Gebündelte Kraft mit Infiniband – neuer Compute-Cluster im CMS

Daniela-Maria Pusinelli | pusinelli@cms.hu-berlin.de

Im April 2010 wurde der bestehende Compute-Cluster durch einen Cluster ersetzt, der aus 32 Knoten besteht, die über ein schnelles Infiniband-Netzwerk miteinander verbunden sind. Damit steht den Nutzern des Computeservice ein System zur Verfügung, das durch sein schnelles Netzwerk und die parallele Datenbereitstellung das Entwickeln und Anwenden von parallelen Programmen sehr unterstützt.

Der bestehende Compute-Cluster, installiert im Jahre 2007, genügt schon lange nicht mehr den Anforderungen an einen zentralen Computeservice. In diesem Jahr wurde es möglich, ein neues System zu beschaffen, das den gewachsenen Anforderungen der Mitarbeiterinnen und Mitarbeiter der Humboldt-Universität Rechnung trägt. Die „Bündelung" der Prozessoren (Cores) mit Hilfe des Infiniband-Netzwerkes ermöglicht einen großen Geschwindigkeitszuwachs bei der Lösung von Problemstellungen.

Vorteile des neuen Clusters

Die Vorteile des neuen Clusters gegenüber dem abzulösenden System bestehen in:

- schnelleren Xeon-Prozessoren
- ausgewogener Intel-Nehalem Architektur
- 256 Cores gegenüber 48 Cores
- 3-fachem RAM von 48 GB
- QDR[1]-Infiniband-Netzwerk mit 40 Gbit/s
- parallelem Lustre-Filesystem

Bei der Beschaffung wurde ebenfalls Wert gelegt auf eine energiesparende Variante. Aus diesem Grund werden Low-Voltage-CPUs verwendet.

Der installierte Compute-Cluster soll Instituten mit einer geringen Computer-Ausstattung die notwendige Rechenkapazität zur Verfügung stellen. Instituten mit einer guten Ausstattung, aber nicht immer ausreichenden Ressourcen, bietet er Überlaufkapazität an. Mitarbeiterinnen und Mitarbeiter, die Problemstellungen bearbeiten müssen, die nur „massiv" parallel oder mit einem enormen Hauptspeicherbedarf zu bearbeiten sind, bekommen vom CMS die Befürwortung, das Rechner-System des HLRN zu benutzen. Weitere Hinweise dazu sind auf den Webseiten des CMS, Stichwort „HLRN", zu finden.

1 Quad Data Rate

Aufbau und Cluster-Verwaltung

Der neue Compute-Cluster besteht aus acht Compute-Servern (Supermicro Twin²-Server, Abb. 1) von je zwei Höheneinheiten. Jeder Server hat vier auswechselbare Knoten, jeder Knoten besteht aus zwei Intel-Nehalem-CPUs, die wiederum je vier Cores haben. Das ergibt eine Anzahl von 32 Knoten mit je acht Cores, also insgesamt 256 Cores. Jeder Knoten ist mit einem Infiniband-Kontroller QDR sowie 48 GB RAM ausgerüstet. Für die lokale Datenablage steht eine SATA-II-Disk von 500 GB zur Verfügung.

Die Intel-Nehalem-Mikroarchitektur besteht aus Intel-Prozessoren mit integriertem Speicherkontroller, der einen schnellen Zugriff auf den Hauptspeicher ermöglicht und damit die Latenzzeit verringert. Ein QuickPath Interconnect zwischen Prozessoren und Chipsatz ermöglicht einen hohen Durchsatz und gute Skalierbarkeit.

Die Rack-Server sind in einen wassergekühlten Schrank im Grimm-Zentrum eingebaut. Die Verwaltung der Compute-Server erfolgt über zwei redundante Master-Server, die in Prozessor- und Speicher-Ausstattung einem Rechenknoten entsprechen (Abb. 2). Zusätzlich sind sie mit FC-Karten ausgestattet, um einen Zugriff auf das SAN[2] zu ermöglichen. Weiterhin verfügen sie über eine lokale SATA-II Disk mit einer Kapazität von 1 TB, auf der zu sichernde, permanente Daten abgelegt werden sollen. Die Master-Server sind verantwortlich für die Neuinstallation des Betriebsystems auf den Rechenknoten sowie deren Start, die Bereitstellung der Entwicklungs- und Anwendungssoftware für die Knoten und die Überwachung des Batchbetriebes. Sie allein kommunizieren in das universitäre Netzwerk. Innerhalb des Compute-Clusters wird über ein internes privates Ethernet-Netzwerk kommuni-

