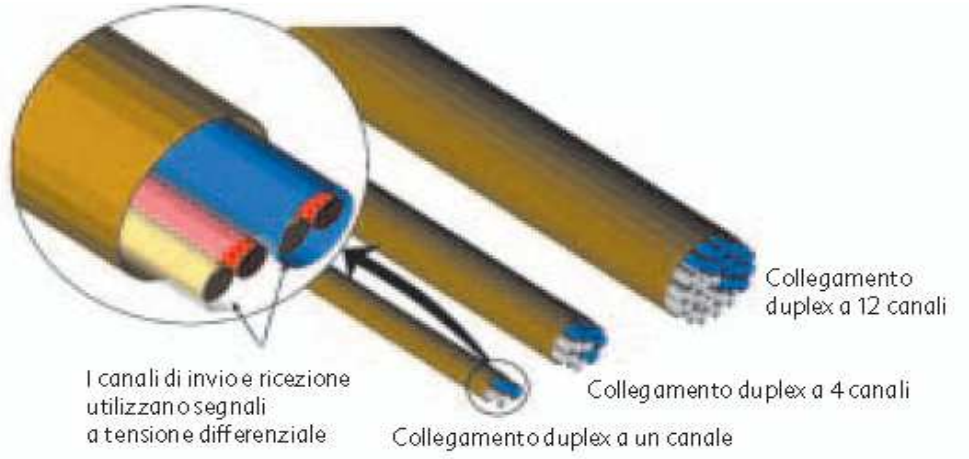

Figura 8 - Coppie 1X, 4X e 12X di un cavo InfiniBand

2.3.1 Dipendenze InfiniBand: protocolli e librerie

IPoIB

Ip over InfiniBand (IPoIB)[24] link driver provvede ad incapsulare pacchetti IP su InfiniBand. IPoIB può tranquillamente essere utilizzato in tutti i dispositivi che impiegano InfiniBand come tecnologia di trasferimento dati, dualmente a come IP si comporta per le reti Ethernet.

Si può usare IPoIB per la risoluzione degli indirizzi e per la gestione dei gruppi multi cast.

SDP

Sockets Direct Protocol (SDP)[25] è un protocollo di trasporto usato sulla rete InfiniBand per permettere operazione sockets-based per ottimizzare le performance di RDMA su IB (come descritto nel sottoparagrafo 2.3.2).

SDP riduce il numero di SW running dentro un process context. Zero-copy[XIII] SDP permette alle applicazione, ai servers e alle CPUs di operare in modo più efficiente in quanto i databases impiegano meno tempo per aspettare la copia dei dati, gli application servers aspettano un tempo inferiore per le risposte e le CPUs possono spendere le capacità di calcolo risparmiate per operare su altri work.

SRP

SCSI RDMA Protocol (SRP)[26] è un upper-layer storage protocol per InfiniBand che permette di far girare comandi SCSI su RDMA ed abilita le reti con host InfiniBand di comunicare con dispositivi di storage tramite Fibre Channel.

Il protocollo SRP usa servizi di comunicazione RDMA che provvedono alla comunicazione tra dispositivi, tramite messaggi di controllo e di gestione per il trasferimento dati.

uDAPL

User Direct Access Programming Library (uDALP)[27] è uno standard user modello API che supporta la rete InfiniBand. uDALP gestisce la correlazione name-to-address, stabilisce le connessioni e il trasferimento affidabile dei dati. La prima responsabilità di uDALP sono la gestione delle connessioni e il trasferimento finale di dati con bassa latenza.

[XIII] Zero-copy: Descrive l'operazione di un computer in cui una CPU non ha l'obbligo di copiare dati da un'area di memoria all'altra.

MPI

MPI - Message Passing Interface[28] è uno standard industriale che consiste in due documenti che specificano una libreria di funzioni per consentire il passaggio di messaggi tra nodi all'interno di un ambiente di calcolo parallelo indipendentemente dalla rete di appartenenza o dall'ambiente di calcolo stessa. Per queste caratteristiche, l'implementazione di MPI permette di utilizzare stack protocollari come TCP, UDP over IP, InfiniBand o Marynet per la comunicazione tra diversi processi. MPI è una componente middleware di tipo software situato tra l'applicazione e l'hardware di rete.
Due implementazioni di MPI sono OpenMPI e MVAPICH.

2.3.2 Strati protocollari IB

I protocolli di InfiniBand sono organizzati come consuetudine in strati, che comprendono il livello fisico, di collegamento, di trasporto, di rete e infine quello applicativo (figura 9).
Lo strato fisico descrive i parametri elettrici e meccanici di InfiniBand, tra cui le specifiche dei canali già esaminate.
I livelli di collegamento e di trasporto definiscono, commutano e consegnano i pacchetti e gestiscono il bilanciamento del carico e le funzioni di QoS.
Lo strato di rete introduce invece l'instradamento attraverso diverse sottoreti e definisce quindi le funzioni dei router.

L'architettura Infiniband prevede che dispositivi finali e nodi di rete comunichino mediante lo scambio di pacchetti dati con modalità trasmissive orientate alla connessione. L'unità base del protocollo di comunicazione è il messaggio, che però può assumere diverse rappresentazioni e significati a seconda del contesto di utilizzo.

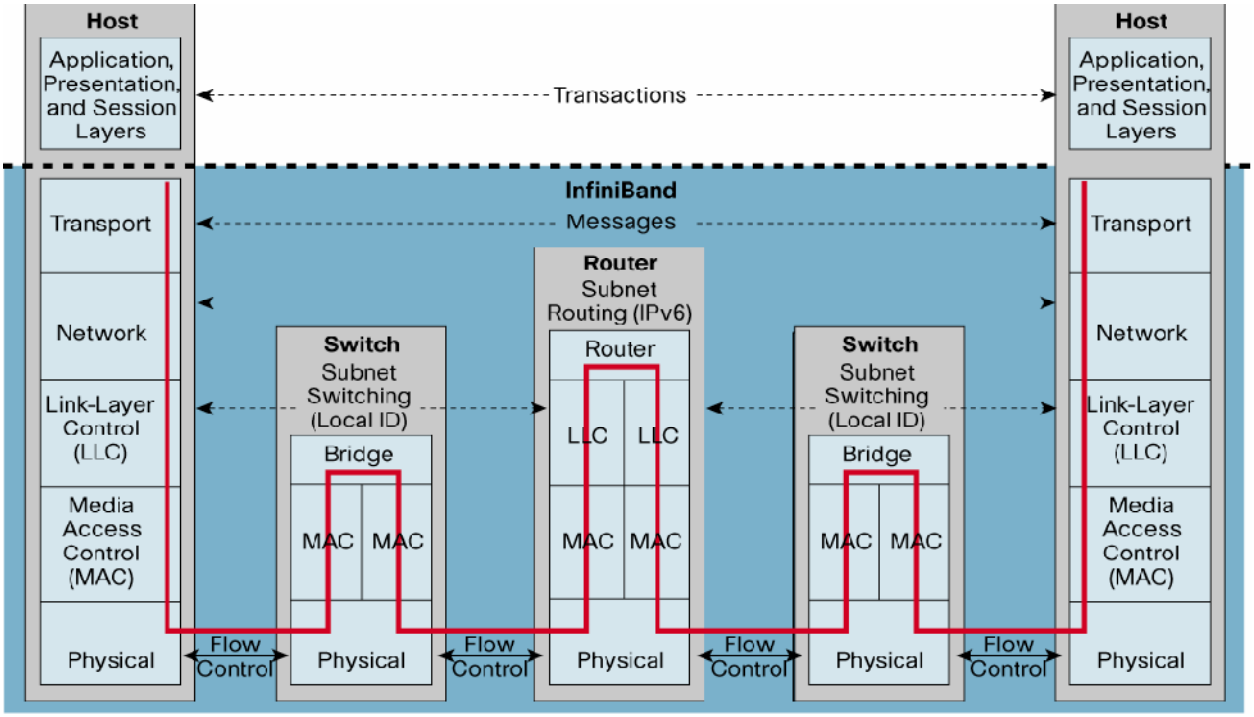

Figura 9 - Strati protocollari dello standard InfiniBand

Il livello di trasporto, sul quale ci siamo soffermati in modo particolare in questo elaborato, ha il compito di garantire la comunicazione tra le applicazioni residenti sul sistema, controllandone l'affidabilità della consegna e del controllo del flusso.

Per il trasferimento dei dati, il dispositivo HCA si affida ad un sistema RDMA (Remote Direct Memory Access)[29] che permette un alto throughput ed una bassa latenza, evitando di sovraccaricare la CPU di interrupt[XIV], che si limita solamente a dare lo start alle operazioni.

Per sfruttare RDMA, una applicazione dovrebbe essere in grado di copiare i dati nella memoria dell'applicazione stessa, e non nel buffer del sistema operativo conformandosi ad una politica zero-copy. RDMA quindi risolve il problema della circolazione dei dati, ma non quello della multiplazione, del sequenziamento e dell'affidabilità.

Per ovviare a questo, InfiniBand utilizza un concetto di queue pairs (QPs), ossia interfacce virtuali associate con l'HCA per inviare e ricedere dati.

[XIV] Interrupt: è un tipo particolare di istruzione della CPU che consente l'interruzione di un processo qualora si verifichino determinate condizioni oppure il processo in esecuzione debba effettuare una richiesta al sistema operativo. È come un segnale o messaggio, generalmente di natura asincrona, che arriva all'interno della CPU per avvisarla del verificarsi di un certo evento.

Quando una applicazione è invocata, è l'applicazione stessa che crea una o più QPs con relativi registri di memoria, dove i dati possono essere scritti. Per il controllo del flusso informativo, l'applicazione crea una coda di completamento per indicare il termine di una operazione. La possibilità di richiedere locazioni di memoria locali o remote è considerata dallo standard IB, per gestire queste eventualità, vengono generati dei permessi a chiave (Local-Key o Remote-Key) che l'HCA verifica.

Il controllo degli errori, e quindi dell'affidabilità del sistema, è affidato a tecniche CRC (*Cyclic redundancy check*). Nello standard InfiniBand se ne definiscono di due tipi:
ICRC, Invariant CRC ha la possibilità di rivelare gli errori nei settori che non sono soggetti a probabili modifiche nel percorso end-to-end;
VCRC, Variant CRC ha la possibilità di rivelare gli errori dei settori che possono subire modifiche, proprio per questo il controllo è eseguito hop-by-hop[XV].

Ultima operazione che lo strato di trasporto compie è quella di suddividere il messaggio ricevuto in segmenti di dimensioni più piccole che andrà a costituire il payload di un pacchetto dati del livello di rete. Ogni pacchetto passato dal livello di trasporto a quello di rete, oltre al payload, contiene anche un identificatore unico globale (GUI) del nodo di destinazione del pacchetto stesso. In linea generale ogni pacchetto può contenere sino a un massimo di 4.096 ottetti, ma il numero effettivo può essere variabile e tarato in funzione del dispositivo o della tipologia di rete di trasporto.
Un'applicazione tipica della tecnologia Infiniband è, per esempio, la interconnessione di server a dispositivi di storage remoti o a reti geografiche. Il server contiene al suo interno un adattatore di canale (Host Channel Adapter, HCA), che connette il controller della memoria agli apparati della rete commutata (in pratica gli switch di rete) tramite uno o più canali bidirezionali. Dispositivi finali, come per esempio apparati di storage, si collegano agli switch della rete mediante un'interfaccia riferita come Target Channel Adapter (TCA), più semplice costruttivamente dell'HCA e con le funzioni specifiche del dispositivo e di quanto serve per operare all'interno dell'architettura Infiniband.

[XV] Hop-by-hop: Termine che indica il susseguirsi di un evento link dopo link. Un hop (ossia un salto) è propriamente identificato come il collegamento unico che collega due dispositivi.

La connessione tra HCA e TCA prevede il ricorso a un link seriale bidirezionale ad alta velocità, che contiene dei canali dedicati alla trasmissione e alla ricezione. In caso di necessità, la capacità globale può essere incrementata aumentando il numero di canali utilizzati contemporaneamente per un link. In pratica questo incremento, a livello di architettura, prevede il ricorso a 1, 4 e 12 coppie di link bidirezionali, con metà canali di ogni link utilizzata per la trasmissione e metà canali per la ricezione. Ogni canale unidirezionale ha una larghezza di 1 bit e una velocità teorica di segnalazione pari a 2,5 Gbps. Ne deriva che i 2, 8 e 24 canali (trasmissione più ricezione) dispongono di una banda teorica di 10, 20 e 60 Gbps rispettivamente.

