La comunicazione nell'ambito di un cluster Beowulf

Introduzione

Si descriverà in questo documento l'utilizzo di cluster di computer per risolvere alcuni problemi che necessitano di ingenti risorse di calcolo e si definirà cosa si intende solitamente per Cluster.
Analizzando i cluster di classe Beowulf si illustreranno le problematiche legate alla loro realizzazione con particolare attenzione a come avviene la comunicazione fra i vari membri del Cluster e quali sono le difficoltà di progettazione ed i benefici prestazionali offerti dall'approccio distribuito alla computazione.
Non fa parte degli scopi di questo testo descrivere in dettaglio la computazione parallela ed il software di supporto ad essa, ci si concentrerà principalmente sulla descrizione dell'hardware utilizzato nei Beowulf e delle possibili reti da essi adottate descrivendone topologia, cablaggio ed apparati attivi.

Cosa è un Cluster di calcolatori ?

Definire una volta per tutte cos'è un cluster di computer non è un'impresa semplice, vi possono essere diverse accezioni del termine in dipendenza dalle esigenze e dalle aspettative di chi li progetta e di chi ne fa uso, nel nostro caso si parlerà di cluster in presenza di un gruppo di calcolatori aventi le seguenti caratteristiche:

- Consiste di molteplici macchine uguali o con caratteristiche simili (di solito esiste una macchina più potente delle altre che funge da frontend, con una connessione alla rete pubblica);
- Vi è una stretta comunicazione fra i membri del cluster (che da quest o momento in poi chiameremo nodi) mediante connessioni di rete dedicate (Beowulf) o non dedicate nel caso dei cosiddetti COW, Cluster Of Workstation, oppure NOW, Network Of Workstation;
- Tutte le macchine condividono risorse (per esempio un filesystem remoto mediante NFS);
- Il software di ogni nodo consente la collaborazione fra i nodi, solitamente facendo uso di librerie per lo scambio di messaggi (MPI ovvero Message Passing Interface) o di altre forme di IPC (Inter-Process Communication).

Data questa definizione, è utile spiegare il motivo per cui il *cluster computing* ha riscosso un grande successo negli ultimi anni fornendo qualche nozione storica che aiuterà anche ad inquadrare la situazione attuale.

Cos'è un cluster Beowulf ?

Lo scopo del progetto Beowulf è quello di studiare il potenziale dei cluster di Personal Computer per calcoli assai complessi. Beowulf si ispira all'idea di una Pila di PC (PoPC, Pile of PC) distaccandosi da quella di Cluster o Network di Workstation (COW/NOW) secondo la quale i membri del cluster concedono solo i cicli di inattività alla computazione distribuita.
Beowulf pone l'enfasi sull'utilizzo di hardware facilmente reperibile, processori dedicati e sull'utilizzo di una rete di comunicazione privata.
Uno degli scopi principali del progetto Beowulf è di raggiungere la migliore proporzione possibile fra prezzo ed efficienza.

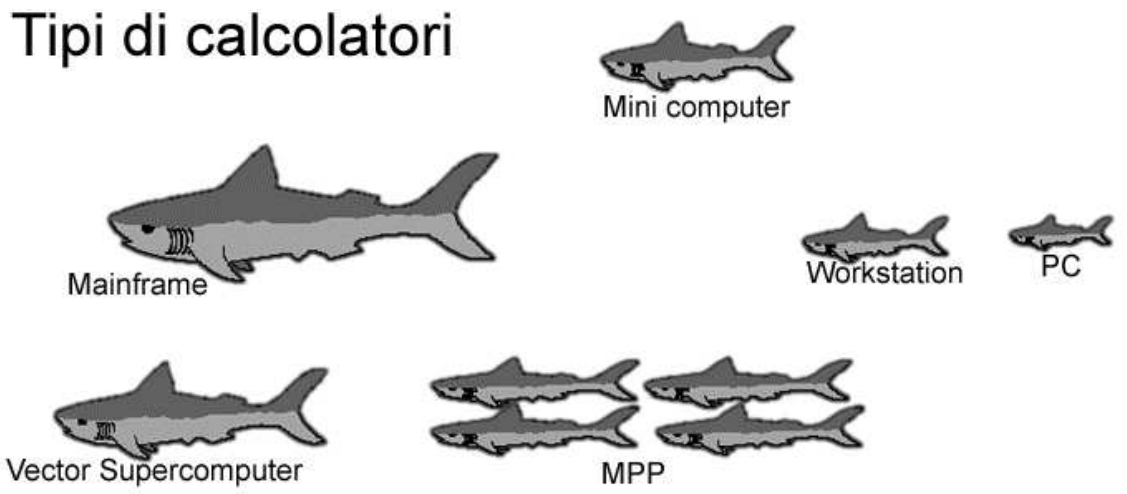
Software di Sistema in un Cluster Beowulf

