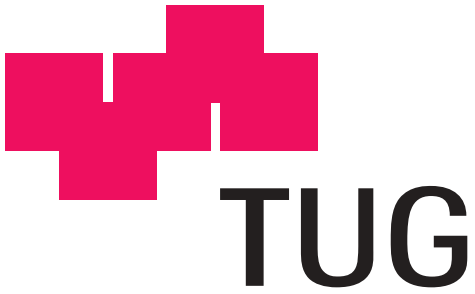
Stocker Gernot

Linux Cluster für bioinformatische Anwendungen

Diplomarbeit



Institut für Elektro- und Biomedizinische Technik

Technische Universität Graz

Krenngasse 37, A - 8010 Graz

Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Gert Pfurtscheller

Betreuer: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Trajanoski Zlatko

Begutachter: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Trajanoski Zlatko

Graz, September 2001

Für meine Eltern

Kurzfassung

Das Ziel dieser Diplomarbeit war Design, Planung und Aufbau eines Beowulf Clusters, der hardwaremäßig aus gewöhnlichen PCs besteht, die über ein Netzwerk verbunden sind und gemeinsam parallele Applikationen unter Linux berechnen können. Dabei wurde der Cluster und die Applikationen auf die speziellen Bedürfnisse der Bioinformatik zugeschnitten.
Aus dieser Diplomarbeit resultiert ein voll funktionstüchtiger Beowulf Cluster, der aus 24 Knoten mit insgesamt 48 Prozessoren und einem Master mit weiteren 2 Prozessoren besteht. Die Knoten untereinander sind mit zwei 100 Mbps FastEthernet-Karten verbunden und greifen damit auf das 240 GB große Plattensystem des zentralen Masters zu, der mit Gigabit-Ethernet integriert ist. Für die Überwachung des Clusters wurde ein plattformunabhängiges Monitoringtool entwickelt, das laufend aktuelle Statusinformationen des Clusters anzeigt. Weiters wurden gängige Werkzeuge der Bioinformatik wie SRS (Sequence Retrival System) , BLAST (Basic Local Alignment Search Tool), WU-BLAST und FASTA für den Clusterbetrieb installiert. Dieses Tool stellt eine Kernkomponente eines genomischen Informationsmanagementsystems dar, die für Forschungsarbeiten der modernen Biowissenschaften unabdingbar ist.
Schlüsselworte: Parallelisierung, Bioinformatik, Beowulf Cluster, Linux

Abstract

Objective of this master thesis was the design and creation of a Beowulf Cluster consisting of lowcost PC's interconnected by a network and executing parallel tasks running on Linux. The cluster and the applications were optimized for bioinformatic purposes. This master thesis resulted in a fully functional compound of 24 PC's with 48 processors including a master with two additional processors. The nodes are interconnected with two 100 Mbps FastEthernet-Networks and access the 240 GB disksystem of the central master through a Gigabitinterface. Additionally, a platform independent monitoring tool was developed and common bioinformatic applications were installed: SRS (Sequence Retrival System), BLAST (Basic Local Alignment Search Tool), WU-BLAST and FASTA The developed tool represents a central component of a genomic informationmanagement system which is essential for modern molecular Biosciences.
Keywords: Parallelisation, Bioinformatics, Beowulf Cluster, Linux,

Abbildungsverzeichnis

Glossar

Beowulf Cluster Rechnerverbund, der sich aus PCs zusammensetzt und über ein Netzwerk parallelisierte Anwendungen berechnet.

CPU Central Processing Unit, ist das zentrale Herz eines jeden Computer der "von Neumann"-Architektur, das jegliche Berechnungen durchführt.

ECC (Error Checking and Correcting) Speichertechnologie, die Bitfehler erkennt und einfache Fehler automatisch korrigiert

PC Personal Computer sind gängige, handelsübliche Computer, die für den hausgebrauch verwendet werden.

parallelisierte Programme sind Programme, deren Algorithmus so angepaßt wurde, daß es auf mehrere, idealerweise unabhängige Unterprozesse aufteilbar ist, und damit auf verschiedenen CPUs verteilt übers Netzwerk oder SMP-intern berechnet werden kann.