Come avviene generalmente in tutte le reti a commutazione, l'architettura permette di ottimizzare l'utilizzo delle risorse trasmissive, ammettendo di suddividere ogni link in un insieme di percorsi virtuali utilizzabili contemporaneamente dalle diverse applicazioni, che sono connesse agli stessi nodi di origine e di destinazione di un link. Per ogni percorso virtuale, all'interno di un link, esiste un meccanismo individuale di controllo del flusso, che permette a una coppia di dispositivi finali di comunicare senza interferire con altri percorsi virtuali relativi ad altri dispositivi. Su ogni link è possibile allocare da un minimo di due a un massimo di 16 percorsi virtuali. Un percorso è comunque riservato per la gestione dei dispositivi, mentre i restanti sono utilizzabili per il trasporto dei pacchetti dati delle applicazioni.

La suddivisione in percorsi virtuali permette di ottimizzare l'utilizzo della rete e occupare una banda corrispondente alle effettive esigenze trasmissive dell'applicazione. Per esempio, diventa possibile suddividere un link a 250 Mbps in percorsi virtuali a 50 Mbps. Come appare evidente, Infiniband ha l'obiettivo di espandere il concetto di I/O al di fuori dello stretto ambito di un server e creare le basi per un approccio globale, allargato alla parte di rete locale e geografica e ai sistemi di storage di nuova generazione.

2.3.3 Gli attori InfiniBand

L'architettura InfiniBand definisce quattro classi di nodi che compongono la rete di comunicazione:

- Host Channel Adapter (HCA): risiede sui server o comunque su piattaforme dotate di Cpu e gestisce la richieste di trasmissione.
- Target Channel Adapter (TCA): dualmente all'HCA, risiede tipicamente sui sistemi di immagazzinamento dati con lo scopo di ricevere le richieste dei server.
- Switch: si occupa della gestione del traffico all'interno di una medesima sottorete.
- Router: instrada il traffico tra le diverse sottoreti.

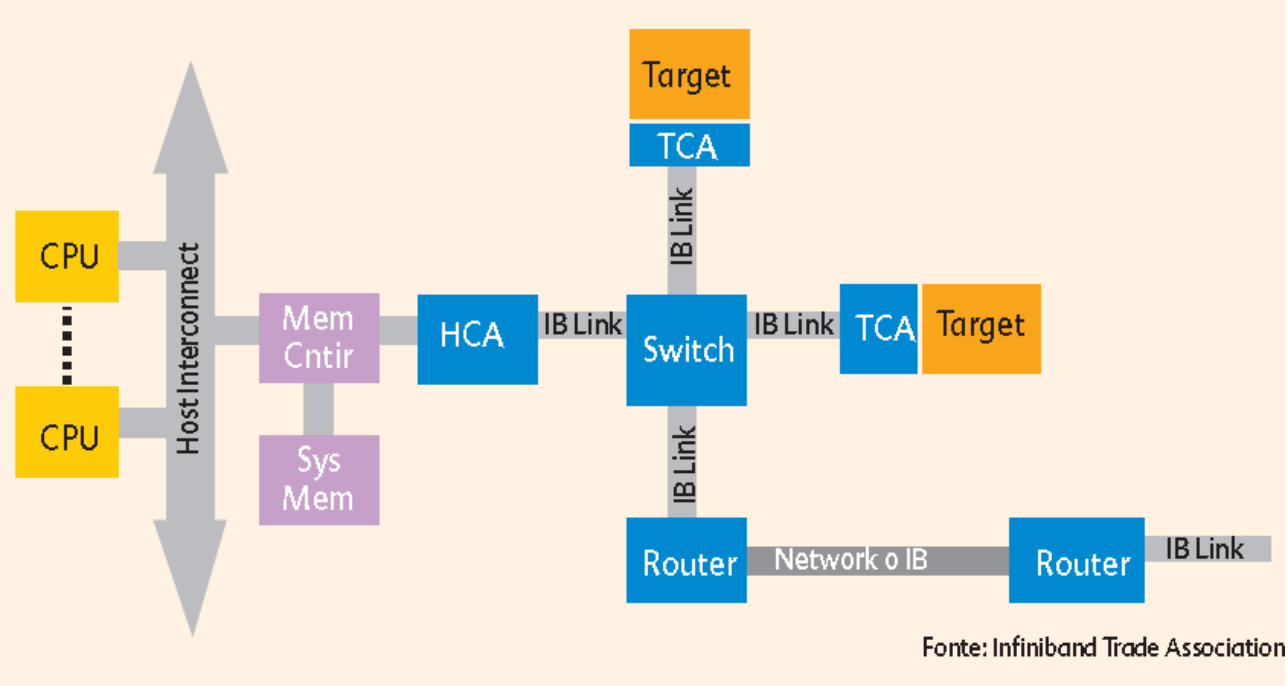

Figura 10 - Connessioni e attori InfiniBand

Ogni channel adapter può avere una o più porte, ognuna delle quali è collegata con uno switch e può dialogare con ogni altra porta sulla rete (figura 10).

Una sottorete è gestita da un nodo con funzioni di Subnet Manager (SM), al quale verrà dedicato un paragrafo a parte perché componente chiave dell'infrastruttura InfiniBand, che monitora il network aggiungendo o eliminando i dispositivi in relazione ai cambia-

menti di topologia e assegnando gli identificativi locali (Lid) ad ogni porta direttamente connessa. Per bilanciare il carico di rete, può venir associato uno stesso Lid a port differenti, come descritto nella sezione 2.9.3 con relazione diretta in figura 15.
Ogni channel adapter possiede poi un Guid (Globally Unique Identifier) che unito al Lid forma un indirizzo globale sulla rete InfiniBand (Gid). Quest'ultimo rispetta lo standard di Ipv6 consentendo una perfetta integrazione tra una rete InfiniBand e altri network. I router utilizzano i Gid per costruire le tabelle di instradamento fra le diverse sottoreti.

2.4 Procedure di inizializzazione IB

Una trasmissione inizia con la richiesta, da parte di un nodo, di un servizio alla rete. L'istanza è quindi passata all'HCA che apre un WQE (*Work Queue Element,* elemento di coda di lavoro) per gestire la comunicazione, e crea una o più coppie di code (Qp, *Queue Pair*) ognuna delle quali è composta da una sezione di invio e da una sezione di ricezione. Un HCA può mantenere attive fino a 224 QPs che forniscono l'interfaccia virtuale al collegamento. I pacchetti possono essere indirizzati sia a un unico target sia a un indirizzo multicast. Dal lato target, la coda di ricezione del WQE alloca i dati al richiedente e, al termine della transazione, sono chiusi i Wqe e le QPs. Gli header dei pacchetti identificano la sorgente, la destinazione e le priorità per applicare regole di QoS, oltre a verificare l'integrità delle informazioni. È supportato anche un sistema a credito simile a quello di Pci Express tramite il quale l'invio dei dati è effettuato solo dopo che è verificata la disponibilità di risorse da parte del ricevente.

2.5 Errori "ibcheckerrors"

Gli errori che possono verificarsi in una interconnessione InfiniBand, e definiti come *non-critical errors* sono i seguenti:

"SymbolErrors": Errore sul singolo pacchetto. Nessuna azione è richiesta ad eccezione che il contatore incrementi di pari passo con LinkRecovers.

"LinkRecovers": Se questo errore incrementa insieme con SymbolErrors può indicare un bad link.

"LinkDowned": Numero di volte che la porta è andata down (di solito per valide ragioni).

"RcvErrors": Questo errore indica un bad link, se il link è interno al dispositivo switch provare a settarlo in SDR, altrimenti controllare il cavo.

"RcvRemotePhyErrors": Questo errore indica un problema "elsewhere" (altrove) nella rete.

"XmtDiscards": Questo errore è sintomo di congestione nei link. E può richiedere di apportare delle modifiche ai parametri dell'HOQ[XVI].

"XmtConstraintErrors": Questo è un errore dovuto ad un errato partizionamento del sistema.

"RcvConstraintErrors": Simil XmtConstraintErrors.

"LinkIntegrityErrors": Può indicare un bad link.

"ExcBufOverrunErrors": Questo è un errore che riguarda il controllo del flusso di una macchina, e può indicare un errore fisico del pacchetto trasmesso.

"RcvSwRelayErrors": Questo errore indica degli errori di trasmissione.

Errori di natura più complessa, non verranno trattati in questo elaborato, perché non rientrano nelle problematiche riscontrate nel sistema HPC CRESCO.

[XVI] HOQ Manager: (Head Of Queue Manager) gestisce I dati presenti nella memoria, e scarta I pacchetti nelle trasmissioni se il parametro di timing HOQ è scaduto.

2.6 Vantaggi e svantaggi di InfiniBand

La tecnologia InfiniBand costituisce un sistema complesso, ma la sua complessità permette di avere prestazioni elevate rispetto ad altri sistemi. Gli stessi attori che compongono la rete, contribuiscono a favorire alte prestazioni: ad esempio i channel adapter, dotati di una propria intelligenza, liberano il sistema operativo e le CPU dalla maggior parte del carico di lavoro di rete. L'architettura a commutazione conferisce inoltre una grande scalabilità al sistema e supporta un alto numero di connessioni simultanee senza rischio di collisioni e senza significativi incrementi di latenza.

Anche l'affidabilità del sistema è garantita dalla possibilità di rendere i collegamenti ridondanti e un eventuale problema a un disco o a una periferica di memoria può essere risolto rimovendo l'unità difettosa senza dover spegnere i server, visto che le unità di storage sono separate dalle CPU.

L'uso di switch con tecniche di forwarding cut-through[XVII] permette al sistema di avere una bassa latenza, che permette prestazioni elevate anche in presenza di una ricezione continua di pacchetti dati.

La latenza di trasmissione, in un contesto di rete basata su protocolli TCP/IP, di un collegamento è il tempo necessario (espresso in millisecondi) impiegato da uno o più pacchetti ICMP[XVIII] a raggiungere un altro computer o server in rete (sia essa Internet o LAN).

Nell'ambito delle reti i fattori che influenzano maggiormente la propagazione del segnale sono il mezzo che trasporta l'informazione e le apparecchiature (per esempio switch o router) che il segnale attraversa nel suo percorso.

Per un sistema di alto livello come CRESCO, avere una bassa latenza nella trasmissione dei dati è di fondamentale importanza tanto quanto una connessione veloce tra i dispositivi.

[XVII] Cut-through: lo switch si limita a leggere l'indirizzo MAC del destinatario e quindi manda il contenuto del frame contemporaneamente alla sua lettura. In questo caso l'invio dei frame non attende la ricezione completa dello stesso. Questo tipo di switch è quello con latenza minore.

[XVIII] ICMP: IInternet Control Message Protocol (ICMP) è un protocollo di servizio che si preoccupa di trasmettere informazioni riguardanti malfunzionamenti, informazioni di controllo o messaggi tra i vari componenti di una rete di calcolatori.

La latenza di uno switch che supporta tecnologia InfiniBand è di 200 nanosecondi per SDR e di 140 nanosecondi per DDR. Il range di latenza end-to-end, in funzione delle diverse case di sviluppo, varia tra 1.07 – 2.6 microsecondi.

Un vantaggio/svantaggio è invece la possibilità di far cooperare trasmissioni di tipo DDR e SDR. Questo permette infatti di inserire nella stessa sotto-rete dispositivi con diverse caratteristiche tecniche, come ad esempio l'inserimento di nodi Cell, ottimi per computazioni grafiche ma operanti solo con tecnologia SDR. Lo svantaggio è il dover adattare il flusso in base alla caratteristica del link, sia esso DDR o SDR. È lo switch che si prende l'onere di adattare le tipologie di flusso, trattando i dati attraverso la tecnica di store-and-forward[XIX], che crea però una latenza maggiore.

Ci sono pochi svantaggi rilevanti per la tecnologia InfiniBand, possiamo individuarli proprio per la giovane età della struttura InfiniBand o nella difficoltà di trattare velocità così importanti nei gateway di frontiera.

2.6.1 Confronto con altre tecnologie di interconnessione

Fondamentale importanza nella scelta di una tecnologia di interconnessione da utilizzare in un cluster HPC la si dà a due parametriti in particolare: Banda e latenza.

Banda di trasferimento elevata è importante per la per la quantità di dati trattati, e per la necessità di avere la maggiore velocità di trasferimento possibile. La latenza è un parametro che potrebbe far aumentare notevolmente i tempi di attesa di un trasferimento dati, soprattutto tra dispositivi che attraversano una discreta quantità di apparati di rete. È importante avere la possibilità di gestire il traffico tramite dispositivi che abbattono notevolmente i tempi di latenza. Questo è quello che avviene con InfiniBand, dove gli switch di rete hanno una latenza interna di 200 nanosecondi. Questo è permesso dalla modalità con la quale i pacchetti vengono instradati, ossia tramite il solo Lid (Local Identifier) che viene identificato tramite l'header del pacchetto IB e instradato in rete.

