Il Laboratorio di Calcolo Distribuito dell'IAPS

Bruno Martino, Giorgio Patria

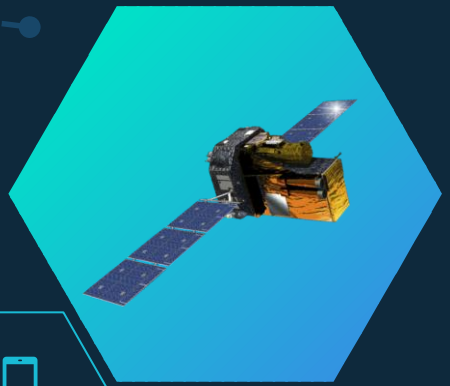
I am Bruno Martino

You can find me at: *bruno.martino@iasi.cnr.it*

Topics

- Aves
- Aves2
- GforC
- VirtLab
- MDbirs

Dal CED di Integral ad Aves



- Per permettere un efficace processamento dei dati prodotti dalla missione Integral (ott. 2002) viene allestito a cura di Memmo FEDERICI un centro di calcolo dedicato presso l'allora IAS
- Col passare del tempo I sistemi di calcolo basati su costose workstations evolvono verso soluzioni basate su comuni PC in ambiente LINUX
- Nell'anno 2008 Federici propone, per ridurre i tempi di calcolo delle analisi dei dati, di migrare il software su una nuova piattaforma basata su un cluster di PC

Il cluster Aves

- La prima versione di AVES (in ambiente Linux DEBIAN 5) è composta da sole 8 macchine quad core, 4Gb di RAM, dischi da 320Gb in condivisione MHDDFS, connettività interna a 1Gb; il gestore di risorse scelto e' SLURM
- Nella seconda fase le macchine diventano 16, il file system viene ospitato da una unita' FreeNAS con connessione dedicata ed il software viene gestito con interfaccia grafica
- Nella sua versione finale, AVES è composto da 30 macchine (per un totale di 120 cores)

Il cluster Aves (II)

Alcune caratteristiche di Aves:

- ha un costo etremamente contenuto
- permette di usare OSA sulle macchine del cluster in una modalità equivalente a quella parallela
- il file system utente e i dati del satellite risiedono su NAS indipendenti connessi al cluster in modalità iScsi
- i dati ISDC vengono sincronizzati giornalmente
- ogni utente ha un profilo in termini di risorse utilizzabili

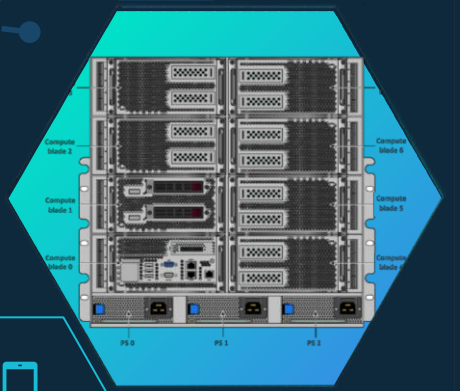
Il cluster Aves (III)

Inoltre:

- mette a disposizione degli utenti una interfaccia grafica semplice e intuitiva per la preparazione e il lancio dei run
- suddivide in gruppi le orbite di interesse e le assegna automaticamente ai differenti cores e nodi
- riduce le latenze trasferendo i dati necessari dallo storage centrale a quello locale (per la durata dei run)
- fornisce tools per verificare la consistenza del data set di interesse e per il monitoraggio dello stato dei run

permette un guadagno nelle performances fino a 70 volte

Aves2



Alcuni limiti di AVES:

- l'obsolescenza delle macchine che compongono AVES rende arduo reperire i componenti di ricambio
- l'architettura a 32 bit non ha permesso di espandere la RAM dei nodi oltre il limite di 4 Gb
- il sistema di storage basato su iSCSi impone severe restrizioni alle prestazioni del file system

Queste (ed altre) considerazioni ci hanno persuasi ad affrontare il progetto di AVES2

Aves2 (II)

AVES2 è basato su di una architettura BLADE UV 2000 (SGI); le sue caratteristiche principali sono:

- Numero di CPU: 160
- RAM disponibile: 1TB
- HD principale: SSD 1TB
- Interconnesione tra I nodi: Numalink 6

La scelta dell' architettura BLADE permette di azzerare i ritardi introdotti dall'uso della rete relativamente ai messaggi di servizio e allo scambio di dati tra i nodi del cluster

Aves2 (III)

- Il gestore delle risorse passa da SLURM a OpenLAVA
- L'uso di CGROUPS permette di assegnare agli utenti le risorse di sistema in modo più granulare
- Un processo supervisore (Shepherd) individua processi fuori controllo e ridefinisce le risorse a loro disposizione
- Il sistema di storage è basato su due unità Qsan F600Q-D316 (2 x 48 TB) connesse al blade tramite link ottico

GforC

- Nello studio di nuovi rivelatori per alte energie, si utilizza spesso di algoritmi di simulazione “Montecarlo” per lo studio delle interazioni tra materia ed energia
- Questo tipo di simulazione è molto dispendiosa in termini di risorse di calcolo e di tempi di esecuzione anche utilizzando macchine performanti
- Uno dei tool più efficaci per svolgere questa attività è Geant4