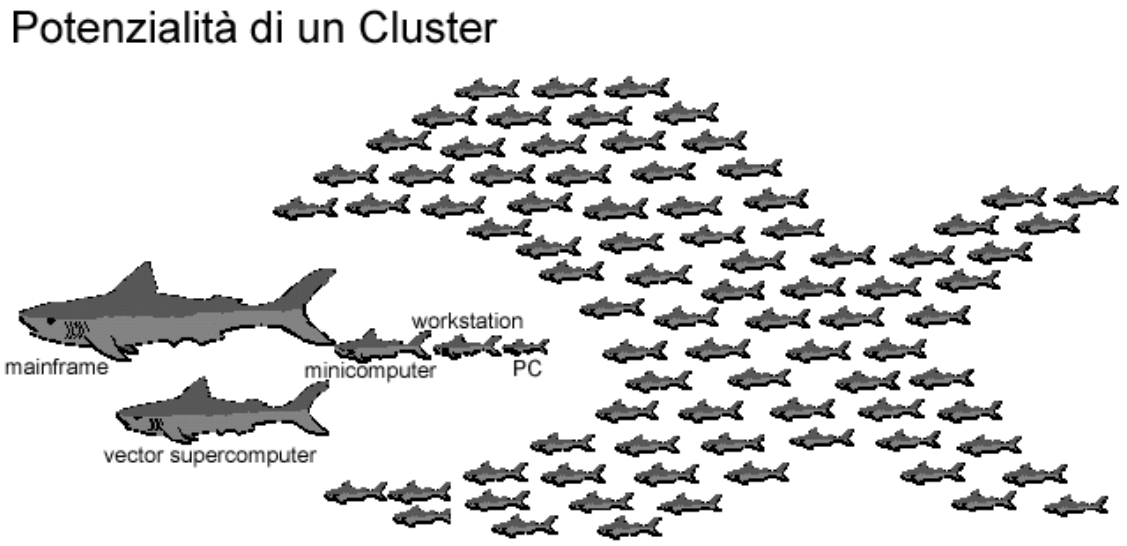
Il software necessario ad un cluster Beowulf è un insieme piuttosto vasto di soluzioni combinate. Tipicamente si utilizza Linux quale sistema operativo poiché è OpenSource, supporta una gran varietà di dispositivi hardware e per la sua semplicità di gestione. Un cluster Beowulf fa uso di MPI oppure di librerie PVM (Parallel Virtual Machine), che consentono all'utente di scrivere programmi usando il passaggio di messaggi per gestire il parallelismo, ma vengono utilizzati anche i classici sistemi di IPC sullo stile di SystemV oppure le librerie per il multithreading. Il passaggio di messaggi costituis ce ad ogni modo il cuore della "parallelizzazione" del calcolo.
La comunicazione fra i processori in Beowulf avviene mediante TCP/IP sulla rete dedicata e l'efficienza di quest'ultima gioca un ruolo fondamentale nelle prestazioni finali del cluster, risultando talvolta il collo di bottiglia che impedisce di fruttare il massimo potenziale di ogni nodo.
Beowulf è stato usato anche per lo studio di fattibilità dell'impiego di più reti locali "in parallelo" per ridurre i tempi di latenza e per aumentare il troughput, come vedremo successivamente.
Questa classe di cluster ha anche portato la comunità Linux ad estendere il kernel del sistema operativo introducendo il concetto di GPID (Global Process ID), ovvero di un identificatore di processo unico per tutto il cluster grazie alla comunicazione kernel-to-kernel fra i vari nodi; altra modalità per individuare univocamente un processo nel cluster fa uso di librerie esterne, l'identificatore è detto in questo caso GPID-PVM per mettere in risalto la compatibilità c on PVM e con il suo PID tipico (PVM Task ID).

Le grandi capacità di calcolo offerte da un Beowulf possono essere messe al servizio di applicazioni di astronomia, geologia, metereologia, fisica o chimica ed in generale di applicazioni di simulazione; applicazioni per le quali un cluster si rivela una buona scelta, presentandosi la necessità di elaborare enormi quantità di numeri (*number crunching applications*) e non si vogliono o non si possono spendere cifre esorbitanti per i Supercalcolatori paralleli classici (come per esempio i calcolatori Cray).

Perchè utilizzare un cluster di computer ?

Per rispondere alla domanda si può evidenziare in primo luogo l'ottimo rapporto prezzo/prestazioni offerto da questo tipo di soluzione.
Esistono diverse categorie di cluster, poiché in questo testo si discute principalmente dei cluster Beowulf si mostreranno i vantaggi dei cluster di Personal Computer evidenziando come si possano raggiungere risultati estremamente soddisfacenti con hardware poco costoso e di facile reperibilità.
Si deve tenere presente, tuttavia, la dipendenza delle prestazioni dal tipo di applicazione: nel caso in cui il calcolo possa essere partizionato in attività indipendenti per poi ricombinarne i risultati (*divide et impera*) la convenienza di un Beowulf è difficile da mettere in discussione, per contro se il calcolo richiede prevalentemente l'utilizzo delle connessioni di rete, altre architetture potrebbero svolgere il compito in maniera più efficiente.
Una illustrazione può aiutare nella comprensione delle potenzialità di questo tipo di computazione distribuita:





Breve storia del progetto Beowulf

Il progetto Beowulf è una creatura di Donald Becker un ricercatore della NASA; egli mostrò il primo cluster da 16 nodi in occasione del progetto ESS (Earth and Space Sciences project) nel 1994. Nei mesi successivi molti altri dipartimenti della società aerospaziale Americana si mostrarono interessati alla cosa ed in seguito vi fu un attento interesse anche da parte delle università statunitensi. Negli anni successivi il successo di Beowulf si allargò vistosamente e cluster di questa classe vengono attualmente utilizzati sempre più in ambiti commerciali, come nel campo delle analisi finanziarie.
La prima creatura di Becker era costituita da 16 macchine dotate di processori DX4 (una versione ibrida fra un 486 ed un Pentium) e da una rete Ethernet che faceva uso della tecnica del channel bonding (alla quale si accennerà in seguito); egli la battezzò Beowulf, ispirandosi al protagonista di un poema della letteratura scandinava che salvò i Danesi dal terribile mostro gigante Grendel.
Il successo del progetto è presumibilmente dovuto anche al fatto che il periodo della sua nascita fosse un "buon momento" della storia dei calcolatori elettronici, i prezzi dell'hardware *pc-style* cominciavano a scendere per via della concorrenza fra produttori e gli strumenti software cominciavano ad acquistare una certa maturità per il calcolo parallelo; come menzionato precedentemente il sistema operativo Linux contribuì notevolmente alla diffusione del progetto, sfruttando questa occasione anche come banco di prova per evidenziare la sua versatilità e flessibilità.

Uno dei primi cluster Beowulf denominato Wiglaf, dal nome di un altro personaggio del poema di cui sopra:

- 16 Processori DX4 100 Mhz;
- 256 K di cache secondaria per ogni nodo;
- 16M di memoria centrale per ogni processore;
- Ogni nodo ha un harddisk da 540M o 1G;
- Tre interfacce di rete Ethernet per ogni nodo, con l'utilizzo del channel bonding e della duplicazione degli indirizzi hardware.

Si noti l'utilizzo di più interfacce di rete su ogni nodo, ciò si rese necessario in quanto i processori erano troppo veloci per una sigola interfaccia Ethernet a 10Mbit/s e gli switch Ethernet erano ancora piuttosto costosi nel 1994; la soluzione adottata permise di realizzare un cluster *ben bilanciato*.