Le alte prestazioni di banda, sono raggiunte dalla possibilità di creare cavi unici con all'interno più coppie di cavi base.

[XIX] Store-and-forward: lo switch legge l'intero frame e ne viene calcolato il cyclic redundancy check (CRC) confrontandolo con il campo FCS all'interno del frame. Solo se i due valori corrispondono il frame viene mandato al destinatario, altrimenti non viene trasmesso. Questi tipi di switch consentono di bloccare frame contenenti errori ma hanno una latenza maggiore.

Anche la codifica di canale permette ad InfiniBand di avere un data rate alto, tramite codifica 8B/10B (su 8 bit di dati sono trasferiti 10 bit di simboli), il transfert rate reale si avvicina moltissimo a quello nominale.
InfiniBand non è l'unica architettura usata per la trasmissione veloce e a bassa latenza tra dispositivi di rete. L'interesse nel creare sistemi HPC di alto livello nel campo dell'ICT (*Information and Communication Technology*)[30], ha permesso lo sviluppo di una quantità elevata di standard protocollari per le trasmissioni dati ad alta banda e a bassa latenza.
Il protocollo Myrinet[31], ad esempio, è un sistema high-speed LAN creato dalla Myricom, per le interconnessioni tra dispositivi di calcolo in un cluster. Myrinet possiede un overhead limitato tra gli strati protocollari, grazie al quale ottiene: alto throughput, minori interferenze, bassa latenza end-to-end. La connessione fisica di Myrinet si compone di due cavi in fibra ottica, upstream e downstream, connessi ad ogni singolo nodo tramite un unico connettore. È un sistema che previene numerosi *fault-tolerance*[xx], coadiuvato anche dai dispositivi di switching. Questi possono essere: flow control, error control, presenza dell'*heartbeat* su ogni link.
La quarta generazione di Myrinet, supporta trasmissioni a 10 Gbps con possibilità di cooperare con la 10 Gigabit Ethernet sul PHY (physicallayer), con latenze dell'ordine dei alcuni microsecondi.

QsNet II è l'ultima nata della famiglia Quadrics Interconnect[5], che si interfaccia ai nodi di calcolo tramite il bus definito dallo standard IO PCI-X. Permette basse latenze, dell'ordine di alcuni microsecondi, e una soddisfacente capacità di banda per link.
I cores di QsNet sono basati su due tipologie ASICs: Elan4 e Elite4. Il primo è un processore di comunicazione che crea l'interfaccia tra la rete high-performance e il nodo di calcolo con uno o più CPUs. Elite4 è un componente switch che può operare con 8 links bidirezionali, operanti in entrambe le direzioni contemporaneamente ad un data rate di 10,5 Gbps.

[xx] Fault-tolerance: è la capacità di un sistema di non subire fallimenti (cioè intuitivamente interruzioni di servizio) anche in presenza di guasti. La tolleranza ai guasti è uno degli aspetti che costituiscono l'affidabilità. È importante notare che la tolleranza ai guasti non garantisce l'*immunità* da tutti i guasti, ma solo che i guasti per cui è stata progettata una protezione non causino fallimenti.

2.7 IBA sul sistema HPC CRESCO

Un sistema HPC ad alte prestazioni, deve supportare una tecnologia di interconnessione che non limiti le prestazioni dei dispositivi utilizzati, incorrendo ad esempio in situazioni di bottleneck[XXI] o latenze troppo elevate.

La tecnologia InfiniBand risolve questo problema, garantendo al sistema CRESCO una capacità di banda tra tutti i link molto elevata ed una latenza bassissima, che permette di ottimizzare le prestazioni fornite dal sistema HPC.

2.8 CRESCO - Computational Research Center for Complex Systems.

CRESCO (Computational Research Center for Complex Systems)[2] è un progetto ENEA, promosso dal MUR (Ministero dell'Università e della Ricerca), che si focalizza, sul piano delle applicazioni e dei contenuti scientifici, sui seguenti ambiti:

- Lo studio di oggetti biologici dal punto di vista “sistemistico” e lo studio di sistemi naturali secondo il paradigma dei sistemi complessi;
- Lo studio si sistemi tecnologici e sociali complessi e delle loro mutue interazioni, e la realizzazione di opportuni strumenti di modelling, la simulazione e il controllo di questi sistemi e di quelle interazioni;
- L’implementazione di soluzioni innovative GRID computing per le attività di ricerca e sviluppo di punta dell’ENEA che richiedono l’utilizzo di risorse computazionali estremamente importanti.

[XXI] Bottleneck: **o** collo di bottiglia**,** è il punto in cui il sistema ha le minori performance tra un insieme di punti da percorrere.

2.8.1 Il sistema HPC CRESCO

Il sistemi CRESCO HPC[32] è stato progettato con lo scopo di fornire un sistema HPC alla stato dell'arte della tecnologia multi-core x86_64.

Le prestazioni ottenute nel giugno 2008, hanno permesso ad ENEA di posizionarsi al 125° posto della classifica mondiale Top500[7] relativa ai sistemi HPC [Rmax = 17.1 TFLOPS nel benchmark HPL].

Entrando più nel dettaglio nella descrizione del sistema HPC CRESCO, parleremo ora delle tecnologie impiegate e della sua struttura architetturale.

È importante fare una panoramica generale sul tipo di richieste effettuate dagli utenti e in particolar modo sulle applicazioni eseguibili sul sistema CRESCO. Gli applicativi disponibili sul sistema sono i principali software di calcolo che il mondo della ricerca usa per sviluppare i propri studi. Tra i codici proprietari, sono disponibili Fluent, Abaqus, Ansys e Nastran, per lo sviluppo di codice invece, sono disponibili i compilatori GNU, INTEL e PORTLAND. Il sistema CRESCO viene utilizzato nella maggior parte dei campi di ricerca dell'ENEA, ma il carico maggiore arriva da studi riguardanti:

- Climatologia;
- Fusione Nucleare, studi di Stabilità e Modellizzazione;
- Fluidodinamica;
- Chimica computazionale.

Per rispondere a tali esigenze, CRESCO è stato strutturato in 3 sezioni di calcolo, cooperanti tra di loro, in modo tale da offrire un servizio ottimale.

La prima sezione risponde alle esigenze di grande memoria e moderata scalabilità parallela, ed è composta da 42 fat nodes IBM x3850-M2 con 4 processori Xeon Quad-Core Tigerton E7330 (2.4GHz/1066MHz/6MB L2), 32 GB di RAM (con 12 extra-fat nodes con 64 GB di RAM) per un totale di 672 cores.

La seconda sezione è orientata ai casi di alta scalabilità, composta da 256 blades IBM HS21 ognuno dei quali supportati da processori dual Xeon Quad-Core Clovertown E5345 (2.33GHz/1333MHz/8MB L2), 16 GB di RAM per un totale di 2048 cores.

Entrambe le sezioni sono interconnesse in rete tramite tecnologia InfiniBand 4X DDR, alla quale è dedicata questa tesi.

La terza sezione del sistema CRESCO, recentemente implementata consiste in 3 sotto-sezioni dedicate all'utilizzo sperimentale di processori particolari, Cell processor, FPGA e GPU.

Tale sezione si compone di 4 blades IBM QS21 con 2 processori Cell BE 3.2 GHz, 6 nodes IBM x3755, 4 sockets AMD Dualcore 8222 equipaggiati con FPGA VIRTEX5 LX330, 4 nodi IBM x3755 con schede NVIDIA Quadro FX 4500X2 e 4700X2 per grafica e video.

Tutte le tecnologie descritte sono inserite in appositi rack, disposti su due file parallele (figura 11), in modo tale da far coincidere la regione calda nella parte interna del corridoio.

Lo storage che supporta la memorizzazione dei dati e lo scambio degli stessi è un IBM/DDN 9550 con 160 TB di memoria, organizzati tramite GPFS File System.

Figura 11 - Centro Elaborazione Dati di CRESCO

2.9 L'architettura di CRESCO

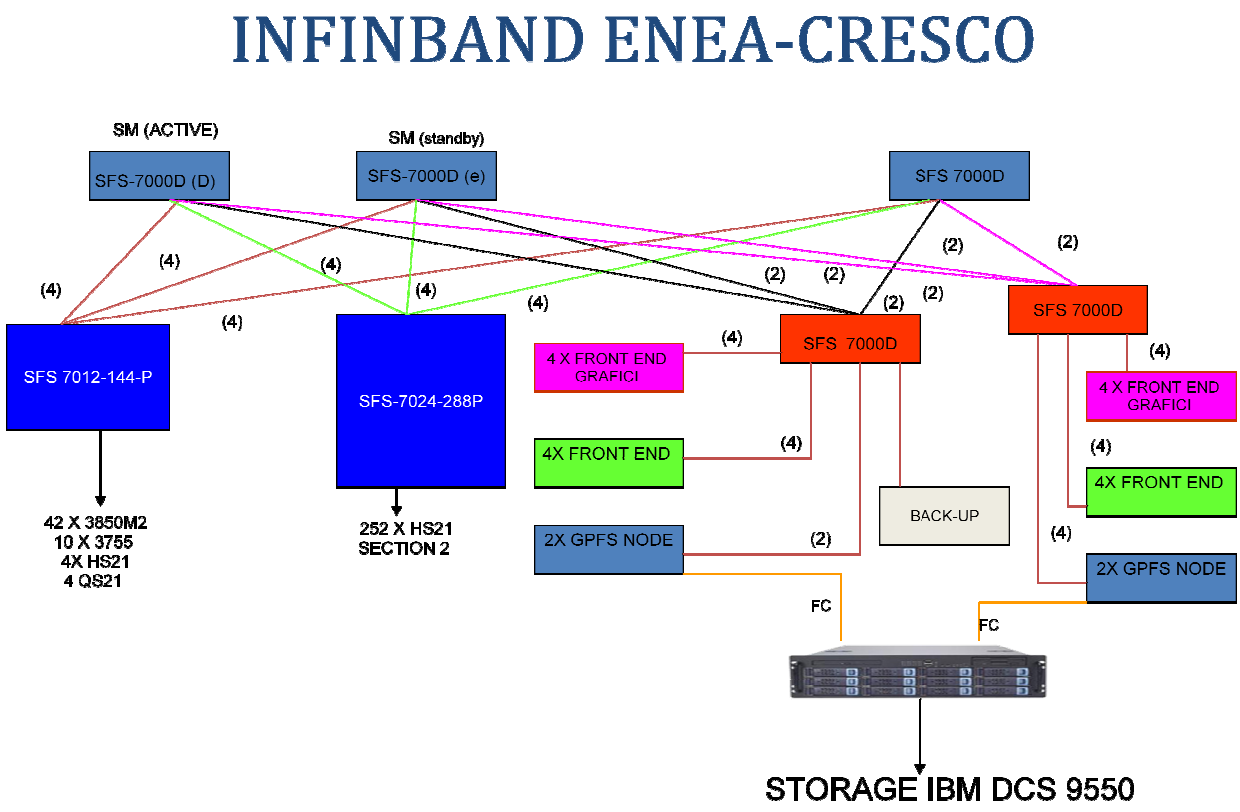

Figura 12 - Architettura dell'infrastruttura di CRESCO

La rete InfiniBand che gestisce le interconnessioni di CRESCO è basata essenzialmente su 5+2 dispositivi switch che costituiscono l'architettura descritta in figura 12.

I primi 5 dispositivi (tutti switch Cisco SFS 7000d) possono essere suddivisi in 2 rami applicativi, il primo (contenente 3 dispositivi, indicati con colore celeste) hanno la funzione di gestire ed amministrare tutte le politiche relative al trasferimento dati, condivisione, segnalazione, gestione I/O, gestione degli errori ed interfaccia con l'esterno. Di questi 3 switch, uno solo viene eletto Master, ed è colui che gestisce l'intera rete IB, prendendo in carico tutta la messaggistica e il forwarding dei dati; i restanti 2 sono detti switch di Standby, che oltre ad aumentare il numero di interconnessioni tra i dispositivi, sono di ausilio al SM nel caso di guasti, prendendo in eredità la funzionalità di Master SM.

Gli altri 2 switch (sempre switch Cisco SFS 7000d ed indicati con il colore rosso) sono connessi direttamente ai nodi dedicati allo storage e ai nodi di front-end.

In InfiniBand, lo switch Master prende il nome di Subnet Manager, ossia il dispositivo che prende in carico la gestione completa della rete. Per la complessità di questo oggetto e per l'importanza che ricopre in questo lavoro sarà descritto nel dettaglio in un paragrafo a parte.

Altri dispositivi chiave del sistema CRESCO sono i 2 switch che permettono la connessione con i nodi di calcolo. Sono il Cisco SFS 7024 composto da 288 porte e il Cisco SFS 7012 composto da 144 porte che formano, insieme ai 2 switch connessi allo storage, il secondo livello del sistema.