RAID (Redundant Array of Independent Disks) ist eine Technologie, die Daten redundant mit unterschiedlichen Sicherheitsstufen abspeichert und beim Ausfall einer Platte eine Wiederherstellung der Daten aufgrund der Zusatzinformationen ermöglicht.

SMP Symmetric Multi Processing bedeutet, daß mehrere Prozessoren in einem System integriert sind, wobei das Betriebssystem darauf achten muß, daß sämtliche verfügbaren Prozessoren im gleichen Maße ausgelastet werden.

SDRAM (Synchronous Dynamic Random Access Memory), Arbeitsspeichertechnologie, die laufend ihre Datensignalpegel auffrischen muß, um ihren Speicherinhalt halten zu können.

Swarming Verfahren zur verteilten Berechnung von nicht parallelisierten Programmen durch die Änderung der Eingabedaten.

Treiber betriebssytemnahe Software, die ein Gerät ansprechbar und nutzbar macht. D.h. diese Software übernimmt die gerätespezifische Kommunikation zwischen Betriebssystem- und Hardwareebene, um für Anwendungen im Benutzerbereich einen einheitlichen Zugriff auf Gerätefamilien zu ermöglichen.

Inhaltsverzeichnis

Kapitel 1

Einleitung und Aufgabenstellung

1.1 BioCluster in Graz

In den letzten Jahren hat sich die Bio- und Genforschung zu einer der Schlüsseltechnologien herauskristallisiert und wird als ein wichtiger Industriezweig der Zukunft gehandelt. Auch in Österreich wurden die Zeichen der Zeit erkannt und der Aufbau neuer Kompetenzzentren initiiert.

Inzwischen wurde durch die enge Kooperation mit führenden Institutionen eine Generation von Forschern an den Grazer Universitäten ausgebildet, die den Biostandort Graz international kompetitiv macht. Hand in Hand geht damit die Entwicklung der Bioinformatik einher, die sowohl zur Erstellung von Hypothesen bei der Laborforschung als auch zur automatisierten Auswertung der Labordaten herangezogen wird.

Um in diesem Sinne gezieltes Arbeiten zu ermöglichen, muß kontinuierlich die EDV-Infrastruktur ausgebaut werden. Als erschwingliche und notwendige Unterstützung der Forschungsarbeiten hat sich an zahlreichen Institutionen die Clusterarchitektur etabliert. Der Begriff "Cluster" ist die Bezeichnung für einen Verband von Rechnern, die gemeinsam an der Lösung eines parallelisierbaren Problems arbeiten. Man spricht von einem Beowulf Cluster [33], wenn sich dieser Rechnerverbund aus Standard-PCs (Personal Computer) zusammensetzt, die über ein Netzwerk verbunden sind. Daher kommt auch die

Bezeichnung Knoten, denn die PCs stellen Knotenpunkte in diesem Rechnernetz dar. Mit einer derartigen Konfiguration läßt sich eine Rechenleistung im "gigaflop"- Bereich (Milliarden Operationen pro Sekunde) erzielen, die durchaus Vergleiche zu Supercomputern zuläßt. Preislich liegt der Beowulf Cluster jedoch weit unter dem Supercomputer, wobei die Kosten durch die Verwendung eines freien Betriebssystemes z.B. Linux weiter reduziert werden.

Ein derartiges Werkzeug bereichert die Arbeit der Biologen und Bioinformatiker in Graz, zumal in Kürze die Produktion der BioChips anlaufen wird und durch den automatisierten Produktionsprozeß große Datenmengen anfallen werden, die abgelegt, aufbereitet und weiterverarbeitet werden müssen. Der Standort Graz will sich aktiv an der Genomanalyse beteiligen. Das hat zur Folge, daß beträchtliche Mengen an externen Daten lokal zur Verfügung gestellt werden müssen, damit die entsprechenden Analyse-Werkzeuge zur Anwendung kommen können. Dementsprechend muß genügend Speicherkapazität in der Umgebung des Clusters eingeplant werden, um effizient und verteilt arbeiten zu können. Für die lokale Verfügbarkeit von Gendatenbanken spricht zusätzlich noch die Möglichkeit der engen Kooperation mit profitorientierten Firmen aus dem LifeScience-Bereich. Proprietäre Datenbanken, die diese Firmen nur ausgewählten Einrichtungen zur Verfügung stellen wollen, können lokal mit den freien Gendatenbanken kombiniert und gegengeprüft werden, ohne daß sie über das Internet geschickt werden müssen.