2 Storage Area Network

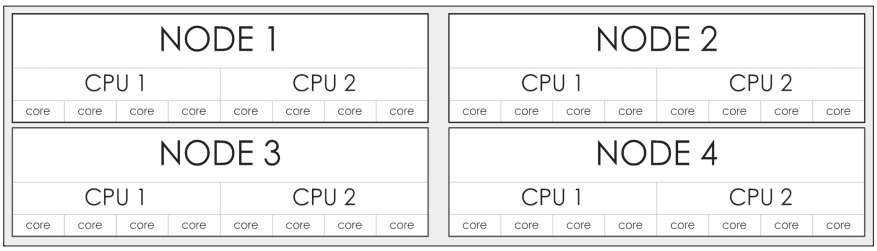

Abb. 1: Aufbau des Compute-Server

ziert. Das interne Infiniband-Netzwerk ermöglicht die schnelle Kommunikation der Knoten untereinander während der Abarbeitung eines parallelen Programms.

Bei Ausfall eines Knoten kann dieser „Hot-Swap" im Compute-Server ausgetauscht und über Kickstart neu gestartet werden. Ein Neustart dauert etwa 10 Minuten. Zur Überwachung der Knoten werden Nagios und Ganglia eingesetzt. Mittels IPMI[1] kann über das Netzwerk auf die Konsolen der einzelnen Knoten zugegriffen werden.

Kommandos, die auf allen Knoten gleichzeitig ausgeführt werden sollen, werden mit Hilfe der PSSH[2] gestartet. Alle weiterführenden Informationen werden über die Webseite des CMS unter dem Stichwort „Computeservice" nachzulesen sein.

Hochverfügbarkeit

Alle Server, Master-Server, Lustre-Fileserver und die Computer-Server, haben redundante Netzteile. Die für die Überwachung der Knoten notwendigen Dienste werden redundant auf den beiden Master-Servern installiert. Diese arbeiten dann im Cluster-Verbund. Ein Server arbeitet als aktiver Server, der andere Server als passiver. Mittels Heartbeat wird der „Herzschlag" kontrolliert, und bei Ausfall des aktiven übernimmt der passive Server alle Dienste. Damit wird eine Hochverfügbarkeit (High Availability) des Compute-Clusters erreicht.

Zugang für Nutzer

Mitarbeiterinnen und Mitarbeiter, die eine Vielzahl von Prozessoren sowie einen größeren Hauptspeicher für ihre Berechnungen benötigen, werden auf dem Compute-Cluster zugelassen. Voraussetzung ist ein Basisaccount am CMS. Dieser wird dann auf formlosen Antrag hin freigeschaltet, das heißt zur LDAP-Gruppe für den Computeservice hinzugefügt. Berechtigte Nutzer loggen sich auf einem der Master-Server mittels SSH ein. Ein direktes Einloggen auf die Rechenknoten aus dem HU-Netzwerk ist nicht möglich, ein Zugang über den Master-Server aber im Bedarfsfall. Eine Firewall schirmt die Master-Server nach außen ab und ermöglicht ein Login nur aus der HU-Domäne.

Auf jedem Rechenknoten des Compute-Clusters stehen den Nutzern Dateisysteme zur Ablage von Daten zur Verfügung:

- das einheitliche Homeverzeichnis `/home/<institut>/<account>`
- das OpenAFS Homeverzeichnis `/afs/.cms.hu-berlin.de/user/<account>`
- das auf einem Knoten lokale, temporäre Verzeichnis `/scratch/<account>`
- das Arbeitsverzeichnis `/work/<account>` auf den Lustre-Servern
- das permanente Verzeichnis `/perm/<account>` auf den Master-Servern

Zu beachten ist, dass die `/scratch`- und `/work`-Verzeichnisse nicht gesichert werden. Außerdem unterliegen diese Verzeichnisse einem Kontrollmechanismus, der Daten, entsprechend ihrem Alter bei Bedarf entfernt.

Daten, die permanent und sicher abgelegt werden sollen, müssen ins `/perm`-Verzeichnis oder in eines der Homeverzeichnisse kopiert werden.

Software

Auf allen Servern läuft als Betriebssystem CentOS LINUX [2], das der RedHat Distribution entspricht.

Programmentwicklung

Für die Programmentwicklung stehen auf dem Compute-Cluster entsprechende Compiler und mathematische Bibliotheken zur Verfügung. Das sind die Intel-Compiler und MKL[3], die Portland-Group-Compiler sowie die frei verfügbaren GNU-Compiler. Zu allen Compilern sind die dazugehörigen Debugger installiert. Die Parallelisierung der Programme erfolgt mit OpenMPI[4], die Kommunikation der Knoten dann über das Infiniband-Netzwerk.

Die Auswahl eines Compilers und einer Compiler-Version erfolgt mit Hilfe eines Skriptes (Module), das im Verzeichnis für Modules gespeichert ist.