Possiamo infatti individuare 2 livelli architetturali, che analizzeremo nello specifico nei paragrafi successivi, per comprendere in modo più dettagliato le connessioni che caratterizzano il sistema CRESCO.

Il primo livello, composto da 3 switch Cisco SFS 7000d (1 SM Master e 2 SM standby).

Il secondo livello, composto da 2 switch Cisco 7000d e 2 Cisco 7024 e 7012

Questa struttura a livelli permette di avere una idea generale di come l'architettura ad albero permette alle comunicazioni tra i nodi di effettuare lo stesso numero di hop per poter comunicare tra di loro e con il SM.

2.9.1 Primo livello

La gestione totale della rete IB di CRESCO è affidata ad un sistema di 5 switch che sono alla base della struttura ad albero della rete. Tre dei cinque switch hanno caratteristiche logiche comuni, nel senso che sono impiegati essenzialmente per smistare il traffico dati e aumentare la capacità di banda del sistema completo. Unica differenza sostanziale è che uno di questi 3 switch è eletto Master con funzionalità si Subnet Manager (SM), ossia il dispositivo che ha la gestione completa delle politiche di gestione della rete.

Figura 13 - Backplane switch Cisco SFS 7000d

Gli altri 2 switch hanno la funzionalità principale di connettere alla rete il sistema di storage e i nodi di front-end.

Switch SFS 7000d

Le caratteristiche di Cisco SFS 7000D [33] sono:

- 24 porte dual-speed InfiniBand 4X che possono operare in modalità 20-Gbps Double Data Rate (DDR) o 10-Gbps Single Data Rate (SDR);
- Banda trasversale Non-bloccante a 480-Gbps con una latenza tra porte minore di 200 nanosecondi;
- Supporta cambi di capacità (da DDR a SDR v.v.) tra link;
- Gestione del sistema tramite Command-Line Interface (CLI), Web, e Java;
- Powered ports to enable flexible copper and optical interface configurations
- Test di diagnostica avanzati.

Cisco SFS 7000D offrendo alta affidabilità, sicurezza, controllo necessario per l'high performance e utility connectivity è il dispositivo ideale per sfruttare in modo ottimale le altissime velocità che la tecnologia InfiniBand offre alla rete. Sfruttato insieme ad altri dispositivi dello stesso tipo, provvede a creare un cluster HPC di notevoli dimensioni, contenente migliaia di nodi di calcolo, supportando una elevata domanda di applicativi propri di sistemi HPC. Si denota quindi una elevata propensione alla scelta di avere una elevata scalabilità rispetto magari ad una leggera complessità realizzativa.
Un'altra caratteristica importante che questo dispositivo offre alla rete è la sua alta affidabilità. Oltre alle ridondanze fisiche che possiede, ossia un sistema di raffreddamento ed alimentazione doppia, vi sono alcuni sistemi software, come il Cisco InfiniBand Subnet Manager, che mantengono costantemente sincronizzati i vari sistemi tra i vari livelli dell'infrastruttura. Questo permette al sistema di ovviare a qualsiasi stato di errore da parte degli switch di primo livello, sostituendo il SM in caso di guasti con un ulteriore switch che prende il suo posto, oppure nel caso che qualche link vada in down permette in tempi brevissimi il rerouting del path.

Tramite il suddetto sistema di gestione, i comandi di gestione e configurazione sono semplificati, essendo costituiti essenzialmente da comandi CLI[XXII] e GUI[XXIII], Telnet[XXIV] o Secure Shell[XXV] (SSH) per permettere anche un controllo in remoto dei dispositivi, fare diagnostica, ed eseguire aggiornamenti.

Riportiamo di seguito esclusivamente la tabella delle caratteristiche fisiche, perché collegate in uno degli aspetti (ossia la temperatura) ad un approfondimento nel 3°capitolo.

Feature	Description
Temperature	• Operating: 32 to 104°F (0 to 40°C). • Storage: -40 to 158°F (-40 to 70°C)
Altitude	• Operating: 10,000 feet • Storage: 40,000 feet
Humidity	• Operating: 8 to 80% non-condensing • Storage: 5 to 90% non-condensing
Vibration	• Operating: .25G, 5 to 300 Hz 15 minutes • Storage: 0.5G, 5 to 300 Hz 15 minutes
Power	• 90 to 264 VAC auto-ranging, 47 to 63 Hz • Maximum power dissipation less than 60W

[XXII] CLI: *command line interface* è la modalità di interazione tra utente ed elaboratore che avviene inviando comandi tramite tastiera e ricevendo risposte alle elaborazioni tramite testo scritto.

[XXIII] GUI: *graphical user interface*, è un paradigma di sviluppo che mira a consentire all'utente di interagire con il computer manipolando graficamente degli oggetti.

[XXIV] Telnet: è un protocollo di rete utilizzato su Internet. È utilizzato per fornire all'utente sessioni di login remoto di tipo linea di comando tra host su internet.

[XXV] Secure Shell: o terminale, è un protocollo che permette di stabilire una sessione remota cifrata ad interfaccia a linea di comando con un altro host.

2.9.2 La figura del Subnet Manager

Ogni sottorete possiede almeno un Subnet Manager (SM)[34]. Ogni SM risiede in una porta di un CA, di un router, o di uno switch e può essere implementato sia in hardware sia in software. Quando ci sono più SMs in uno sottorete, solo una tra questi dovrà essere il master, i rimanenti saranno posti in standby. La scelta di ENEA di posizionare il SM su uno degli switch (in particolare su uno dei Cisco SFS 7000d, in quanto unici switch a supportare implementazione in hardware) è per lo più dipesa dalla scelta di avere una rete ad albero, ossia dove vi sono degli elementi centrali che smistano il traffico dati. Per logica, posizionare l'attore che gestisce l'intera rete in un punto centralizzato è sembrata la più corretta, avendo la possibilità di raggiungere, da quella posizione, tutti gli altri attori della rete con lo stesso numero di hop indistintamente.

Il master SM è l'attore chiave di tutto il sistema, inizializza e configura la sottorete InfiniBand. Il SM è eletto nella parte di inizializzazione della rete, ed è responsabile di :

- Rilevare la topologia fisica della rete;
- Assegnare il Local Identifiers (LIDs) ad ogni endnodes, switch and routers;
- Stabilire i possibili paths tra gli endnodes;
- Scoprire cambi di topologia della rete e gestire l'aggiunta o la cancellazione dei nodi.

Ogni dispositivo della rete è anche un SMA, ossia un Subnet Manager Agent, componente che permette al dispositivo stesso di comunicare con il master SM. La comunicazione avviene tramite messaggi chiamati SMP (subnet manager packets).

Una volta che uno switch è eletto master, rimane tale fino a quando non occorre qualche problema relativo al suo funzionamento, come un guasto o una manutenzione programmata, ad esempio nel nostro caso, se c'è la necessità di effettuare un aggiornamento specifico del firmware[XXVI] della macchina stessa.

[XXVI] Firmware: indica il vero e proprio sistema operativo dell'apparato, che ne gestisce tutte le funzioni, possiede un'interfaccia utente spesso non banale (accessibile via porta seriale, o via rete con i protocolli SNMP, telnet, SSH, http, TFTP o anche FTP per il trasferimento di file di configurazione o nuove versione del firmware), permette di monitorare ed intervenire sul funzionamento dell'apparato e di modificarne la configurazione.

2.9.3 Secondo livello

La parte di rete più interessante è stata la parte intermedia di switching, ossia quella che connette gli switch del primo livello ai nodi di calcolo.

La struttura interna di questi 2 switch, da un lato il 144 porte a formare il cluster 1, dall'altro il 288 porte a formare il cluster 2, ha assunto una struttura interna particolare che descriveremo nel paragrafo seguente, dopo aver analizzato il principio di funzionamento degli apparati in questione e le principali caratteristiche tecniche.

SFS 7012 (144 porte) e SFS 7024 (28 porte)

I dispositivi switch Cisco SFS 7012D e 7024D [35] forniscono rispettivamente 144 e 288 porte 4X Infiniband non-blocking[XXVII]. Ogni porta può funzionare sia in Double Data Rate (DDR), con un data rate di 20 Gbps, oppure in Singel Data Rate (SDR) con un data rate di 10 Gbps.

Entrambi gli switch, supportano la connettività con dispositivi hot-swap[XXVIII], componenti che sono l'ideale per la costruzione di cluster HPC di grandi dimensioni.

Questi dispositivi offrono un sofisticato sistema di gestione della rete, capacità che semplifica il monitoraggio, la diagnostica e la manutenzione per dare all'amministratore di rete sempre una visione di insieme riguardo l'andamento prestazionale.

Figura 14 - Front e Back plain del Cisco SFS 7012d

[XXVII] Non-blocking: è una forma di I/O processing che permette agli altri processi di continuare a lavorare anche durante la trasmissione dei dati.

[XXVIII] Hot-swap: con il termine hot-swap intendiamo una interfaccia che permette il collegamento e/o lo scollegamento di un dispositivo anche a sistema avviato.

Switch di questo tipo sono tipicamente dotati di un agente di gestione, ovvero un piccolo programma software che permette di controllarne e monitorarne il funzionamento. Questo è accessibile sia attraverso una porta seriale (gestione out-of-band) che attraverso la stessa rete ethernet (gestione in-band). In questo caso, all'agente di gestione viene normalmente assegnato un indirizzo IP, e questo risponde ai protocolli SNMP, telnet e/o ssh, http.

SDR - 24-Port Switch	Expected Error Rate Per Time Span				
BER Qualification	Second	Minute	Hour	Day	
1E-12	0	14	864	20,736	IBTA Standards
1E-15	0	0	1	21	Cisco Standards
DDR - 24-Port Switch	Expected Error Rate Per Time Span				
BER Qualification	Second	Minute	Hour	Day	
1E-12	0	29	1,728	41,472	IBTA Standards
1E-15	0	0	2	41	Cisco Standards

SDR - 288-Port Switch	Expected Error Rate Per Time Span				
BER Qualification	Second	Minute	Hour	Day	
1E-12	6	346	20,736	497,664	IBTA Standards
1E-15	0	0	21	498	Cisco Standards
DDR - 288-Port Switch	Expected Error Rate Per Time Span				
BER Qualification	Second	Minute	Hour	Day	
1E-12	12	691	41,472	995,328	IBTA Standards
1E-15	0	1	41	995	Cisco Standards

Figura 15 - Confronto del BER tra IBTA Standard e Cisco Standard

Possiamo vedere in figura 14 come grazie all'affidabilità del dispositivo in questione, la BER[XXIX] stimata dal costruttore sia ben al di sopra della normale aspettativa che la tecnologia InfiniBand fornisce. Questo particolare può far capire quanto l'azienda Cisco abbia investito nella tecnologia InfiniBand, e quanto voglia mantenere riservate le informazioni tecniche relative ai dispositivi proprietari.

Internamente, uno switch è costituito da una o più schede munite di porte. Ad ogni porta può essere connesso un nodo, che può essere una stazione, un altro switch, un hub o altro dispositivo di rete.

[XXIX] BER: Bit Error Ratio, è il rapporto tra i bit non ricevuti correttamente e i bit trasmessi. Evidenzia quanto della originaria trasmissione viene perso o giunge distorto all'apparecchio ricevente a causa, ad esempio, di disturbi nel canale di trasmissione, di problemi degli impianti, di malformazioni originarie del flusso dati

Quando un nodo *A* cerca di comunicare con un nodo *B*, il comportamento dello switch dipende dalla *scheda* cui è collegato *B*:

- se *B* è collegato ad una porta sulla stessa scheda cui è collegato *A*, la scheda stessa inoltra i frame in arrivo su tale porta;
- se *B* è collegato ad una scheda diversa da quella cui è collegato *A*, la scheda invia i frame ad un canale di trasmissione interno detto *backplane*, caratterizzato da elevata velocità (tipicamente sull'ordine di molti Gbps), che provvede a consegnare il frame alla scheda giusta.

L'intelligenza necessaria a gestire le porte di ingresso e di uscita (riconoscere i frame, estrarre gli indirizzi sorgente e destinazione, trasmetterli) è distribuita sulle singole schede.
È importante descrivere quindi non solo le connessioni esterne del dispositivo, ossia le 144 o 288 porte, ma anche i collegamenti interni che permettono il forwarding dei pacchetti.

Spine e Leaf

Facciamo quindi una distinzione tra Spine e Leaf, ossia tra connessioni interne ed esterne del sistema:

- Una Spine identifica un collegamento interno tra port dello stesso dispositivo.
- Una Leaf identifica un gruppo di 12 porte esterne e di 12 porte interne.