Presente e futuro del progetto Beowulf

Dal 1994 la potenza di calcolo dei processori per PC è aumentata ed il loro costo è calato notevolmente, ciò rende l'opzione Beowulf ancora più appetibile per chi ha particolari esigenze di calcolo, ma introduce problematiche nella progettazione e realizzazione dei cluster legate soprattutto all'eliminazione dei colli di bottiglia che non permetterebbero un uso intenso delle risorse a disposizione. La velocità delle memorie di massa o la latenza delle rete dedicata ai cluster possono vanificare l'obiettivo di ottenere una macchina macina-numeri economicamente conveniente, in poche parole per ottenere un cluster *ben bilanciato* è necessario un minimo di progettazione per quanto concerne la componentistica e l'infrastruttura di comunicazione.
Per quanto riguarda l'ottimizzazione delle memorie di massa si possono ottenere prestazioni soddisfacenti adottando soluzioni su bus SCSI ed eventualmente configurazioni RAID 0 o 0+1.
Per l'ottimizzazione della rete di comunicazione, aspetto sul quale ci concentreremo maggiormente, vi sono principalmente due strategie non esclusive che si prefiggono scopi equivalenti:

- Contenere i tempi di latenza passando a reti più evolute, ciò spinge ad adottare GigabitEthernet per la rete privata oppure architetture innovative quali Myrinet;
- Utilizzare topologie alternative con le quali realizzare la rete dedicata del cluster, sono esempi di questo approccio l'uso di Fat Tree Network o dela più recente idea Flat Neighborhood Network descritta successivamente.

Altro aspetto da considerare nella costruzione di un Beowulf è la disposizione fisica dei suoi nodi, in generale è possibile utilizzare gli chassis standard del tipo tower o middletower (per poter ospitare molteplici interfacce di rete) oppure dei rack da 19"; in ogni caso è necessaria una buona ventilazione del locale che ospita il cluster.

Ecco una immagine di uno dei più grandi cluster Beowulf al mondo, si trova nell'università di Stanford, negli USA ed è utilizzato per ricerche sugli algoritmi genetici. Ha circa 1000 nodi assemblati con hardware convenzionale ed adotta una rete Fast Ethernet che fa uso combinato di switch ed hub. L'applicazione specifica non richiede tempi di latenza critici, ma principalmente forza bruta nell'affrontare la computazione.
Macchine simili sono utilizzate anche per applicazioni di Data Mining e Knowledge Discovery, campi in cui la grande mole di dati da analizzare richiede ingenti risorse di calcolo e giustifica lo sforzo nella progettazione e nella realizzazione.

La comunicazione in un Beowulf

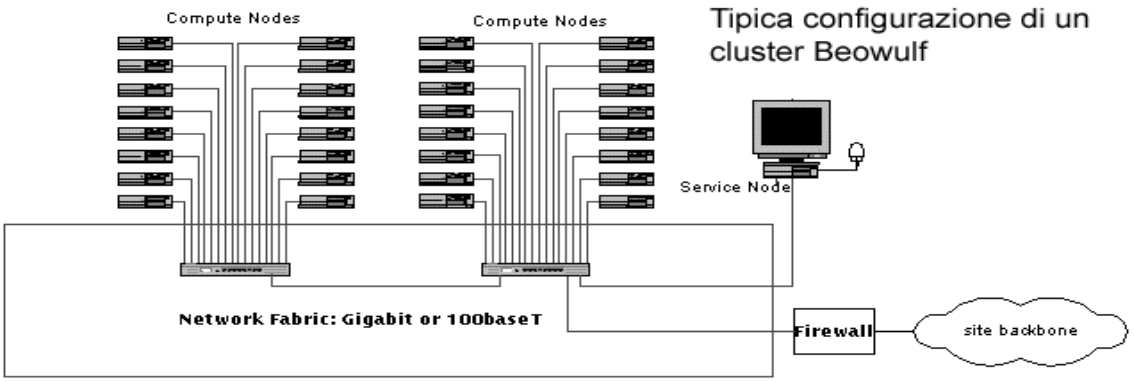
La struttura generale di un cluster Beowulf prevede che vi sia un nodo "master" che coordini il lavoro degli altri nodi e permetta all'utente di interfacciarsi al sistema, solitamente questa macchina ha accesso alla rete esterna (rete pubblica, aziendale o di ateneo) per fornire l'accesso da remoto.
Per l'amministrazione e la manutenzione è possibile collegare ogni nodo del cluster ad uno switch per monitor e tastiera oppure ad una rete di servizio, ma ciò che più conta per la buona salute del cluster è l'efficienza della rete dedicata allo scambio di dati e messaggi fra processori; in alcuni casi una rete a 100Mbit/s switched classica può risultare valida, ma vi sono applicazioni che esigono latenze talmente basse da non ritenere adeguata una soluzione standard.

Protocolli utilizzati nel cluster

Sulla rete privata del cluster i protocolli maggiormente utilizzati attualmente sono quelli della suite TCP/IP, si tende ad utilizzare la versione 4 del protocollo IP per la sua indiscutibile maturità (pur avendo essa dei difetti), ma la versione 6 potrà portare sicuri benefici alle reti per cluster, essa infatti offre funzionalità di auto-configurazione, controllo di flusso e consente l'utilizzo di pacchetti di grosse dimensioni detti *jumbograms* che potranno rendere la comunicazione internodo meno pesante, si noti inoltre che, per sua natura, la rete privata di un Beowulf non necessita di essere compatibile con vecchi sistemi di comunicazione che non supportano IPv6. Tuttavia per la transizione alla versione 6 sarà necessario che tutte le librerie di sistema e quelle per il Message Passing si integrino perfettamente con il nuovo protocollo.
Con la versione 4 di IP, l'auto-configurazione dei nodi del cluster avviene mediante BOOTP, un protocollo di bootstrap basato su UDP/IP, definito in RFC 951, che permette a stazioni non configurate (dette anche *stateless*), appartenenti ad una rete TCP/IP, di apprendere da un server remoto diverse informazioni quali il proprio indirizzo di rete, l'indirizzo del router di default ed il nome di un eventuale file da caricare in memoria ed eseguire. In un cluster Beowulf, BOOTP viene utilizzato per comunicare ai nodi unicamente la configurazione di rete, benchè esso sia ben più ricco di funzionalità; la richiesta inoltrata da un nodo "stateless" viene indirizzata al server bootp, qualora l'indirizzo di quest'ultimo non sia noto al client, esso spedisce un pacchetto in *broadcast* (all'indirizzo 255.255.255.255); questo indirizzo è da intendere come di "broadcast sul cavo locale" e non è valido nel caso in cui il bootp server appartenga ad una rete diversa da quella dei client (i nodi del cluster). Uno dei vantaggi offerti da BOOTP rispetto a protocolli più datati, quali RARP, è di essere indipendente dal livello fisico e datalink della rete (in riferimento al modello ISO/OSI) in modo da poter essere utilizzato su ogni stazione che supporti la Internet Protocollo Suite, senza alcuna necessità di modifiche al sistema operativo per avere accesso "raw" ai pacchetti.