Anwendungssoftware

Alle Anwendungen, die auf dem Compute-Cluster gerechnet werden sollen, sind im Batchbetrieb zu starten. Als Anwendungssoftware werden die vorhandenen Pakete Gaussian (03 und 09), ORCA, Turbomole, Matlab, Maple und R (CRAN) zur Verfügung gestellt. Die benötigte Software mit entsprechendem Versionsstand wird analog zu den Compilern über ein Module geladen. Weitere Anwendungen können bei Bedarf zusätzlich installiert werden.

Gaussian

Das Programmpaket Gaussian dient der Berechnung der elektronischen Struktur von Molekülsystemen. Es ist für theoretische und experimentelle Chemiker von Interesse.

Gaussian 09 wird auf dem Cluster in der Version A.02 installiert. Es kann auf einem Knoten mit maximal 8 Cores gestartet werden, wobei wahrscheinlich nur 4 Cores einen signifikanten Speedup aufweisen werden. Geeignet ist hier die Batch-Queue „par" (Tab. 1). Für eine pa-

1 Intelligent Platform Management Interface
2 Parallel Secure Shell
3 Mathematical Kernel Library
4 Message Passing Interface

Queue	Priorität	Prozessoren	Memory	Slots	Runtime
inter	+15	1	1 GB	8	3 h
short	+10	1	4 GB	32	6 h
bigpar	+5	8, 16, 24, 32	40 GB	128	48 h
par	0	4,8	20 GB	64	24 h
long	-5	1	4 GB	32	96 h

Tab. 1: Queuekonfiguration

rallele Rechnung über einen Knoten hinweg verwendet Gaussian die Parallelisierungs-Software Linda. Es hat sich aber schon auf dem aktuellen Cluster gezeigt, dass Linda nicht sehr performant ist, so dass die Software nicht wieder gekauft wurde. Aber innerhalb eines Knotens rechnen einige Links mit SMP-Routinen, was einen signifikanten Speedup bringt.

Bei Bedarf kann auch die vorgehende Version Gaussian 03 installiert werden. Alle oben gemachten Aussagen treffen auch auf Gaussian 03 zu.

GaussView

Das Programm GaussView als ein geeignetes grafisches Werkzeug zur Erstellung von Molekülen, der Manipulation der erstellten Moleküle und zur graphischen Darstellung der Ergebnisse von Gaussian Rechnungen wird in der Version 3.09 zur Verfügung gestellt. Es ist über die Batch-Queue „inter“ (Tab. 1) zu starten.

Turbomole

Turbomole ist ein Programmpaket für „ab initio“ quantenchemische Berechnungen, das von der Gruppe um Prof. Dr. Reinhard Ahlrichs an der Universität Karlsruhe entwickelt wurde. Die parallele Version ist MPI basiert. Derzeit haben wir eine Lizenz für die Version 6.1, die geeignete Batch-Queue ist „par“ (Tab. 1).

ORCA

Das Programmpaket ORCA ist ein vielseitges Werkzeug für Berechnungen in der Quantenchemie mit dem Fokus auf spektroskopische Eigenschaften von Molekülen. Es wurde an der Universität Bonn entwickelt und steht auch schon auf dem aktuellen Cluster zur Verfügung. Es läuft als serielle (Queue „long“) oder parallele MPI-Version (Queue „par“).

Paralleles Filesystem Lustre

Zwei zu den Master-Servern baugleiche Server werden für die Bereitstellung des objektorientierten, verteilten Lustre-Filesystems genutzt. Lustre [3] ist ein Open-Source-Produkt, das von der Firma SUN-Microsystems (jetzt Oracle) weiterentwickelt und unterstützt wird.

Beide Server werden als Meta-Data-Server (MDS) und als Object-Storage-Server (OSS) arbeiten. Aber nur ein MDS ist aktiv, der andere ist im passiven Zustand. Mittels Heartbeat wird überprüft, ob der aktive Server noch am Leben ist und gegebenenfalls der passive aktiviert. Für das Lustre-Filesystem `/work` wird ein Meta-Data-Target (MDT) benötigt. Weiterhin werden fünf Object-Storage-Targets (OST) angelegt, auf einem Server drei und auf dem anderen zwei. Der Vorteil des Lustre-Filesystems besteht im parallelen Lesen/Schreiben eines Daten-Objektes direkt von/auf den/die OSTs. Die Bandbreite des Datenzugriffs skaliert nahezu linear mit der Anzahl der OSTs.

Batchbetrieb

Die Verwaltung der Rechenaufträge (Jobs) erfolgt über Sun Grid Engine (SGE) [4], ebenfalls freie Software, die schon in vergangenen Jahren auf den Compute-Servern des CMS im Einsatz war.