Le prime connettono tra di loro vari Leaf, creando la struttura interna che descriverò in seguito; le seconde possono essere usate per connettere lo switch sia a dispositivi simili o nodi di calcolo, sia a dispositivi di storage. I collegamenti interni sono molto importanti per uno switch in quanto possono pregiudicare l'andamento dell'intero sistema a causa di un errato bilanciamento del carico e delle risorse impiegate.
Per collegamento, intenderemo la connessione fisica, sia essa interna al backplain sia esterna con cavo IB, tra due ports identificati univocamente con il LID.

Il 144 e il 288 porte possiedono rispettivamente 12 e 24 leaf con 12 porte interne e 12 esterne ognuno. Per le Spine invece, rispettivamente 3*2 (3 spine chip A e 3 spine chip B) per il 144 porte e 6*2 (6 spine chip A e 6 spine chip B) per il 288 porte.
Questo concorre a strutturare il secondo livello dell'architettura in questo modo (facendo riferimento al 144 porte, analogo per struttura al 288 porte):

La struttura dello switch può essere schematizzata in tre strati, il primo composto da un unico Lid connesso direttamente agli switch del primo livello da un lato e in backplain ad altri Lid, il secondo strato composto da 6 Lid che connettono internamente il primo e il terzo strato, a sua volta composto da 11 Lid ai quali si connettono direttamente i vari nodi di calcolo.
I tre dispositivi switch 7000d con funzionalità di SM sia che siano master o standby sono connessi univocamente al Lid 6 del dispositivo a 144 porte. Per ottimizzare il bilanciamento del carico, la scelta fatta è stata quella di non connettere un solo cavo IB tra questi dispositivi, ma ben 4 cavi IB a questo livello, così da avere 20*4 Gbps tra il primo e il secondo livello(strato 1) dell'architettura.
I Lid che fanno parte del secondo strato sono: 8, 9, 10, 11, 20, 23 e sono connessi con il Lid 6 attraverso 2 connessioni 20*2 Gbps sempre per bilanciare il carico dei dati.
È importante sottolineare la presenza di una differenza sostanziale tra 2 di questi Lid, ossia l'8 e il 9, rispetto agli altri, ossia il settaggio della switch port 0 come "enhanced" e non come "base" rispetto a Lid analoghi dello stesso strato. Argomento che verrà approfondito sottoparagrafo 3.2.2.
I Lid del secondo strato sono connessi a loro volta con altri 11 Lid che compongono il terzo strato di switching.
I Lid che fanno parte del terzo strato sono: 30, 31, 32, 34, 36, 38, 45, 48, 50, 56, 397 e sono connessi direttamente ai nodi di calcolo del cluster 1.

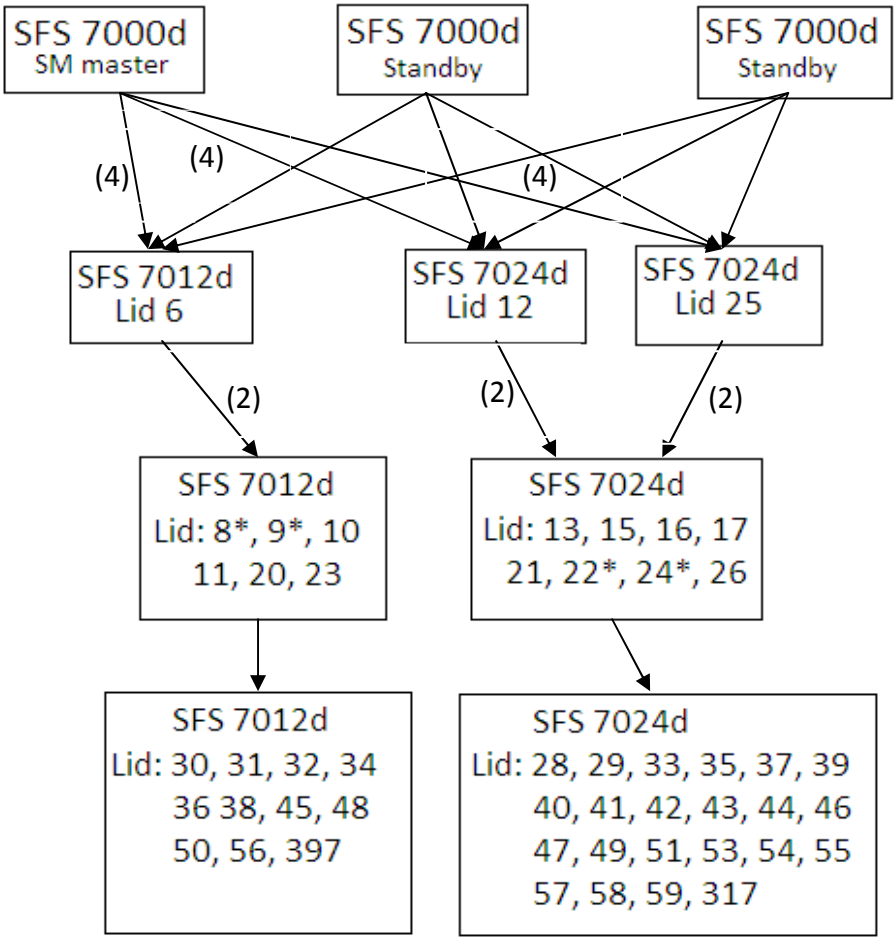

Figura 15 - Collegamenti tra I e II livello (I, II e III strato).

Capitolo 3. L'ottimizzazione del sistema CRESCO

L'obbiettivo primo del lavoro svolto sul sistema HCP CRESCO è stato quello di trovare delle soluzioni di ottimizzazione che portassero la rete ad avere le prestazioni nominali. L'approccio seguito è stato quello di incentrare l'attenzione sull'infrastruttura di rete InfiniBand, responsabile delle interconnessioni tra i dispositivi di calcolo.

La motivazione principale di questo approccio nasce dalla necessità di trovare soluzione al caso di job che terminano in condizione di errore per problemi di time-out imputabili alla rete InfiniBand.

Una attenzione particolare è stata data al dispositivo chiave dell'intera architettura, ossia al Subnet Manager Master, descritto nella sezione 2.9.2, il quale ha il compito di monitorare l'intero sistema, e prevenire l'incorrere di problematiche di questo tipo.

3.1 Comandi IB per il monitoring

L'infrastruttura InfiniBand mette a disposizione dell'utente, ma in particolar modo ad amministratori di rete, una quantità elevata di comandi per compiere operazioni di diagnostica.

Per capire le politiche di forwarding dei dati è stato fondamentale ricreare l'architettura globale della rete con un livello di astrazione molto basso. Serviva un comando che permettesse di avere una descrizione completa, link per link, porta per porta, dell'intera rete. IBNETDISCOVER, ha fornito, ovviamente in forma analitica e non grafica, la struttura dell'intero sistema, e ha permesso di analizzare la struttura interna ed esterna dei componenti.

Ogni porta attiva del sistema ha un identificativo locale (Lid), attribuito dal SM che permane fino alla de attivazione della porta. Tramite questo identificativo, il comando ibnetdiscover, analizza i collegamenti di ogni singolo Lid, e ne riporta le connessioni.

Per avere un’idea più chiara del comando, riporteremo alcune righe dell’output, che verranno descritte in dettaglio.

```
vendid=0x5ad
devid=0xb924
sysimgguid=0x5ad03011df98a
switchguid=0x5ad00001df98a(5ad00001df98a)
Switch  24 "S-0005ad00001df98a"         # "Cisco Switch SFS7000D" enhanced port 0
lid 4 lmc 0
[24]     "S-0005ad000704340b"[9]          # "SFS-7012 GUID=0x0005ad0002043644 Leaf
10, Chip A" lid 6 4xDDR
[23]    "S-0005ad000704340b"[10]          # "SFS-7012 GUID=0x0005ad0002043644 Leaf
10, Chip A" lid 6 4xDDR
[22]     "S-0005ad0007043200"[1]          # "SFS-7024 GUID=0x0005ad000304320a Leaf
13, Chip A" lid 25 4xDDR
[21]     "S-0005ad00070431f0"[9]          # "SFS-7024 GUID=0x0005ad000304320a Leaf
11, Chip A" lid 12 4xDDR
[20]     "S-0005ad0007043200"[2]          # "SFS-7024 GUID=0x0005ad000304320a Leaf
13, Chip A" lid 25 4xDDR
[19]     "S-0005ad00070431f0"[12]         # "SFS-7024 GUID=0x0005ad000304320a Leaf
11, Chip A" lid 12 4xDDR
[18]     "S-0005ad00001df9a8"[14]          # "Cisco Switch SFS7000D" lid 2 4xDDR
[17]     "S-0005ad00001df9a8"[13]          # "Cisco Switch SFS7000D" lid 2 4xDDR
[12]     "S-0005ad000704340b"[8]          # "SFS-7012 GUID=0x0005ad0002043644 Leaf
10, Chip A" lid 6 4xDDR
[11]     "S-0005ad000704340b"[7]          # "SFS-7012 GUID=0x0005ad0002043644 Leaf
10, Chip A" lid 6 4xDDR
[2]      "S-0005ad00001dfec1"[2]           # "Cisco Switch SFS7000D" lid 7 4xSDR
[1]      "S-0005ad00001dfec1"[1]           # "Cisco Switch SFS7000D" lid 7 4xSDR


vendid=0x66a
devid=0xb924
sysimgguid=0x5ad090d04320a
switchguid=0x5ad0007043200(5ad0007043200)
Switch   24 "S-0005ad0007043200"         # "SFS-7024 GUID=0x0005ad000304320a Leaf
13, Chip A" base port 0 lid 25 lmc 0
[24]     "S-0005ad000604346c"[6]         # "SFS-7024 GUID=0x0005ad000304320a Spine
5, Chip A" lid 24 4xDDR
[22]     "S-0005ad000604311d"[12]        # "SFS-7024 GUID=0x0005ad000304320a Spine
4, Chip A" lid 26 4xDDR
[20]     "S-0005ad000604332c"[8]         # "SFS-7024 GUID=0x0005ad000304320a Spine
1, Chip A" lid 22 4xDDR
[19]     "S-0005ad000604312a"[4]         # "SFS-7024 GUID=0x0005ad000304320a Spine
3, Chip A" lid 21 4xDDR
[17]     "S-0005ad100604311d"[1]         # "SFS-7024 GUID=0x0005ad000304320a Spine
4, Chip B" lid 16 4xDDR
[16]     "S-0005ad100604346c"[7]         # "SFS-7024 GUID=0x0005ad000304320a Spine
5, Chip B" lid 17 4xDDR
[15]     "S-0005ad100604332c"[5]         # "SFS-7024 GUID=0x0005ad000304320a Spine
1, Chip B" lid 15 4xDDR
[14]     "S-0005ad100604312a"[9]         # "SFS-7024 GUID=0x0005ad000304320a Spine
3, Chip B" lid 13 4xDDR
[12]     "H-0002c9030001224c"[1](2c9030001224d)          # "cresco2x088 HCA-1" lid
132 4xDDR
[11]     "H-0002c90300010efc"[1](2c90300010efd)          # "cresco2x091 HCA-1" lid
129 4xDDR
[10]     "H-0002c90300012148"[1](2c90300012149)          # "cresco2x089 HCA-1" lid
134 4xDDR
[7]      "H-0002c90300011780"[1](2c90300011781)          # "cresco2x247 HCA-1" lid
161 4xDDR
[6]      "H-0002c903000117b8"[1](2c903000117b9)          # "cresco2x086 HCA-1" lid
131 4xDDR
```

```
[5]     "H-0002c90300012388"[1](2c90300012389)          # "cresco2x090 HCA-1" lid
135 4xDDR
[4]     "H-0002c90300012624"[1](2c90300012625)          # "cresco2x085 HCA-1" lid
139 4xDDR
[3]     "H-0002c90300012438"[1](2c90300012439)          # "cresco2x087 HCA-1" lid
140 4xDDR
[1]     "S-0005ad00001df98a"[22]                        # "Cisco Switch SFS7000D"
lid 4 4xSDR
[2]     "S-0005ad00001df98a"[20]                  # "Cisco Switch SFS7000D" lid 4
4xSDR

vendid=0x5ad
devid=0x6274
sysimgguid=0x5ad00000c16b7
caguid=0x5ad00000c16b4
Ca      1 "H-0005ad00000c16b4"          # "cresco1x013 HCA-1"
[1](5ad00000c16b5)          "S-0005ad0007043667"[12]  #  lid  352  lmc  0  "SFS-7012
GUID=0x0005ad0002043644 Leaf 4, Chip A" lid 56 4xDDR
```

Tavola 1 - Output del comando ibnetdiscover relativo ai dispositivi: SHS 7000d, SFS 7024 e HCA.