La rete, il vero cuore del cluster

In un cluster vengono utilizzati numerosissimi servizi di rete, si è parlato di BOOTP, si è accennato ai sistemi di scambio dei messaggi quali MPI e PVM quali tecniche avanzate di comunicazione interprocesso, vi sono esigenze di scambio di dati fra i nodi sul filesystem di rete e vi sono spesso servizi per la gestione e la manutenzione dei nodi del cluster; una soluzione intelligente, qualora vi siano esigenze particolari, può rivelarsi quella di usare una rete diversa per diversi gruppi di servizi (per esempio una rete per il traffico di dati ed una per lo scambio di messaggi).
Le scelte da fare nella progettazione dell'infrastruttura di comunicazione vengono influenzate prevalentemente dalle necessità che si hanno: se si desidera un sottosistema di IPC a velocità elevata allora i costi aumentano, ma vi sono soluzioni estremamente valide fra cui le reti Myrinet; per il network I/O le reti switched di derivazione Ethernet (10/100/1000) risultano in genere una buona soluzione. Nella stragrande maggioranza dei casi si utilizza la rete IPC (o talvolta quella di gestione) anche per il network I/O.

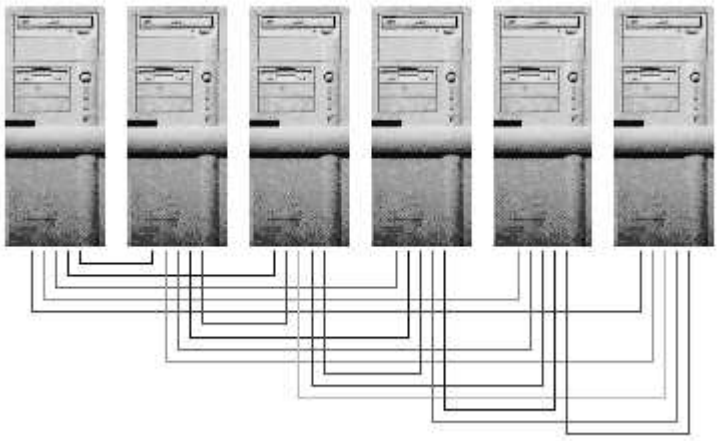


Per la gestione della rete (ove necessaria) di può ricorrere ad apparati attivi amministrabili (*managed*) e per la gestione degli host si utilizza tipicamente il protocollo SNMP.

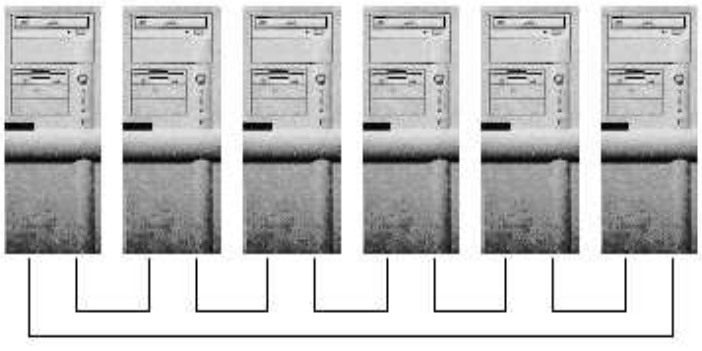
Un parametro importante per valutare l'efficienza di una rete dedicata di un cluster è la "Bisection Bandwidth" (in italiano: banda di bisezione) che valuta la connettività del cluster nel suo insieme, la Bisection Bandwidth teorica è calcolata considerando ogni macchina divisa in due, usando il partizionamento con la connettività pessima e sommando la banda dei collegamenti fra le due metà; misurare la bontà di questo parametro è importante per applicazioni che richiedono una comunicazione globale nell'intero cluster.
Altro fattore che impatta fortemente sulle prestazioni della rete è la latenza nel trasporto dei messaggi; TCP è un protocollo ben progettato, ma non si adatta in modo ideale al Message Passing, per ottenere risultati migliori è tuttavia necessario hardware dedicato e costoso, in lieve contrasto con la filosofia di Beowulf. Sono in fase di definizione anche altri standard per l'interconnesione, quali InfiniBand che si basa sull'idea di una *Virtual Interface Architecture* (VIA) che presenterà una interfaccia per le applicazioni indipendente dal tipo di rete.

Topologie di rete nei cluster Beowulf

Nell'affrontare la progettazione della rete privata del cluster ci si ritrova spesso a dover fare considerazioni di carattere economico ma con un occhio a quello che sarà il futuro lavoro di gestione e manutenzione.
La realizzazione del cluster non presenta difficoltà insormontabili, ma una installazione facilmente gestibile richiede una analisi attenta della struttura del cluster. La topologia che fornisce le prestazioni ottimali, non solo per un cluster, è quella totalmente magliata, come illustrato nell'immagine sottostante:

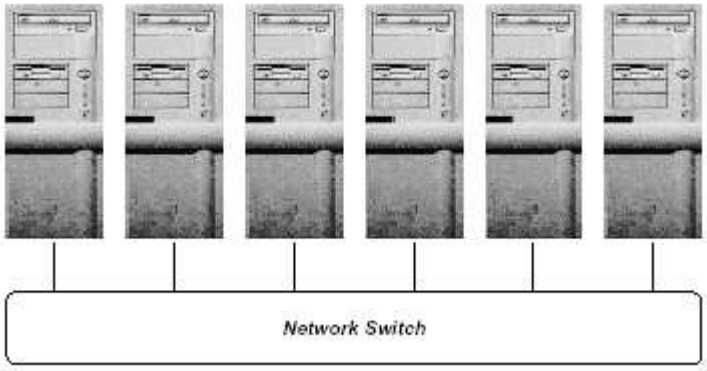


Una configurazione in cui ogni processore è collegato direttamente a tutti gli altri offre la massima banda e la minima latenza, ma sfortunatamente per un cluster di N nodi sono necessarie N-1 connessioni, quindi altrettante interfacce di rete; utilizzando una tipica scheda madre per PC il limite superiore per l'installazione di interfacce di rete è fra 4 e 6, e pur ipotizzando l'uso di schede multi-interfaccia la soluzione è da scartare per cluster di dimensioni medie e grandi. Se la comunicazione diretta fra i nodi non è possibile si deve quindi accettare un limite al numero di connessioni, e si deve prevedere una qualche forma di instradamento per garantire ad ogni nodo la raggiungibilità di ogni altro. Un'altra possibile struttura potrebbe consentire due interfacce per nodo ed essere configurata ad anello, nel modo seguente:

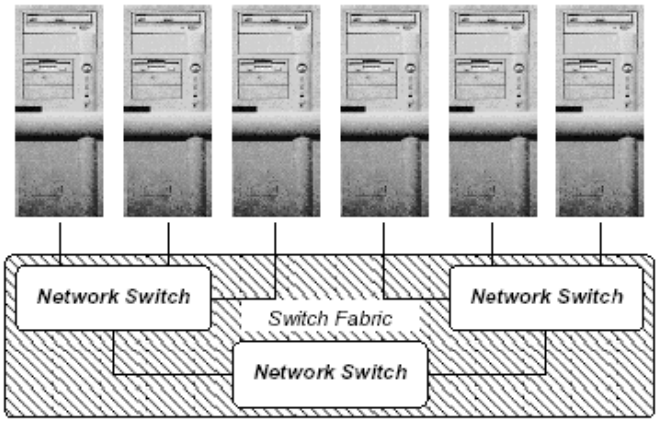


Una struttura di questo tipo risulta essere assai economica non avendo bisogno di switch e non ha limiti legati al numero di nodi; essa presenta tuttavia molti svantaggi quali il crescente tempo per gli instradamenti e la conseguente crescita della latenza.

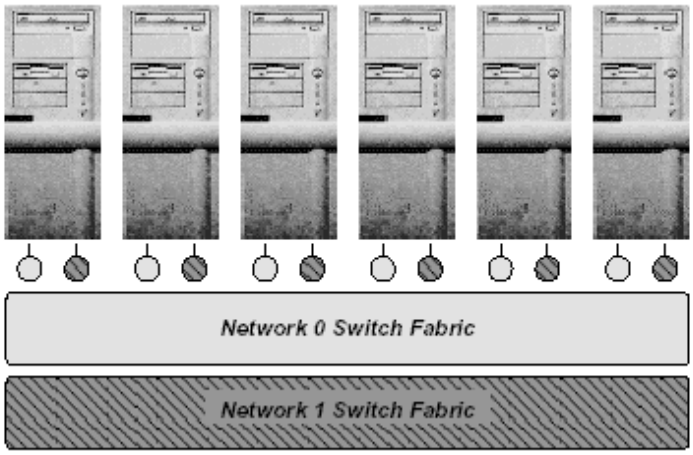
Una soluzione atta a minimizzare i tempi di latenza senza ricorrere alla connessione diretta è quella di dotare ogni nodo di una interfaccia di rete a banda larga e collegarlo ad uno switch che commuti opportunamente le comunicazioni; in questo caso ogni nodo sarà connesso direttamente al nodo con cui intende dialogare solo per il tempo effettivo necessario alla comunicazione, in particolare è possibile adottare tecnologie quali Gigabit Ethernet, Myrinet, GigaNet o la più comune Fast Ethernet, passeremo in rassegna queste diverse alternative dopo la discussione sulle topologie di rete. Una rete switched "ideale" è rappresentata nella figura sottostante:



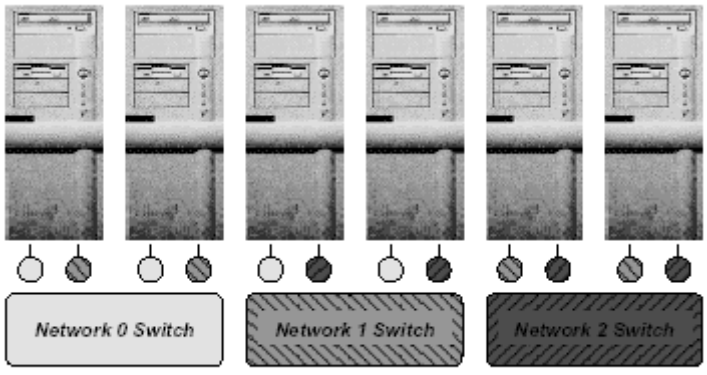
Ancora una volta, però, ci si trova in difficoltà nel caso di cluster di grosse dimensioni in quanto gli switch con un grande numero di porte hanno costi proibitivi, specialmente nel caso di reti ad altissima velocità e non si dimentichi che in questo ultimo caso una sola interfaccia di rete può arrivare a costare più del nodo che la ospiterà.
E' necessaria dunque una soluzione che mantenga i vantaggi di quella appena descritta, ma che cerchi di ovviare a quelli che sono i suoi limiti: costo elevato, e talvolta impossibilità di realizzazione (nel caso di un cluster con centinaia di nodi è impensabile sperare in un unico switch con diverse centinaia di porte).
L'approssimazione più vicina a quella con unico switch consiste nell'adottare una *switch fabric* (una tela di switch), una struttura gerarchica che necessita però di essere progettata; alcune reti da gigabit/s consentono infatti una maggiore flessibilità nella realizzazione della tela rispetto ad altre: Gigabit Ethernet per esempio consente solo topologie a stella gerarchica (quindi ad albero) mentre le reti Giganet (che implementano per la prima volta il concetto di Virtual Interface Architecture) permettono anche topologie "piu' dense" e più performanti tipicamente dette a *fat-tree.*