Ein Standard SGE-System besteht aus einer Zelle, die von einem „master host“ verwaltet wird und die eine beliebige Anzahl von „execution hosts“ enthält. Die Aufträge werden von den „submit hosts“ abgeschickt. Bezogen auf den Compute-Cluster überwacht der Master-Server die Aktivitäten der Zelle, während die Rechenknoten die Ausführung der Jobs, die ihnen vom Master-Server zugeteilt werden, überwachen. Alle Aufträge müssen vom Master-Server abgeschickt werden.

SGE unterstützt parallele Umgebungen (z. B. OpenMPI) und bietet die Möglichkeit, Jobs an andere Queueing-Systeme zu schicken. Ein Grafisches-User-Interface QMON ermöglicht die Kontrolle des Queueing-Systems. Das Abschicken der Jobs erfolgt gewöhnlich über ein Shell-Skript. Auf eine Beispielsammlung von Batch-Skripten kann jeder Nutzer über das Verzeichnis `/perm/skripte` zugreifen.

Die Verteilung der in den Queues wartenden Jobs (Pending Liste) erfolgt nach Kontrolle der Last auf den Knoten, an die mit der geringsten Last. Die Fair-Share-Policy sichert, dass neue Jobs

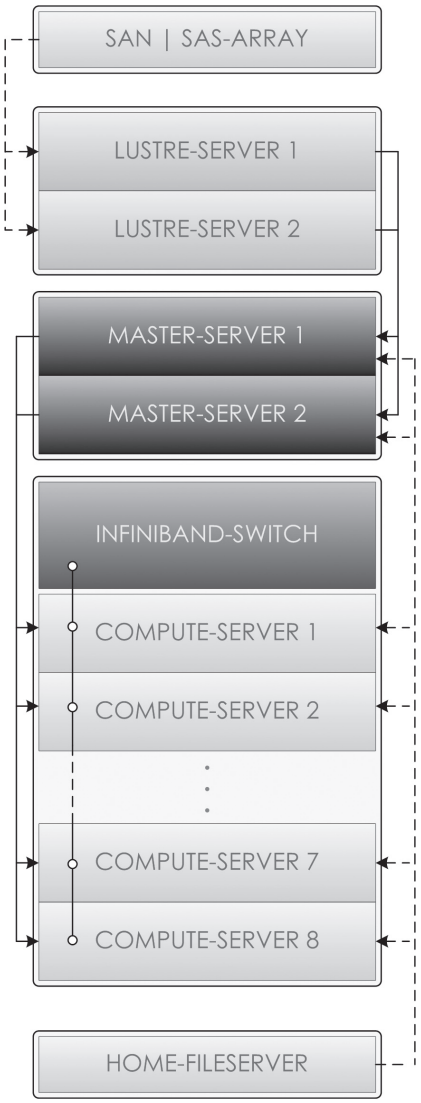

Abb. 2: Aufbau des Compute-Clusters

eines Nutzers, der bereits einen Job in einer Queue hat, hinter die Jobs anderer Nutzer gestellt werden, die die gleiche Priorität haben. Die vorgesehene Queue-Konfiguration ist in der Tabelle 1 dargestellt.

Die Spalte mit den Prioritäten gibt darüber Auskunft, welche Jobs Vorrang vor anderen haben. Interaktive Anwendungen sollen sofort zum Start kommen, ebenso Jobs in der Queue `short`. In der Queue `bigpar` können für die Anzahl der Prozessoren (Cores) nur Vielfache von acht angegeben werden. Damit wird verhindert, dass diese Jobs nicht alle Cores eines Knoten benutzen. Ob das sinnvoll ist, wird die Praxis zeigen. Die Queue `par` dagegen kann vier oder acht Cores nutzen, da bleiben dann eventuell vier Slots frei für serielle Jobs der Queues `short, long` oder `inter`. Die Anzahl der Slots in der Tabelle gibt Auskunft über die Anzahl der Cores, die von Jobs genutzt werden dürfen. Demnach ist denkbar, dass
4 Jobs (je 32 Cores) in der Queue `bigpar`
8 Jobs (je 8 Cores) in der Queue `par`
48 Jobs (je 1 Core) in der Queue `short`
12 Jobs (je 1 Core) in der Queue `long`
4 Jobs (je 1 Core) in der Queue `inter`
gleichzeitig rechnen und damit alle 256 Cores „beschäftigt“ sind.

Die Mitarbeiterinnen und Mitarbeiter der einzelnen Arbeitsgruppen werden auch unter SGE zu Gruppen zusammengefasst. Das ermöglicht es, Gruppen den Zugriff auf bestimmte Queues zu erlauben sowie die Ressourcen auf die Gruppen zu verteilen, wenn die Fair-Share-Policy eingestellt ist.

Literatur

[1] https://www.hlrn.de/home/view
[2] http://www.centos.org/
[3] http://wiki.lustre.org/index.php/Main_Page
[4] http://wikis.sun.com/display/gridengine62u5/Home