GforC (II)

- Ogni simulazione consiste in un programma C++ compilato utilizzando le sue librerie
- L'utente deve fornire (come dati iniziali) la geometria, la lista dei processi fisici, e gli stati iniziali del sistema
- Per agevolare questo tipo di attività è stato realizzato il cluster GforC (8 x 16 GB di RAM, 8 x 1 TB disco, 64 cores totali, file system di tipo OCFS2)

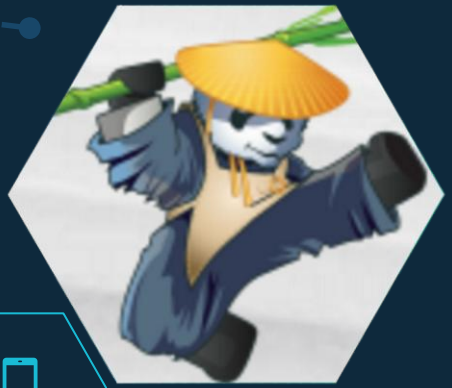
GforC (III)

- L’architettura è mutuata dal cluster AVES (modalità pseudo-parallela slicing automatico, GUI l'utente ...)
- La simulazione è distribuita su un numero di istanze pari al numero dei core permettendo un numero di particelle maggiore rispetto ad istanza singola (long int; N x 2^31)
- Utilizzato per studiare il background del microcalorimetro XMS (missione ATHENA: ESA) e la polarizzazione nel rivelatore CZT (missione NuStar: NASA)

VirtLab

- Il sistema HAVmS nasce con l'obiettivo di ottenere un sistema in alta disponibilità economico e con tempi quasi nulli nel ripristino del servizio in caso di guasto
- L'infrastruttura è basata su una coppia di macchine virtuali, una attiva ed una dormiente, ospitate da due server in ambiente XEN hypervisor
- I due server hanno indirizzi IP distinti e diversi da quello delle due macchine virtuali (unico)

VirtLab (II)

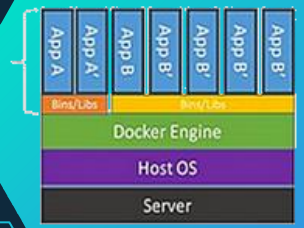


- Lo stato corrente della macchina attiva è replicato in quella dormiente da Remus (componente HA di XEN)
- I file systems delle due macchine virtuali sono sincronizzati da un processo DRBD
- In caso di guasto, Remus provvede ad attivare la macchina dormiente ed a isolare definitivamente quella in avaria
- Gli applicativi della macchina in avaria riprendono a girare coerentemente con l'ultimo checkpoint in quella "risvegliata"

VirtLab (III)

- LXC è un meccanismo leggerissimo per virtualizzare sistemi
- I containers Linux adottano un approccio completamente diverso rispetto alle tecnologie di virtualizzazione come Xen e KVM
- Le macchine virtualizzate condividono con il sistema ospitante il kernel utilizzando un sistema molto efficiente di isolamento e di sicurezza
- LXC è un’evoluzione di chroot per implementare sistemi completi virtuali, con l’aggiunta di meccanismi di gestione avanzate delle risorse attraverso cgroup

VirtLab (IV)



Un container è capace di offrire:

- un deployment semplificato
- una disponibilità rapida
- un controllo più granulare

Docker puo' "impacchettare" un'applicazione e le sue dipendenze in un contenitore virtuale, non effettua la portabilità delle macchine virtuali o dei SO, ma rende portabile il codice con cui l'applicazione è scritta

MDBirs

Il data base MongoDB fornisce scalabilità orizzontale utilizzando una tecnica chiamata sharding che distribuisce i dati su partizioni chiamate shards.

Lo sharding è assolutamente automatico, una specie di plug-and-play ...

Un replica set è un gruppo di demoni MongoDB con un arbitro (opzionale).

MDBirs (II)

In rapporto ai sistemi tradizionali (RDB) quelli non relazionali offrono maggiore velocità, delocalizzazione geografica e replicazione automatica dei dati

La combinazione di sharding/replica set permette di ottenere:

- scalabilita orizzontale illimitata
- memorizzazzione geografica secondo varie politiche (p.e.dati di strumento vicino agli utenti interessati)
- disponibilità continua dei dati

MDBirs (III)

MDBirs è nato per testare l'uso di mongoDB in rapporto a mySQL nella gestione dei dati astronomici; i test sono stati effettuati utilizzando il catalogo GSC (test di importazione e di interrogazione con vari tipi di query) in quanto:

- è a singola tabella (no relazioni)
- ha dati eterogenei (51 campi)
- è esteso (945.592.683 oggetti)
- è completo di posizioni, classificazione e magnitudini

E poi?

…GPGPU …

Sono state fatte delle prove relative ad applicazioni di dinamica molecolare utilizzando una GPU s1070

Studi sul comportamento del traffico in ambiente urbano (F. Farrelly,prog. Omnipolis) con una C2050

Analisi di evoluzione dinamica di ammassi globulari con codici Nbody (M. Merafina) con una GTX 780

Grazie!