Abschließend ist noch anzumerken, daß rund um den Cluster und dessen Möglichkeiten bereits durch die Präsenz der Infrastruktur der Grundstein für internationale Wettbewerbsfähigkeit und deren Folgeprojekte gelegt worden ist. Mittlerweile hat sich herausgestellt, daß der Cluster und seine Umgebung ein Kristallisierungspunkt für die Anwerbung internationaler Forschungsprojekte (6. Rahmenprogramm der EU, ESA Projekte etc.) ist, die den Standort Graz im internationalen Wettbewerb an die vorderste Stelle positioniert.

1.2 Aufbau eines Beowulf Clusters

1.2.1 Hardware

Die Hardware eines Beowulf Clusters besteht aus drei wesentlichen Bestandteilen, die den Anforderungen entsprechend dimensioniert werden müssen:

1. Cluster-Knoten: Die Cluster-Knoten sind die eigentlichen Arbeiter, auf die eine parallele Aufgabenstellung verteilt wird. Diese PCs müssen eine möglichst hohe Rechenleistung erbringen, um die gestellte Aufgabe in kürzester Zeit zu lösen. Damit ist nicht nur ein schneller Prozessor für die rasche Lösung des gestellten Problemes verantwortlich, sondern auch die gesamte Konfiguration um den Prozessor herum:

   - Hochwertiger und schneller Speicher (RAM), um den Prozessor nicht unnötig einzubremsen,
   - ein schnelles Motherboard mit einer großen BUS-Breite und
   - geeignete Steckkarten, z.B. Netzwerkkarten,

   sind die Komponenten eines System, die aufeinander abgestimmt werden müssen, um negative Folgen zu verhindern.

2. Master-Server: Als Master bezeichnet man den Rechner, von dem aus die Prozesse auf die Knoten verteilt werden. Dort werden die benötigten Daten zentral gespeichert und bei Bedarf auf alle Knoten verteilt. Er ist das Tor des Clustersystems zur Außenwelt, mit dem sich die Benutzer verbinden und ihre parallelisierten Programme starten können.

3. Netzwerk : Die Verbindung der einzelnen Knoten untereinander und die Verbindung der Knoten zum Master spielen eine zentrale Rolle im Aufbau des Clusters. Ist die Kommunikation der Knoten innerhalb des Rechnerverbandes zu langsam, kann die Rechenleistung schneller Knoten nicht ausgenützt werden. Sie müssen laufend auf andere Knoten warten, da die schnell berechneten Ergebnisse über das Netzwerk zu

den wartenden Knoten zu langsam transferiert werden. In weiterer Folge kommt es zu einem Kollaps des Netzwerkes.

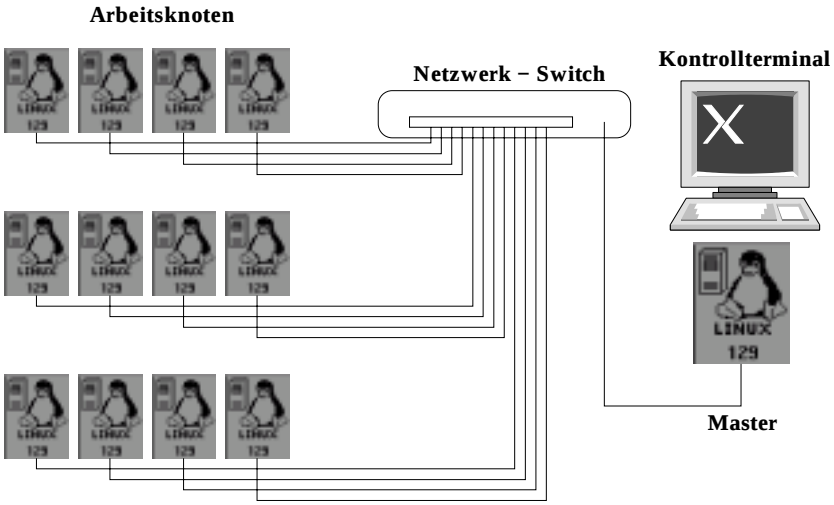


Abbildung 1.1: prinzipieller Aufbau eines Beowulf Cluster

1.2.2 Software

Mit einer optimierten Hardware ist ein Cluster nach der Installation eines netzwerkfähigen Betriebssystems schon voll einsatzbereit und könnte bereits verwendet werden. Dies funktioniert aber nur dann, wenn man im Programm sämtliche Kommunikation zwischen den einzelnen Prozessen übers Netzwerk selbst implementieren würde. Die Software sollte dabei bereits berücksichtigen, daß sie den Netzwerkverkehr möglichst minimiert, um den beschrieben Kollaps zu verhindern.