Nel caso si adotti una rete Fast Ethernet la strategia orientata alle *switch fabric* rende i tempi di latenza vicini a quelli che si avrebbero con un unico grande switch, in quanto i collegamenti di uplink fra gli switch hanno tipicamente velocità superiori (anche di un ordine di grandezza) a quelli delle schede di rete e non introducono ritardi catastrofici. Il problema in questo caso riguarda piuttosto la velocità delle interfacce dei nodi, infatti per molte applicazioni 100Mbit/s potrebbero non essere sufficienti, ma a questo problema si può rimediare utilizzando il channel bonding, come nel primo Beowulf. Questo modus operandi è largamente supportato da Linux; in breve: più schede di rete vengono viste dal nodo come un'unica interfaccia con velocità più elevata, in questo modo si risolve il problema della banda; ci si preclude, tuttavia, la possibilità di utilizzare le molteplici schede di rete per attuare connessioni contemporanee a più nodi diversi. L'immagine a lato mostra come realizzare l'infrastruttura per il *channel bonding*.

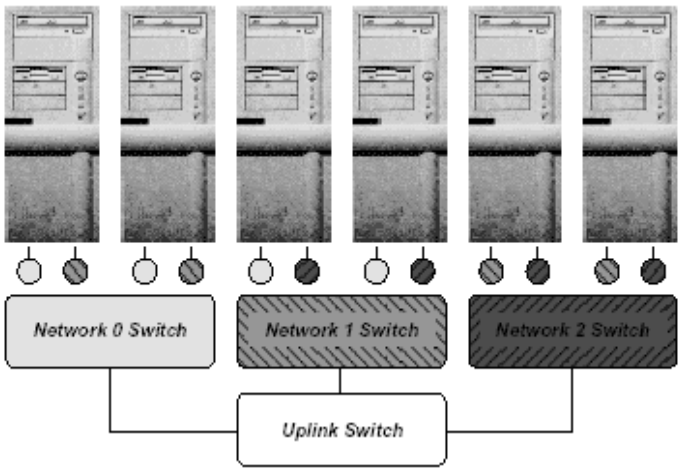


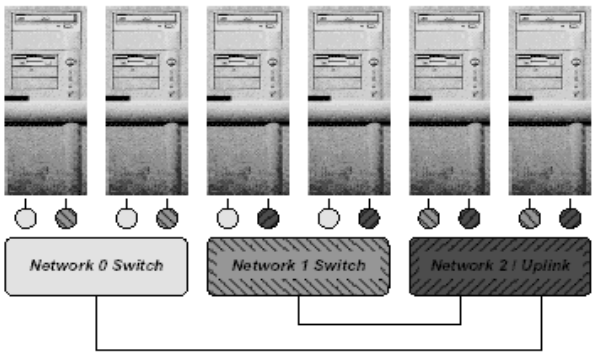
L'evoluzione del concetto di *switch fabric* che pone soluzione a tutti i problemi visti sinora è stata battezzata Flat Neighborhood Network, in questo caso gli switch sono connessi unicamente alle interfacce dei nodi e non agli altri switch e quindi i tempi di latenza sono mediamente contenuti, inoltre il costo di realizzazione è basso tenendo conto del fatto che si utilizzano switch con la stessa banda passante delle interfacce dei nodi. Tuttavia questa soluzione non è esente da difetti, infatti alcune coppie di nodi del cluster hanno solo una connessione a "interfaccia singola" con la minima latenza, pur potendo ottenere più banda solo con instradamenti più complessi. Il secondo problema è il routing, che diventa una questione complessa al crescere delle dimensioni del cluster. Si osservi la figura, le prime due macchine hanno due sottoreti in comune (neighborhoods) e possono comunicare fra di loro come se vi fosse il channel bonding, ma il bonding delle due interfacce della prima macchina non funzionerà nel mandare un messaggio alla terza macchina, poiché esse hanno una sola subnet comune (hanno un solo vicino comune).



Pur senza utilizzare l'equivalente del channel bonding, l'instradamento è complicato; nell'illustrazione precedente, la simmetria introdotta semplifica la cosa, ma ciò non è vero in generale per le FNN.

La definizione più generale di FNN fornita da coloro i quali hanno avuto l'idea è la seguente: una rete che adotta una topologia in cui tutte le comunicazioni punto-punto "importanti" (solitamente tutte, ma non necessariamente) sono implementate con una sola latenza di switch (*single switch latency*). Tuttavia è conveniente, in pratica, dotare la FNN di uno switch addizionale in uplink (come nell'immagine a destra) con gli switch della FNN poiché questo può fornire un multicast migliore ed un più efficiente I/O verso sistemi esterni (per esempio altri cluster). Questo switch di "secondo livello" può essere visto anche come un punto di connessione per eventuali nodi "di riserva".





Nel caso in cui uno degli switch della FNN abbia porte sufficienti a disposizione, è possibile "innestare" ad esso lo switch di uplink, si noti che nel caso si adottino switch con un gran numero di porte è sempre possibile progettare la rete in modo da riservare punti connessione sufficienti per l'uplink su uno degli switch della FNN.

Concludendo, una FNN è scalabile, fornisce senza sforzi servizi in multicast e I/O verso l'esterno e offre prestazioni elevate ad un costo più che ragionevole; essa necessita tuttavia di un routing gestito opportunamente e può eventualmente essere adattata alle esigenze specifiche dell'applicazione che farà uso del cluster. Lo studio del routingin una FNN è importante sia per la sua progettazione e realizzazione fisica, sia per ottenere le prestazioni migliori nell'utilizzo; questo documento non tratterà le tecniche per generare le tabelle di instradamento di una FNN, per ulteriori dettagli vedere i riferimenti bibliografici.

Il primo cluster ad adottare una topologia ad FNN è stato realizzato alla University of Kentucky utilizzando 66 nodi con processori a 700Mhz, 264 interfacce di rete a 100Mbit/s (4 per ogni nodo) e 9 switch.