L’output del comando (tavola 1) descrive in modo dettagliato le singole connessioni di ogni dispositivo di rete. Analizza ogni singolo Lid e tutte le 24 porte del Lid stesso, sia direttamente connesse tramite cavo esterno sia di connessione in backplain.

Ad esempio, per gli switch Cisco SFS 7000d, descritti nel capitolo 2, vengono riportati gli identificativi locali e globali (lid e guid) e le connessioni con gli altri dispositivi, con numero di porta (da 1 a 24), nome del componente, Lid del componente e l’informazione sulla capacità del link, ossia se si sta sfruttando una connessione 4X DDR o 4X SDR.

Questa analisi ha permesso di comprendere tutte le connessioni fisiche della rete, ricavando la topologia completa dell’ infrastruttura di CRESCO.

3.2 Ottimizzazione del sistema CRESCO

Eseguita l’analisi del sistema HPC CRESCO, si è proceduto sviluppando diverse ipotesi di soluzioni a problematiche dovute all’infrastruttura IB di CRESCO.

3.2.1 Bilanciamento del carico

Una delle prime considerazioni fatte, analizzando la topologia della rete, è stata quella di verificare se il carico veniva distribuito equamente tra i due cluster che compongono CRESCO.

I link che connettono il primo livello architetturale al secondo livello architetturale fanno tutti riferimento al LID 6 per il cluster 1 e ai LID 12e 25 per il cluster 2. Questo è sembrato in una prima analisi un eccessivo sovraccarico degli switch SFS 7012 e SFS 7024 (che chiameremo in questa parte di tesi per praticità, rispettivamente 144 porte e 288 porte).

Con riferimento alla figura 15 possiamo notare invece come il bilanciamento del carico è stato ottenuto aumentando il numero di link tra il primo e il secondo livello, andando così a distribuire il carico in modo performante sia sul Lid 6 che sui Lid 12 e 25. Infatti l’architettura progettuale ha previsto 4 link tra gli apparati in questione, che ripartiscono il carico in modo ottimale, senza incorrere in un collo di bottiglia a livello di switching.

3.2.2 Errori Interconnessioni IB

Appurato che la distribuzione del carico in funzione delle possibili richieste era stato ripartito in modo ottimale, abbiamo cercato di capire il perché alcuni job terminavano in condizione di errore.

Un approccio al problema essenzialmente pratico è stato quello di analizzare la messaggistica di errore ottenibile con il comando IBCHECKERRORS, che permette di visualizzare su schermo il superamento di determinate soglie di errore occorse sui componenti IB della rete, facendo riferimento a: tipologia di errore, Lid e port number dove si è verificato l’errore, e numero di messaggi che hanno superato la soglia.

Gia la presenza di una soglia, che è possibile modificare da linea di comando sui dispositivi di switching, ha fatto pensare alla predisposizione da parte di IB nell'incorrere ad una quantità di errori, dovuta magari all'elevata velocità di trasferimento o al numero eccessivo di messaggi di segnalazione che passano in rete, senza però interferire in modo determinante sulle prestazioni finali del sistema.

Lanciando il comando IBCHECKERRORS per la prima volta, ci siamo trovati di fronte ad un numero molto elevato di errori, dovuti al superamento delle soglie da parte dei counters che gestiscono questi errori.

L'utilizzo di tale comando ha permesso di identificare e riparare interfacce difettose sia su gli switch e su gli HCA.

Una volta eliminati i dispositivi in errore chiaramente identificati, l'analisi degli errori si è rivolta verso gli errori residui.

Analizzando le varie tipologie di errori, descritte nel capitolo 2, non siamo riusciti subito ad individuare la causa di questi ultimi errori sistematici, in quanto non vi era una connessione strettamente logica tra i vari errori, e in molte occasioni occorrevano degli errori multipli (ossia di diverso tipo) su una stessa porta relativa ad uno specifico Lid.

La presenza di soglie per il monitoring degli errori e quindi la presenza di contatori incrementali, ha senso se poi si ha la possibilità di cancellare il passato, ossia fare un clear dei contatori per controllare se le modifiche apportate al sistema abbiano un riscontro di funzionalità.

Vengono in aiuto, con questa base di ipotesi, altri due comandi di monitoring IB, da usare in concomitanza, ossia IBCLEARCOUNTERS e IBCLEARERRORS.

Il primo resetta i contatori di soglia, il secondo resetta gli errori occorsi.

L'output di questi due comandi, dava ripetutamente lo stesso tipo di errore, riportato in questa sezione per una analisi diretta.

```
[petricca@cresco2-f2 ~]$ sudo ibclearcounters
ibwarn: [9947] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
ibwarn: [9948] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
ibwarn: [9949] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
ibwarn: [9951] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
```

Tavola 2 - Output del comando ibclearcounters

Come possiamo vedere, il monitoring automatico di IB riporta 4 ibwarn, ossia 4 identici errori occorsi sull' infrastruttura, indicando come problematica **Node type not CA**.
Per sottolineare l'occorrenza di questi errori, riporteremo anche l'output del comando ibclearerrors:

```
[petricca@cresco2-f2 ~]$ sudo ibclearerrors
ibwarn: [10470] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
ibwarn: [10471] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
ibwarn: [10472] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
ibwarn: [10474] main: AllPortSelect not supported
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
```

Tavola 3 - Output del comando ibclearerrors

In questa prima fase di analisi, la presenza persistente di questi 4 errori sistematici, ha dato l'input per ulteriori controlli incrociati.
Avevamo trovato utilità in questa strategia, ossia ripulire i counters e gli errors, per vedere come si sarebbe comportato il monitoring di IB una volta rilanciato il comando IBCHECKERRORS.
Identificando con t_0 il tempo di lancio del comando ibcheckerrors (subito dopo aver prima resettato i contatori), e con t_i gli istanti successivi di lancio del comando ibcheckerrors, il controllo effettuato è stato quello di verificare in che modo gli errori occorrevano sul monitoring di IB.
Con questa operazione, siamo riusciti a classificare 2 tipologie di macro errori: una che dipende effettivamente dal mal funzionamento di qualche componente InfiniBand, l'altra tipologia è quella degli errori che si propagano. Lanciando il comando all'istante t_0 si visualizzavano una serie di errori su determinati port e su determinati LID. Rilanciando il comando dopo un periodo di circa 10 minuti, occorreva l'incremento sostanziale di errori su ulteriori port (oltre ai precedenti) e una costante presenza di errori su LID già precedentemente individuati. Insistendo con questo procedimento, nei 10 minuti successivi la situazione degli errori andava solo peggiorando.

Una ipotesi plausibile scaturita da questi eventi è stata quella di attribuire il sostanziale incremento di errori negli istanti ti alla impossibilità di compiere un clear totale dei counters, confermato dagli errori IB descritti in tabella 2 e 3.

Switch port 0: enhanced or base.

Gli switch con tecnologia InfiniBand integrata, in aggiunta alle normali porte esterne, hanno una porta speciale nominata switch port 0 o anche eventualmente management port.

Switch port 0 è una porta virtuale che non richiede di essere fisicamente presente, è esonerata da complicazioni dovute al rispetto delle specifiche del physical layer, ma è richiesto l'essere conforme con le specifiche del link layer.

Ci sono 2 sottotipi di switch port 0, in aggiunta alla base switch port0, c'è una opzionale enhanced switch port 0.

Enhanced switch port 0 fornisce una alternativa architetturale per aggiungere una porta switch o una porta CA, integrata nel sistema o come componente separato.

Uno switch che ha settata la switch port 0 enhanced può essere considerato un CA, ossia un punto terminale della comunicazione tra 2 link.

Settare una switch port 0 come enhanced vuol dire attribuire al nodo funzionalità di TCA (Target Channel Adapter), ossia permettere alla switch port 0 di eseguire funzionalità "enhanced".

Andando a ricontrollare quali dei Lid della rete avessero settato la switch port 0 come enhanced, si è scoperta una sospetta relazione tra questo particolare settaggio, gli errori che occorrevano sistematicamente dopo il reset dei counters, e il monitoring degli errori con il comando ibcheckerrors.

Tra le switch port 0 settate enhanced, vi erano sia i 5 switch Cisco SFS 7000d che altri 4 Lid settati in questo modo. La verifica successiva è stata quella di controllare se i Lid con la switch port 0 enhanced fossero proprio i Lid che occorrevano in errore con il comando ibcheckerrors.

```
[petricca@cresco2-f2 ~]$ sudo ibcheckerrors
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port all:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 9 port 24
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 24:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 9 port 22
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 22:
FAILED
#warn: counter RcvSwRelayErrors = 11172         (threshold 100) lid 9 port 20
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 20:
FAILED
#warn: counter RcvSwRelayErrors = 3334  (threshold 100) lid 9 port 19
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 19:
FAILED
#warn: counter RcvSwRelayErrors = 4411  (threshold 100) lid 9 port 16
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 16:
FAILED
#warn: counter RcvSwRelayErrors = 131   (threshold 100) lid 9 port 15
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 15:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 9 port 8
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 8:
FAILED
#warn: counter RcvSwRelayErrors = 2903  (threshold 100) lid 9 port 5
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 5:
FAILED
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port all:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 8 port 14
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 14:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 8 port 13
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 13:
FAILED
#warn: counter RcvSwRelayErrors = 37403         (threshold 100) lid 8 port 1
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 1:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 8 port 18
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 18:
FAILED
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port
all:  FAILED
#warn: counter RcvRemotePhysErrors = 2501       (threshold 100) lid 24 port 22
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 22:
FAILED
#warn: counter XmtDiscards = 2423       (threshold 100) lid 24 port 19
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 19:
FAILED
#warn: counter XmtDiscards = 313        (threshold 100) lid 24 port 13
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 13:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 24 port 3
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 3:
FAILED
#warn: counter RcvSwRelayErrors = 251   (threshold 100) lid 24 port 2
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 2:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 24 port 1
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 1:
FAILED
#warn: counter RcvSwRelayErrors = 225   (threshold 100) lid 24 port 6
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 6:
FAILED
```

```
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port
all:  FAILED
#warn: counter RcvRemotePhysErrors = 129        (threshold 100) lid 22 port 24
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 24:
FAILED
#warn: counter XmtDiscards = 497        (threshold 100) lid 22 port 15
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 15:
FAILED
#warn: counter XmtDiscards = 1161       (threshold 100) lid 22 port 13
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 13:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 22 port 9
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 9:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 22 port 2
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 2:
FAILED
#warn: counter RcvSwRelayErrors = 214   (threshold 100) lid 22 port 8
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 8:
FAILED

## Summary: 372 nodes checked, 0 bad nodes found
##          1378 ports checked, 25 ports have errors beyond threshold
```

Tavola 4 - Output del comando ibcheckerrors

Come si evince dal report degli errori (tavola 4), anche dopo aver resettato i contatori che gestiscono gli alarm di soglia, permangono 25 errori su altrettante porte dovuti al superamento delle soglie. I LID che occorrono in questi errori sono quattro. Ossia i LID 9, 8, 24, 22. LID che per altro sono gli unici ad avere settato la switch port 0 come enhanced.

Si nota inoltre che gli errori provocati da questi Lid non potevano essere cancellati dai comandi di clear.

A questo livello possono essere fatte solo ipotesi sull'origine di tali errori che si ripetono sistematicamente:

- Una ipotesi è che lo stato di port enhanced dipenda da una mal configurazione del sistema
- Un'altra ipotesi, sempre plausibile, coinvolge la capacità del firmware di gestire correttamente i port enhanced.

Andando a leggere lo standard InfiniBand abbiamo potuto verificare esclusivamente le differenze sostanziali tra una switch port 0 settato enhanced o base, e non il perché i progettisti di CRESCO hanno scelto di settare sui 4 Lid in questione la switch port 0 come enhanced.

Tramite queste informazioni però, si è potuto formulare una richiesta di risoluzione del problema specifica verso i responsabili della manutenzione e controllo del sistema CRESCO. È stata quindi aperto un ticketing verso l'help desk di Cisco dove si richiede esplicitamente la risoluzione di questa problematica individuata.

3.2.3 Tool ibChkErrs.php

Avere uno strumento in grado di eseguire dei comandi di diagnostica in pipe e con un parsing ordinato, permette all'amministratore di velocizzare le operazioni di monitoring della rete, e rendere queste stesse informazioni il più leggibili possibili.

È stato realizzato un tool che gestisse gli errori InfiniBand, dando la priorità ad una facile lettura dell'output. L'approccio eseguito è stato segnato comunque dall'esperienza avuta utilizzando i comandi IB di diagnostica, e in particolare ibclearcounters, ibclearerrors e ibcheckerrors.

L'output del comando ibcheckerrors, riporta un elenco di errori che vengono serializzati in linee di testo di non immediata comprensione in quanto le informazioni chiave (come Lid, port e device) sono distanziati da altri caratteri inutili al fine della lettura.