Es gibt in diese Richtung Bestrebungen, daß Programmierer auf bereits vorgefertigte und teilweise standardisierte Kommunikationssoftware zurückgreifen können. Wie diese Programmpakete arbeiten und welche Vor- bzw. Nachteile sie mit sich bringen, wird zu einem späteren Zeitpunkt (siehe Kapitel Clustersoftware Seite 34) noch detaillierter beschrieben.

1.3 Anforderungen an das System

Aus den Anforderungen, die die Bioinformatik stellt, resultieren folgende zwei Fragen, deren Klärung maßgeblich die Planung beeinflußt:

- **WELCHE** Daten werden **WO** benötigt?

  Die Bioinformatik arbeitet mit riesigen Gendatenbanken, deren Größen sich im Gigabyte-Bereich befinden. Mit verschiedenen Verfahren werden diese Datenmassen, welche im normalen Textformat vorliegen, durchkämmt und auf ihre Ähnlichkeit untersucht. Wenn nun ein solcher Vorgang auf mehreren Knoten parallelisiert durchgeführt werden soll, müssen die gesamten bzw. ein Großteil der Daten auf jedem Knoten zur Verfügung stehen. Möglichst viele Daten müssen daher lokal auf den Knoten abgelegt und zwischengespeichert werden, um mit angemessener Geschwindigkeit darauf zugreifen zu können. Es bieten sich 2 Möglichkeiten an:

  - Jeder Knoten wird von der Speicherkapazität so ausgelegt, daß die Datenbanken lokal auf ihren Platten abgespeichert werden können. Damit ist unmittelbar ein sehr hoher Wartungsaufwand verbunden, um die sich rasch ändernden Daten aktuell zu halten. Hiermit ist aber noch nicht berücksichtigt, daß herkömmliche Platten zu klein sind, um alle Daten halten zu können.
  - Die Datenbanken werden zentral an einer dezidierten Stelle abgelegt, welche über das Netzwerk schnell erreichbar sein muß, d.h. der Durchsatz des Netzwerkes muß so dimensioniert sein, daß die Transferzeit der Daten zu den Knoten minimiert wird.

- **WIEVIEL** muß bei der Lösung eines Problems **BERECHNET**, **WIEVEL** muß **KOMMUNIZIERT** werden?

  Aufgrund des Algorithmus und dessen Implementierung kommt es zu einer mehr oder weniger intensiven Kommunikation zwischen den Arbeitsprozessen auf den Knoten. Je nach Netzwerktechnologie resultiert daraus eine Überlastung des Netzwerks, da Berechnungskommunikation und Dateiaustausch übers Netz erfolgen.

Das Gegenteil dazu wäre die Variante des sogenannten "swarmings" . Damit können nicht-parallelisierte Programme auf den Knoten mit unterschiedlichen Parametern und Eingabedaten gestartet werden. Diese laufen völlig autonom, - jeder für sich - ab und liefern Ergebnisse, die an zentraler Stelle wieder gesammelt werden können. In der Bioinformatik werden beide angeführten Prinzipien verwendet. Aus diesem Grund muß den resultierenden Anforderungen (schnelles Netzwerk, schnelle lokale Zwischenspeicherung etc.) Rechnung getragen werden.