Tecnologie di interconnessione

Dopo aver scelto le caratteristiche del nodo del cluster e durante la valutazione della topologia di rete da adottare è di fondamentale importanza decidere quale sottosistema di comunicazione (tecnologie di rete e protocolli) risulta più conveniente ricordando che esso sarà il "collante" che renderà un'insieme di PC un cluster Beowulf dalle magnifiche prestazioni.
Si tenga presente che uno degli scopi principali di un Beowulf è quello di ottenere il miglior rapporto prezzo/prestazioni; nonostante per applicazioni particolari si possa decidere di dare maggiore importanza alla componente prestazionale, in molti casi una soluzione volta ad ottimizzare i costi fornisce una potenza di calcolo accettabile; con l'aumentare della potenza dei processori per PC è, tuttavia, ragionevole rivolgere l'attenzione anche verso soluzioni meno diffuse (e sicuramente più costose) e più efficienti, per evitare che la rete sia il fattore limitante alla valorizzazione completa del cluster.
Dopo aver descritto brevemente le varie tecnologie attualmente disponibili si discuterà brevemente dell'efficienza dei protocolli utilizzati nelle reti per cluster ponendo l'accento sui protocolli di trasporto.

Fast Ethernet:

Le reti Fast Ethernet (802.3u), hanno velocità massima di 100Mbit/s e sono fra le più usate nei cluster di piccole e medie dimensioni, le interfacce di rete di questo tipo hanno praticamente "costo zero" nell'economia di realizzazione di un cluster ed anche gli apparati attivi hanno costi altamente convenienti. Inoltre questi ultimi hanno spesso porte di uplink con velocità nell'ordine del Gigabit e ciò consente la realizzazione di *switch fabric* senza l'introduzione di latenze dannose per la bontà della rete.

Gigabit Ethernet:

Evoluzione delle reti Fast Ethernet, le reti Gigabit (802.3z) hanno velocità massima di 1Gbit/s ed hanno il pregio di conservare la compatibilità con le reti 802.3 (tranne che in alcuni casi), esse utilizzano preferibilmente cablaggi in fibra ottica, ma è possibile realizzare cablaggi in rame anche se con limiti precisi sulla qualità dei cavi e sulla lunghezza dei collegamenti punto-punto. Le reti Gigabit introducono delle novità molto apprezzate nel campo del *cluster computing;* si citi in primo luogo l'adozione (opzionale) di frames di dimensione maggiore ai classici 1500 bytes, si possono avere frames fino a 9000 bytes, chiamati *jumboframes* pagando il prezzo di perdere la compatibilità con il passato; altre caratteristiche interessanti sono il controllo di flusso, e l'aggregazione degli interrupt che consente di formare gruppi di interruzioni per evitare all'host di gestire diversi eventi eccezionali in modo distinto. Nel prossimo futuro sarà inoltre possibile acquistare interfacce di rete capaci di fare e switch Gigabit ad un prezzo ragionevole, soprattutto se rapportato ai benefici in termini di efficienza che essa offre ed alle promesse in termini di affidabilità.

Myrinet:

Myrinet è una tecnologia proprietaria con velocità fino a 2 Gbit/s full-duplex tempi di latenza molto bassi e con monitoraggio continuo dei canali di comunicazione, essa consente cablaggi in fibra ottica ed offre buone caratteristiche di scalabilità nel caso di grossi cluster. Per sfruttare al massimo le sue potenzialità utilizza un sistema per lo scambio di messaggi ottimizzato chiamato GM message-passing system. Myrinet è una ottima soluzione per i cluster medio grandi soprattutto come infrastruttura per la parte MPI, e non è raro vedere cluster che adottano soluzioni ibride con Ethernet e Myrinet, come vedremo più avanti. Il software per il supporto di Myrinet è facilmente reperibile, ma il prezzo delle interfacce e degli apparati attivi non ne giustifica l'applicazione nel caso di piccoli cluster.

Giganet cLAN:

Giganet è la prima rete ad implementare lo standard Virtual Interface Architecture (VIA) e propone interfacce ad alte prestazioni studiate per esigenze di clustering, dando ad esse il nome di CLAN interfaces (interfacce per una "Cluster Area Network"). La banda ed i tempi di latenza sono simili a quelli di Myrinet ed anche il costo è agli stessi livelli. Precisamente Giganet consente di raggiungere i 3 Gbit/s, ma i prezzi sono proibitivi considerando anche il fatto che una tale banda non può essere utilizzata efficientemente dai normali PC (di cui un cluster Beowulf intende fare uso) a causa dei loro limiti architetturali (a livello di BUS). Anche gli apparati attivi hanno prezzi elevati e la scalabilità della rete diviene assai complessa nel caso di grandi cluster. Nella gran par te dei casi Giganet non si rivela conveniente per un cluster Beowulf, ma, come detto ripetutamente, applicazioni particolari potrebbero giustificare costi maggiori pur di ottenere tali prestazioni. Forse in futuro la disponibilità di apparati Giganet a prezzi ragionevoli sarà facilitata dal diffondersi dell'architettura VIA alla quale si è accennato in precedenza.

Nota sui protocolli di trasporto nei cluster Beowulf

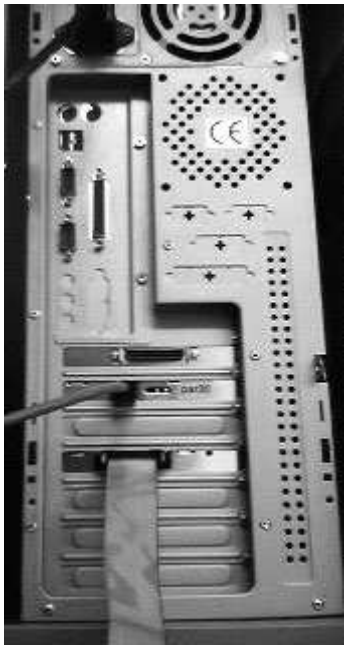
I cluster che fanno uso di sistemi MPI necessitano di un livello trasporto rapido ed efficiente e, data la piccola dimensione dei messaggi, una implementazione standard del protocollo di trasporto potrebbe non rivelarsi adeguata, si prenda ad esempio il TCP ed il suo utilizzo nei Beowulf, TCP aggrega tipicamente i piccoli messaggi per effettuare la spedizione in modo più efficiente, questo è un comportamento ragionevole per normali applicazioni, ma non è in genere il migliore per i cluster che necessitano di latenze molto basse nello scambio dei messaggi. Per ovviare a questo problema la comunità Linux ha prontamente fornito modifiche all'implementazione di tcp che consentissero una gestione più snella dello scambio di piccoli messaggi.