Nel caso specifico, come descritto nella sezione 3.2.2, il problema della sistematicità di alcuni errori presenti ancora su CRESCO, non permette di decifrare in modo corretto le informazioni dell'output del solo comando ibcheckerrors. Questo deriva dall'impossibilità da parte del gestore della sezione IB di rimuovere gli errori generati dai quattro Lid che hanno settato la loro switch port 0 come enhanced. Gli errori che vengono generati successivamente ad un clear dei contatori possono dipendere dalla sistematicità degli errori dovuti ai suddetti Lid.

Per aggirare questo problema, e visualizzare correttamente solo gli errori reali del sistema, si è inserito nel tool la possibilità di eseguire un clear degli errori e dei counters che permettesse di eliminare gli errori sistematici.

Senza ricorrere a questa attenzione l'amministratore di rete rileverebbe errori InfiniBand anche su Lid che nelle nostre ipotesi di sezione 3.2.2 segnalerebbero errori conseguenti agli errori critici già verificati.

In previsione della risoluzione delle problematiche dovute ai quattro Lid con switch port 0 enhanced, è stata inserita una opzione nel codice che permettesse all’amministratore di decidere se effettuare o no il clear degli errori già esistenti.
Il tool realizzato permette di avere un elenco ordinato per tipologia di errore occorso, specificando Lid, port, device, Spine o Leaf.
Il vantaggio oggettivo che abbiamo ottenuto è stato quello di avere uno strumento pratico e pulito in grado di individuare errori di connessione InfiniBand o guasti relativi a schede InfiniBand.
Riportiamo in Appendice I, il tool realizzato, l’output del tool, l’output del comando ibcheckerrors per un confronto soggettivo dell’utilità del programma

3.2.4 La temperatura

L’opportunità avuta di studiare un sistema così complesso come CRESCO, ha permesso di spaziare su vari aspetti che caratterizzano tecnologie innovative come questa. Un aspetto che spesso si sottovaluta, ma di particolare importanza è il concetto di temperatura di un sistema.
Il problema della temperatura di un sistema di calcolo distribuito o di un qualsiasi sistema informatico, è solitamente controllato da dei sensori interni agli apparati, che permettono il monitoring della temperatura interna del sistema.
Il sistema HPC CRESCO impiega a pieno carico 150 kilowatt di potenza, che vengono dissipati nella sala di calcolo, la cui temperatura è mantenuta a livelli accettabili da un sistema dedicato di raffreddamento.
Tale sistema di raffreddamento del è svolto da degli armadi appositi, chiamati *chiller,* che soffiano aria fredda sotto la pavimentazione della sala di calcolo. Tramite delle apposite griglie, l’aria fredda viene spinta sul lato esterno del corridoio dove sono locati i racks, e permette al sistema di mantenere la temperatura delle macchine stabile e adeguata.
Nel caso di malfunzionamento del sistema di raffreddamento si è osservato un incremento del numero di schede IB guaste, diminuendo l’affidabilità dell’intera infrastruttura.

3.2.5 Osservazione collegamenti SDR e DDR.

La tecnologia InfiniBand supporta sia dispositivi DDR sia SDR. Si è osservato un comportamento, potenzialmente anomalo, che vede, a fronte di una presenza quasi totale di dispositivi che supportano connessioni 4X DDR, un transfer-rate in up-link sempre limitato a Single Data Rate.

Il comando di diagnostica InfiniBand IBNETDISCOVER, descrive quale velocità di banda è applicata per ogni link che connette due dispositivi, sia essi connessi esternamente con cavo IB, sia con connessione interna nello stesso dispositivo.

Disegnando e mettendo particolare attenzione alle connessioni dell’intera rete, si evince la seguente cosa:

Dividendo il flusso in due rami, ossia quello che dai nodi di calcolo arriva al master (che chiameremo UP) e quello che dal master arriva ai nodi di calcolo (che chiameremo DOWN) si nota come il traffico DOWN sia tutto 4x DDR, mentre il traffico in UP sia tutto 4xSDR.

L’idea iniziale è stata quella di capire se alcuni apparati connessi alla rete, come ad esempio i nodi Cell (con trasmissione dati 4xSDR), andassero a condizionare le prestazioni dell’intera rete InfiniBand.

Pur avendo capito che l’architettura InfiniBand poteva adattarsi a diverse capacità di banda e quindi far cooperare SDR con DDR, si è provato a scollegare i dispositivi che impongono alla rete esclusivamente velocità 4xSDR per verificare se effettivamente i collegamenti UP rimanessero ancora 4xSDR.

Questo tentativo, non ha prodotto gli effetti sperati, mantenendo le capacità delle connessioni inalterate. Possiamo quindi ipotizzare che la tecnologia InfiniBand impone queste caratteristiche al sistema, oppure, che ci sono dei parametri sul set dei dispositivi che alterano le capacità dei link in UP.

Questo aspetto sarà comunque approfondito in un secondo momento, e per mancanza di tempo non inserito in questo elaborato.

3.3 Prestazioni CRESCO e Top 500

Il sistema HPC CRESCO è stato istallato nel maggio 2008 dall'azienda ComputerVar[36] in collaborazione con IBM e della stessa Cisco che ha fornito gli apparati di rete.

Attraverso il benchmark HPL (compilato con altri tool di monitoring con gcc 4.1.2, Goto-BLAS 1.24 e con libreria OpenMPI 1.2.5) si sono riscontrate a pieno carico delle performance che hanno portato ad un risultato, espresso in flops (ossia operazioni in virgola mobile), di 17.1 TFlops; con l'utilizzazione di 2500 cores, con 1250 processi ognuno avente due threads, usando 15GB dei possibili 16 a disposizione nel caso di N=720000.

Durante il run, entrambe le sezioni di calcolo sono state utilizzate.

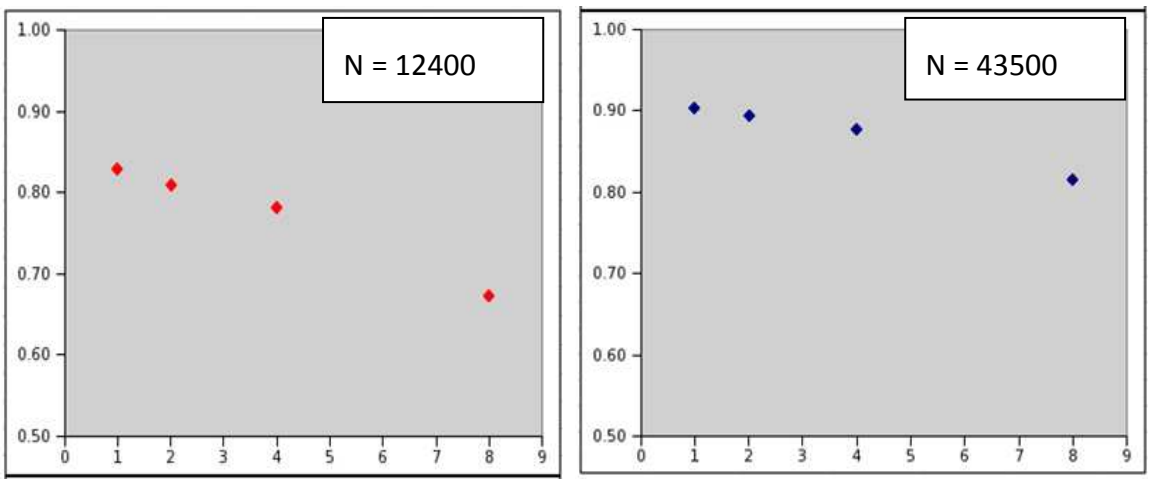

Figura 16 – Risultato test HPL su singolo blade.

La figura 16 mostra l'efficienza della CPU tramite il benchmark HPL fatto girare su un singolo blade, dual socket Xeon E5345, in due casi applicativi con differente dimensione del problema: N=12400 e N=43500, contro il numero di threads per un massimo di 8 (1 thread/core).

Conclusioni

Il lavoro di tesi svolto ha analizzato l'infrastruttura della rete ad alta banda e bassa latenza su architettura InfiniBand utilizzata da sistema di calcolo ad alte prestazioni CRESCO ed ha contribuito ad ottimizzare il suo funzionamento.

In particolare i comandi di diagnostica e messaggistica di errori che lo stack software di InfiniBand mette a disposizione permettono, con attenta analisi, di configurare e mantenere il sistema ad uno stato operativo di funzionamento ottimale. È emerso tuttavia, come alcune tipologie di errori, dovuti plausibilmente ad operazioni di configurazione non appropriate o al malfunzionamento del firmware degli apparati possano incrementare notevolmente l'incorrere di errori relativi alle interconnessioni tra nodi di calcolo.

Durante la fase di analisi sono emerse poi anche le difficoltà oggettive che un amministratore di rete trova nel gestire un sistema HPC con interconnessioni InfiniBand. Tali difficoltà nascono innanzitutto da una manualistica disponibile poco adeguata a fornire le informazione ad un livello di dettaglio tale da permettere operazioni che possano ottimizzare la rete stessa.
D'altro canto, la possibilità di aprire chiamate verso la casa produttrice degli apparati (Cisco), ha permesso di incrementare il "know-how" relativo ad aspetti sia tecnici sia propriamente di evoluzione prestazionale della rete.
In particolare, si è giunti alla conclusione che un sistema di calcolo ad alte prestazioni come CRESCO ha bisogno di un continuo monitoring della sua rete InfiniBand, per evidenziare le problematiche tali da perturbare il buon funzionamento dell'infrastruttura.
A tal fine è stato sviluppato uno script di test che fornisce dati sintetici sullo stato degli errori correnti prodotti sulla infrastruttura InfiniBand.
Tale script è stato messo a disposizione degli amministratori di sistema.

Appendice I

L'appendice riporta lo script ibChkErr, il suo modo di utilizzo ed un esempio dei risultati ottenibile dal tool realizzato in linguaggio PHP (PHP Hypertext Preprocessor) (tavola 5).

Tool IbChkErr

```
#!/usr/bin/php
<?php

function bubbleSort( &$items ) {
        $temp = "";
        $size = count( $items );
        for( $i = 0; $i < $size-1; $i++ ) {
               for( $j = 0; $j < $size - 1 - $i; $j++ ) {
                      if( $items[$j+1][0] < $items[$j][0] ) {
                             $temp = $items[$j];
                             $items[$j] = $items[$j+1];
                             $items[$j+1] = $temp;
                      }
               }
        }
 }

 function printPrePostLine(){
                 echo "\n----------------------------\n";
        }
        function printUsage($pName, $param){
                echo "Unrecognized option $param\n";
                echo "Usage: $pName [-noclear]\n\n";
                echo "Now exiting.\n\n";
        }

        /*Global variables declaration*/
        $clearCounters="sudo ibclearcounters";
        $clearErrors="sudo ibclearerrors";
        $checkErrors=" sudo ibcheckerrors -nocolor";

        $errorLineStart="#";
        $errorNotWarn="Error ";
        $errorCresco="cresco";
        $errorCisco="(cisco";
        $errorArray=array();


        $clearCommand="$clearCounters&&$clearErrors";
        $finalCommand=$checkErrors;

        printPrePostLine();
        /*Processing options*/
        if($argc>1){
                if($argv[1]=="-noclear"){
                        echo "Checking for new errors\n\n";
                        $execChecking=exec($finalCommand, $output);
                }
```

```
                else{
                        printUsage($argv[0], $argv[1]);
                        return;
                }
        }
        else{

                echo "\nClearing previous errors\n\n";
                $execClearing=exec($clearCommand, $outClear);
                echo "Checking for new errors\n\n";
                $execChecking=exec($finalCommand, $output);
        }



        $errorCount=count($output);
        $arrayCount=0;

        #Parsing #warn line
        #format:
        # #warn: counter SymbolErrors = 65535     (threshold 10) lid 8 port 13

        for($i=0; $i<$errorCount-3; $i++){
                if(substr($output[$i],0, 1)==$errorLineStart){
                        $eWarn=explode(" ", $output[$i]);
                        $i++;
                        while(substr($output[$i],0, 1)!=$errorLineStart){

             /*Parsing subsequent lines
              *Format:
              *
              *Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2,
              Chip A) port 5: FAILED
              *perfquery: iberror: failed: smp query nodeinfo: Node type not CA
              *Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1,
              *Chip A) port all: FAILED
              */

if(!strncmp($output[$i],$errorNotWarn,strlen($errorNotWarn)))
{$eError=explode(" ", $output[$i]);
        if(!strncasecmp($eError[5],     $errorCresco,     strlen($errorCresco))){
        $errStr="$eError[7] $eError[8] - $eError[5] - $eError[6]";
        $errorArray[$arrayCount][0]=$eWarn[2];
        $errorArray[$arrayCount][1]=$eError[4];
        $errorArray[$arrayCount][2]=$errStr;
                                        }
               else{
                      #Error check on lid 2 (Cisco Switch SFS7000D) port 19:
               ESC[1;031m FAILED
                      if(!strncasecmp($eError[5],$errorCisco,strlen($errorCisco)))
                     {
                             $errStr="$eError[8] $eError[9] - $eError[7]";
                             $errorArray[$arrayCount][0]=$eWarn[2];
                             $errorArray[$arrayCount][1]=$eError[4];
                             $errorArray[$arrayCount][2]=$errStr;
                                                }
                                 else{
$errStr="$eError[11] $eError[12] - $eError[5] - $eError[7] $eError[8]";
$errorArray[$arrayCount][0]=$eWarn[2];
$errorArray[$arrayCount][1]=$eError[4];
$errorArray[$arrayCount][2]=$errStr;
                                                }
                                        }
$arrayCount++;
                                 }
                                 $i++;
                        }
                        $i--;
```

```
        }
}

/*sorting*/
bubbleSort($errorArray);

/*displaying*/
$errorArrayCount=count($errorArray);
for($i=0; $i<$errorArrayCount; $i++){

        $errorType=$errorArray[$i][0];
        echo "+ $errorType\n";
        while($errorArray[$i][0]==$errorType){
                $lid=$errorArray[$i][1];
                $errString=$errorArray[$i][2];
                $errString=str_replace(",","",$errString);
                $errString=str_replace("(","",$errString);
                $errString=str_replace(")"," - ",$errString);
                $errString=str_replace(":","",$errString);
                echo"\t|_(lid $lid) - $errString\n";
                $i++;
                if($i>=$errorArrayCount)
                        break;
        }
        echo "\n";
        $i--;
}

printPrePostLine();

for($i=$errorCount-3; $i< $errorCount; $i++){
        echo "$output[$i]\n";
}
```