1.4 Aufgabenstellung

Das Ziel der Diplomarbeit war Planung, Aufbau und Installation eines Beowulf Clusters, der auf die oben angeführten Bedürfnisse der Bioinformatik abgestimmt ist. Die Planung mußte sich in einem finanzierbaren Rahmen bewegen und am Ende sollte eine Hardware-Empfehlung an die Verantwortlichen abgegeben werden, damit diese nach einer Überprüfung den Ankauf tätigen konnten.
Der Aufbau mußte in einer klimatisierten Umgebung und gemäß der gewählten Topologie durchgeführt und die Einzelkomponenten aber auch das Zusammenspiel des Systems als funktionierendes Ganzes intensiv getestet werden.
Nach Abschluß der Hardwarekonfiguration und der intensiven Tests im Dauerbetrieb mußte in Hinblick auf eine zukünftige Erweiterung ein einfaches Installationssystem des Betriebssystems Linux entwickelt werden, das ohne größeren Aufwand erlaubt neue Knoten in den Rechnerverband automatisiert zu integrieren. Im Anschluß daran mußte für die Benutzer des Systems eine Form von zentraler Authentifizierung gewählt werden, die ein Erreichen eines jeden Knotens und damit die volle Nutzung des Systems ermöglichte. Allerdings mußte auch vorgesehen werden, daß dezidiert Teile des Systems für wenige Benutzer separiert werden können, um für spezielle, meist kurzfristige Projekte exklusive Sub-Cluster bilden zu können.
Es sollten sämtliche Formen der Parallelisierung, wie sie im Anhang auf Seite 59 beschrieben werden, unterstützt und den Anwendern des Clustersystems auf einfache Weise

zugänglich gemacht werden. Selbiges gilt für den Aufbau der Daten-Infrastruktur, wobei jeder Benutzer die Möglichkeit haben muß, von jedem Knoten im Cluster auf einen zentralen Datenpool zugreifen zu können.
Vorausgesetzt wurde eine möglichst einfache Wartung und eine laufende Überwachung des Systems, die den normalen Arbeitsbetrieb nicht einschränken sollte, um schnell auf Ausfälle oder Ungereimtheiten im System reagieren zu können.
Zuguterletzt sollte noch die Möglichkeit geschaffen werden, Anwendungen der Bioinformatik via Webinterface zugänglich zu machen, wobei bereits bestehende Systeme auf ihre Tauglichkeit geprüft und eventuell auch eigene Erweiterungen programmiert werden sollten.
Während der Arbeit sollten laufend wichtige Erkenntnisse für den Aufbau des Clusters dokumentiert werden, um auch später Designentscheidungen verstehen und Anpassungen vornehmen zu können. Damit sollte die Grundlage für die Wartbarkeit des Systems nach dem Aufbau garantiert werden.

Kapitel 2

Methoden und Ergebnisse

Der Aufbau eines Beowulf Clusters besteht im Wesentlichen aus drei Schritten:

- Bestimmung der Anforderungen an das Gesamtsystem
- Festlegung und Ankauf der Hardware
- Installation der Software

Daraus resultiert auch die Gliederung des folgenden Kapitels, das grundsätzliche Überlegungen möglicher Realisierungen und Designentscheidungen beinhaltet, aber auch auf eine grobe Skizzierung der Installation abzielt.

2.1 Hardware

Für die Auswahl der Cluster-Hardware gilt allgemein, daß nur erprobte Kombinationen von Geräten und deren Treiber (Software, welche den Zugriff und die Nutzung eines Gerätes ermöglicht) zielführend sind. Die Erfahrung aus vorangehenden Pilotprojekten hat gezeigt, daß im Einzeleinsatz betriebene Hardware unter Linux meist unproblematisch funktioniert. Taucht dort aber nur das geringste Problem, z.B. bei der Initialisierung, auf, so kann davon ausgegangen werden, daß dasselbe Problem im Clustereinsatz den gesamten Betrieb stört oder gar unmöglich macht. Genau dieser kritische Vorgang ereignet sich

auf allen Knotenrechnern im selben Zeitraum und führt zu unvorhersehbaren Folgen und unerklärlichen Seiteneffekten im Gesamtsystem (siehe Anmerkung zur Netzwerkkonfiguration auf Seite 25).

2.1.1 Cluster-Knoten

Als Cluster-Knoten kommen in Beowulf Cluster gewöhnliche PCs zum Einsatz, deren Kernkomponenten Prozessor, Arbeitsspeicher und Festplatten-Subsystem qualitativ hochwertig sein müssen. Allerdings sollen der Wert eines Knoten in einem akzeptablen Rahmen bleiben, damit er bei einem Ausfall problemlos ersetzbar ist.