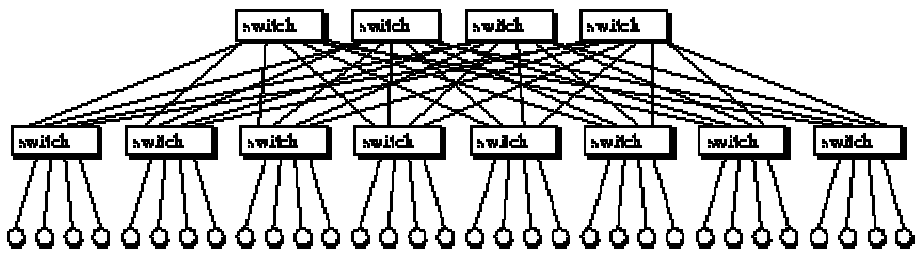
Esempi di Cluster Beowulf

Si è già parlato del primo Beowulf e dei primi esperimenti sulle FNN, si mostreranno ora ulteriori esempi di cluster, alcuni dei quali realizzati nelle università, se ne descriveranno le caratteristiche presentando delle immagini.

Parnass2

Un'esempio di cluster che utilizza molte delle tecnologie di cui si è parlato precedentemente si trova al dipartimento di Matematica ed Applicazioni dell'Università di Bonn, esso ha una rete Fast Ethernet per l'I/O generico e per lo scambio di dati e si appoggia ad una rete Myrinet per lo scambio di messaggi.
Ogni nodo di Parnass2 è basato su un PC con doppio processore Intel Pentium II ed usa il sistema operativo Linux. Vi sono 128 processori, la rete Fast Ethernet switched e la rete Myrinet configurata a fat-tree.
Eccone alcune immagini: nella prima si notano le due interfacce di rete (802.3u e Myrinet), la seconda mostra un dettaglio della disposizione dei nodi del cluster sugli scaffali, nell'ultima immagine è rappresentata la topologia a *fat-tree* utlizzata per la rete di Message-passing.





The Stone Soupercomputer

Un altro progetto assai interessante è quello iniziato da, Forrest M. Hoffman, William W. Hargrove, and Andrew J. Schultz; essi sono riusciti a realizzare un cluster beowulf a costo zero e di conseguenza (stando alle goliardiche formule degli autori) dalle prestazioni infinite:

Essi sostengono che le persone sono spesso interessate al rapporto prezzo-prestazioni per i loro calcolatori. Poiché il costo dello Stone SouperComputer è approssimativamente nullo, qualsiasi risultato in termini di prestazioni fa in modo che il rapporto cercato sia uguale a zero.

$$\frac{\text{Price}}{\text{Performance}} = \frac{\sim 0}{\text{anything}} \rightarrow 0$$

Il rapporto prestazioni-prezzo è perfino più interessante. Gli autori asseriscono che con qualsiasi risultato prestazionale il rapporto in esame tende all'infinito.

$$\frac{\text{Performance}}{\text{Price}} = \frac{\text{anything}}{\sim 0} \rightarrow \infty$$

Pur non volendo insistere troppo sulle formule precedenti, che pur formalmente valide sono volutamente provocatorie, si deve riconoscere che in generale l'approccio al supercalcolo con i Cluster Beowulf risulta estremamente conveniente. Lo Stone SouperComputer prende il nome da una minestra di verdure chiamata *Stone Soupe* nella quale è possibile trovare un gran numero di vegetali differenti, allo stesso modo il cluster di cui stiamo parlando ha un insieme altamente eterogeneo di PC che collaborano alla computazione; essi sono stati tutti donati ai realizzatori. Lo Stone SouperComputer è utilizzato per applicazioni di analisi del territorio.

A sinistra una immagine dello Stone Super Computer; sulla destra una tabella che mostra i vari tipi di nodi che lo costituiscono.

Nodes in Cluster Today
133
Pentium Nodes Today
53
Alpha Nodes Today
5

Beowulf per tutte le esigenze

Anubis

Anubis è il nuovo cluster realizzato dal Dipartimento di Fisica dell'Università di Pisa, esso fa uso di processori convenzionali, ma montati in moduli da 1 unità per rack 19", le sue caratteristiche principali sono: 14 nodi con doppio AMD Atlon MP a 1200Mhz e 13 nodi con doppio AMD Athlon MP 1800+, ogni nodo ha 1Gb di RAM ed utilizza il sistema operativo Linux redHat 7.1 e il software Mosix.

RedStone Personal Cluster

Redstone è un piccolo cluster basato su Linux con otto nodi con processori Intel Pentium III e switch Fast Ethernet integrato che si prefigge lo scopo della massima facilità di gestione per i piccoli laboratori che necessitano di discrete risorse di calcolo.

Riferimenti Bibliografici

Beowulf website - http://www.beowulf.org

Beowulf Underground - http://www.beowulf-underground.org

Linux HPC Clusters - **Dan Owsley**

Compiler Tecniques for Flat Neighbothood Networks - **Dietz and Mattox**
http://aggregate.org

Sommario

Introduzione 1

Cosa è un Cluster di calcolatori ? 1

Cos'è un cluster Beowulf ? 2

Software di Sistema in un Cluster Beowulf 2

Perchè utilizzare un cluster di computer ? 3

Breve storia del progetto Beowulf 4

Presente e futuro del progetto Beowulf 5

La comunicazione in un Beowulf 6

Protocolli utilizzati nel cluster 6

La rete, il vero cuore del cluster 7

Topologie di rete nei cluster Beowulf 8

Tecnologie di interconnessione 12

Nota sui protocolli di trasporto nei cluster Beowulf 13

Esempi di Cluster Beowulf 14

Beowulf per tutte le esigenze 16

Riferimenti Bibliografici 17