Tavola 5 - Script in php del tool ibChkErr

Per lanciare il tool bisogna posizionarsi su una delle macchine di CRESCO abilitate alla gestione del monitoring InfiniBand, ed editare la stringa ibChkErrs.php con l'indicazione del percorso dove è locato il file nella struttura AFS.

Output del tool ibChkErr

L'output che è possibile visualizzare è il seguente:

```
[petricca@cresco2-f2 ~]$ ./ibChkErrs.php
----------------------------
Clearing previous errors

Checking for new errors

+ RcvRemotePhysErrors
        |_(lid 24) - port 22 - SFS-7024 - Spine 5
        |_(lid 22) - port 24 - SFS-7024 - Spine 1

+ RcvSwRelayErrors
        |_(lid 9) - port 20 - SFS-7012 - Spine 2
        |_(lid 9) - port 19 - SFS-7012 - Spine 2
        |_(lid 9) - port 16 - SFS-7012 - Spine 2
        |_(lid 9) - port 15 - SFS-7012 - Spine 2
        |_(lid 9) - port 8 - SFS-7012 - Spine 2
        |_(lid 9) - port 5 - SFS-7012 - Spine 2
        |_(lid 8) - port all - SFS-7012 - Spine 1
        |_(lid 8) - port 1 - SFS-7012 - Spine 1
        |_(lid 8) - port 18 - SFS-7012 - Spine 1
        |_(lid 24) - port all - SFS-7024 - Spine 5
        |_(lid 24) - port 3 - SFS-7024 - Spine 5
        |_(lid 24) - port 2 - SFS-7024 - Spine 5
        |_(lid 24) - port 1 - SFS-7024 - Spine 5
        |_(lid 24) - port 6 - SFS-7024 - Spine 5
        |_(lid 22) - port all - SFS-7024 - Spine 1
        |_(lid 22) - port 9 - SFS-7024 - Spine 1
        |_(lid 22) - port 2 - SFS-7024 - Spine 1
        |_(lid 22) - port 8 - SFS-7024 - Spine 1

+ SymbolErrors
        |_(lid 2) - port all - SFS7000D -
        |_(lid 2) - port 19 - SFS7000D -
        |_(lid 9) - port all - SFS-7012 - Spine 2
        |_(lid 9) - port 24 - SFS-7012 - Spine 2
        |_(lid 9) - port 22 - SFS-7012 - Spine 2
        |_(lid 8) - port 14 - SFS-7012 - Spine 1
        |_(lid 8) - port 13 - SFS-7012 - Spine 1

+ XmtDiscards
        |_(lid 24) - port 19 - SFS-7024 - Spine 5
        |_(lid 24) - port 13 - SFS-7024 - Spine 5
        |_(lid 22) - port 15 - SFS-7024 - Spine 1
        |_(lid 22) - port 13 - SFS-7024 - Spine 1

----------------------------
## Summary: 373 nodes checked, 0 bad nodes found
##          1380 ports checked, 26 ports have errors beyond threshold
```

Tavola 6 - Output del tool ibChkErr.

Prima di analizzare l'output del tool (tavola 6), riporteremo l'output del commando ibcheckerrors (tavola 7), per poter poi fare un confronto maggiormente approfondito.

```
[petricca@cresco2-f2 ~]$ sudo ibcheckerrors
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port all:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 9 port 24
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 24:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 9 port 22
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 22:
FAILED
#warn: counter RcvSwRelayErrors = 12566         (threshold 100) lid 9 port 20
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 20:
FAILED
#warn: counter RcvSwRelayErrors = 3624  (threshold 100) lid 9 port 19
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 19:
FAILED
#warn: counter RcvSwRelayErrors = 4796  (threshold 100) lid 9 port 16
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 16:
FAILED
#warn: counter RcvSwRelayErrors = 134   (threshold 100) lid 9 port 15
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 15:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 9 port 8
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 8:
FAILED
#warn: counter RcvSwRelayErrors = 3127  (threshold 100) lid 9 port 5
Error check on lid 9 (SFS-7012 GUID=0x0005ad0002043644 Spine 2, Chip A) port 5:
FAILED
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port all:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 8 port 14
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 14:
FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 8 port 13
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 13:
FAILED
#warn: counter RcvSwRelayErrors = 41181         (threshold 100) lid 8 port 1
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 1:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 8 port 18
Error check on lid 8 (SFS-7012 GUID=0x0005ad0002043644 Spine 1, Chip A) port 18:
FAILED
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port
all:  FAILED
#warn: counter RcvRemotePhysErrors = 2501       (threshold 100) lid 24 port 22
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 22:
FAILED
#warn: counter XmtDiscards = 2423       (threshold 100) lid 24 port 19
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 19:
FAILED
#warn: counter XmtDiscards = 313        (threshold 100) lid 24 port 13
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 13:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 24 port 3
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 3:
FAILED
#warn: counter RcvSwRelayErrors = 300   (threshold 100) lid 24 port 2
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 2:
FAILED
```

```
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 24 port 1
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 1:
FAILED
#warn: counter RcvSwRelayErrors = 225   (threshold 100) lid 24 port 6
Error check on lid 24 (SFS-7024 GUID=0x0005ad000304320a Spine 5, Chip A) port 6:
FAILED
perfquery: iberror: failed: smp query nodeinfo: Node type not CA
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port
all:  FAILED
#warn: counter RcvRemotePhysErrors = 129        (threshold 100) lid 22 port 24
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 24:
FAILED
#warn: counter XmtDiscards = 497        (threshold 100) lid 22 port 15
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 15:
FAILED
#warn: counter XmtDiscards = 1161       (threshold 100) lid 22 port 13
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 13:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 22 port 9
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 9:
FAILED
#warn: counter RcvSwRelayErrors = 65535         (threshold 100) lid 22 port 2
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 2:
FAILED
#warn: counter RcvSwRelayErrors = 230   (threshold 100) lid 22 port 8
Error check on lid 22 (SFS-7024 GUID=0x0005ad000304320a Spine 1, Chip A) port 8:
FAILED

## Summary: 373 nodes checked, 0 bad nodes found
##          1380 ports checked, 25 ports have errors beyond threshold
```

Tavola 7 - Output del comando ibcheckerrors

È evidente come il tool realizzato semplifichi l'output del commando, eliminando le informazioni inutile o ripetitive che caratterizzano la maggior parte dei comandi di monitoring di qualsiasi sistema informatico.

Aver scelto di elencare gli errori per "tipologia di errore" ci permette prima di tutto di capire la gravità del problema, ossia se occorre un intervento immediato sull'infrastruttura, e poi risulta più pratico per l'amministratore risolvere lo stesso problema per tutti i Lid coinvolti, per poi passare ad analizzare gli ulteriori tipi di errore.

Gli errori in questione vengono detti anche "non-critical errors" in quanto non portano gravi conseguenze all'intero sistema. Sono comunque errori che devono essere risolti perché possono andare ad inficiare sulle prestazioni generali del sistema HPC.

Bibliografia

[1] ENEA - www.enea.it

[2] CRESCO - www.cresco.enea.it.

[3] InfiniBand - http://www.infinibandta.org/home; InfiniBandTM Architecture Specification Volume 1 Release 1.2.1

[4] Myrinet - www.myri.com.

[5] Quadrics - http://www.quadrics.com

[6] Computational science - http://www.computationalscience.org/.

[7] Top 500 supercomputer sites - http://www.top500.org.

[8] Classificazione di Flynn - http://dida.fauser.edu/sistemi/sistem5/arch_p.htm.

[9] Architett. di Von Neumann - http://it.wikipedia.org/wiki/ArchitetturadivonNeumann.

[10] Legge di Amdahl - http://it.wikipedia.org/wiki/Legge_di_Amdahl.

[11] Linpack - http://www.netlib.org/benchmark/linpackjava/.

[12]IBMbenchmark - www.ibm.com/systems/management/director/about/director52/extensions/powerexec.html

[13] HPL - http://www.netlib.org/benchmark/hpl/.

[14] RedHat - www.redhat.it/.

[15] www.platform.com

[16] Open AFS - http://www.openafs.org/.

[17] GPFS - Schmuck, Frank; Roger Haskin (January 2002). "GPFS: A Shared-Disk File System for Large Computing Clusters" (pdf).Proceedings of the FAST'02 Conference on File and Storage Technologies: 231-244, Monterey, California, USA: USENIX. Retrieved on2008; http://www-03.ibm.com/systems/clusters/software/gpfs/.

[18] LUSTRE - http://wiki.lustre.org/index.php?title=Main_Page.

[19] Needham-Schroeder http://it.wikipedia.org/wiki/Protocollo_di_Needham-Schroeder.

[20] PCI - http://it.wikipedia.org/wiki/Peripheral_Component_Interconnect.

[21] DDR SDRAM - http://it.wikipedia.org/wiki/DDR_SDRAM.

[22] SPD - SERIAL PRESENCE DETECT STANDARD, General Standard JEDEC Standard No. 21–C - http://en.wikipedia.org/wiki/Serial_Presence_Detect.

[23] PCI Express - http://it.wikipedia.org/wiki/PCI_Express.

[24] IPoverIB - RFC 4755 del 2006

[25] SDP - http://en.wikipedia.org/wiki/Sockets_Direct_Protocol.

[26] SRP - http://en.wikipedia.org/wiki/SCSI_RDMA_Protocol.

[27] uDALP – User Direct Access Programming Library Version 1.0 Date: june 21, 2002 - www.datcollaborative.org/uDAPL_doc_062102.pdf

[28] OpenMPI - http://en.wikipedia.org/wiki/Openmpi.

[29] RDMA - http://en.wikipedia.org/wiki/Rdma; http://www.rdmaconsortium.org/home.

[30] ICT - http://en.wikipedia.org/wiki/Information_and_Communication_Technology.

[31] Mirinet - http://en.wikipedia.org/wiki/Myrinet.

[32] International Supercomputing Conference, Dresden 17-19/6/2008, Architecture and performances of the CRESCO HPC system. S. Migliori, G. Bracco, P. D'Angelo, D. Giammattei, M. De Rosa, S. Pierattini, S. Podda, A. Quintiliani, S. Raia, A. Funel, G. Richelli (*) ENEA-FIM, L.Tevere Thaon di Revel (Roma) Italy (*) IBM Italia http://www.cresco.enea.it/Documenti/web/?id=presentazioni/Poster_E-Science_2008/.

[33] Cisco SFS 7000D InfiniBand Server Switch - http://www.cisco.com/en/US/prod/collateral/ps6418/ps6419/ps6421/product_data_sheet0900aecd804c2272.html

[34]Subnet Manager - Cisco High-Performance Subnet Manager for InfiniBand Server Switches Release 1.1; http://www.cisco.com/en/US/products/ps6985/.

[35] Cisco SFS 7012D e 7024D InfiniBand Server Switches http://www.cisco.com/en/US/prod/collateral/ps6418/ps6419/ps6421/ps6987/product_data_sheet0900aecd804c323b_ps6421_Products_Data_Sheet.html.