Prozessor Die Festlegung des Prozessors (CPU) ist eine zentrale Aufgabe bei der Auswahl der Cluster-Hardware, da davon die Rechenleistung, aber auch das gesamte Umfeld der CPU im Knoten abhängt. Steht die Auswahl des Prozessors fest, kann erst über weitere Komponenten wie Motherboard, Arbeitsspeicher (RAM) etc. entschieden werden. Es stehen zwei Kategorien zur Auswahl:

- 32-Bit Systeme
- 64-Bit Systeme

Das bedeutet, daß 32- bzw. 64-Bit-Daten in einem Rechenschritt verarbeitet werden können. Die momentan gängige PC-Architektur verwendet 32-Bit-Prozessoren, auf die nun genauer eingegangen wird. 64-Bit-Architekturen werden bei der Diskussion über mögliche zukünftige Ausbauvarianten (siehe Seite 47 ff.) noch detaillierter behandelt.
Für den Einsatz im Cluster kommen 3 Prozessortypen in Frage:
Intel^{TM}s *PIII* ist ein mittlerweile ausgereifter Prozessor, der mit einer Taktfrequenz zwischen 450MHz und 1GHz erhältlich ist. Seine Herstellung beruht auf 0.18-micron Technologie (Strukturbreite bei der Halbleiter-Fertigung) und kann mit 133MHz FrontSideBus-Frequenz (Frequenz, mit der Transferbus zwischen CPU und RAM betrieben wird) angesprochen werden. Zu seinen Vorteilen kann die geringe Latenzzeit und die hohe Bandbreite von 256Bit zwischen Prozessor und Cache gezählt werden. Hinzu kommt noch, daß

Motherboards optimal auf diesen Prozessortyp abgestimmt und wegen der weiten Verbreitung günstig zu erstehen sind. Ausschlaggebend für den Einsatz im Cluster ist auch noch seine SMP-Fähigkeit (Symmetric Multi Processing), damit mehrere Prozessoren in einem Knoten verwendet werden können. Im Vergleich zu seinen Konkurrenten hat er einen geringen Stromverbrauch, was sich in seiner geringeren Hitzeentwicklung äußert. Schließlich besitzt der PIII auch noch die ausgereifte MMX- und SSE-Technologie (Prozessorkerninterne Optimierungen für den MultiMedia- und Streamingbereich), die durch die Verwendung geeigneter Kompiler zusätzliche Optimierungen ermöglichen. Als Nachteile kann die maximal erzielbare Transferrate von 1GB/sec zwischen CPU und RAM aufgrund der Architektur angeführt werden, die sich besonders im SMP-Betrieb äußert. Dort müssen sich nämlich beide CPUs ein Gigabyte pro Sekunde teilen.
Der direkte Konkurrent zum PIII ist der *Athlon* von AMD^{TM}, der durch seinen Preis und die Geschwindigkeit besticht, die besonders im Coprozessor-Bereich auffällig ist. Er ist im Bereich von 500 MHz bis 1,1 GHz erhältlich und kann aufgrund der Punkt-zu-Punkt-Einbindung der CPUs 2.1GB/sec an Daten transferieren. Ein Nachteil dieses Chips ist die starke Temperaturentwicklung während des Dauereinsatzes, da sein Stromverbrauch bedeutend höher ist als der seiner Konkurrenz. Zum Zeitpunkt der Planung war dieser Prozessortyp zwar SMP-fähig, allerdings gab es weder Motherboard-Chipsätze, die mehrere AMD-Prozessoren unterstützten, noch Hersteller, die deren Fertigung innerhalb kurzer Zeit geplant hatten. Die Athlon-Technologie ist wegen der Punkt-zu-Punkt-Verbindungen der CPUs zum Motherboard besser als der PIII auf den Dual-Betrieb vorbereitet und kann daher gleichzeitig mehrere Prozessoren mit 2,1GB/sec ansprechen.
Die dritte in Frage kommende Prozessorgeneration ist Intel^{TM}s *P4-Familie*, deren Entwicklung während der ersten Recherchen noch nicht ausgereift war. Der Unterschied beim Prozessortakt im Vergleich zur Leistung war eher gering und konnte keinesfalls den beträchtlichen Preisunterschied rechtfertigen. Der Preisunterschied lag nicht nur beim Prozessor, sondern auch bei den kaum vorhandenen, dafür aber überteuerten und teilweise fehlerhaften Motherboards und deren RAMBUS-Bestückung (siehe Arbeitsspeicher). Mittlerweile hat sich die Situation durch die rasante Entwicklung des P4-Prozessors und

dessen Komponenten verbessert. Die Taktfrequenz dieses Prozessors hat im Vergleich zu den damaligen 1,3GHz die 2GHz-Grenze bereits überschritten. Die Daten werden mit einer Geschwindigkeit von 400MHz zwischen CPU und RAM transferiert, wobei dabei 3,2GB an Daten pro Sekunde ausgetauscht werden.

Arbeitsspeicher (RAM) Je nach gewählter CPU kommen unterschiedliche Technologien beim Arbeitsspeicher zum Einsatz. PIII und Athlon kommen mit SDRAMs (Synchronous Dynamic Random Access Memory) aus, die mit 100 bzw. 133MHz Taktfrequenz betrieben werden und nur zur steigenden Flanke Daten übernehmen können. "Dynamic Random Access Memory" heißt diese Technologie deshalb, weil die Speicherstellen stets lesend aufgefrischt werden müssen, um den Dateninhalt erhalten zu können. Bemerkenswert ist bei den SDRAMs die optimale Abstimmung auf die PIII-Familie bezüglich des Datentransfers zwischen RAM und CPU. Der SDRAM-Baustein liefert genau 1GB Daten pro Sekunde, das eine PIII-CPU und dessen BUS-System aufzunehmen imstande ist. Für SDRAM spricht außerdem das Preis-Leistungsverhältnis, das jedoch wieder relativiert wird, wenn die Größe der benötigten RAM-Module im Gigabyte-Bereich liegt. Es ist jedoch wichtig, daß der Arbeitsspeicher eine hohe Qualität aufweist und die ECC-Technologie (Error Checking and Correcting) implementiert. Damit werden hardwaremäßig laufend Checksummen über den Speicherinhalt gebildet, sowohl SingleBit-Fehler als auch MultipleBit-Fehler (ein bzw. mehrere Bits im Speicherregister weichen vom eigentlichen Speicherinhalt ab) detektiert und eventuelle SingleBit-Fehler automatisch und transparent korrigiert.
Als alternative Verbesserung zu den gewöhnlichen SDRAMs sind noch DDR-SDRAMs (Double Data Rate-Synchronous DRAM) zu erwähnen, die im Gegensatz zu den gewöhnlichen SDRAMs auf beiden Taktflanken Daten übernehmen können. Damit verdoppelt sich der Datendurchsatz, bringt aber keine Leistungserhöhung wegen der BUS-Limitierung des PIIIs.
Für die P4-Prozessorfamilie werden derzeit ausschließlich Motherboards hergestellt, die RAMBUS-RAMs (RAM der Firma RAMBUS^{TM}) unterstützen. Damit kann eine be-

deutende Erhöhung des Datendurchsatzes zwischen CPU und RAM erzielt werden, da RAMBUS-RAMs im Gegensatz zu den 133MHz des SDRAMs mit bis zu 800MHz Speichertakt beim Datentransfer betrieben werden können. Diese RAM-Technologie wurde von Intel^{TM}zum de facto Standard ausgerufen und verwendet, was den beträchtlichen Preis erklärt.

Festplatten-Subsystem Bei der Festlegung des Festplatten-Subsystems kommt es auf die verwendete Schnittstelle (Interface) an, die den Kommunikationskanal zwischen Festplatte und dem übrigen System aufbaut. Zum einen kann die bei PCs gängige IDE/ATA-Technologie (Intelligent Drive Electronics oder Integrated Drive Electronics) verwendet werden. Diese implementiert ein Standardinterface für Festplatten, bei dem der Kontroller zur Ansteuerung des Gerätes in der Festplatte integriert ist. IDE/ATA ist wegen seiner weiten Verbreitung im Vergleich zu seiner Konkurrenz sehr günstig und erreicht je nach Implementierung eine maximale Transferrate zwischen 66 und 100MB/s. Damit ist aber nur das Interface mit einer derartig hohen Transferrate spezifiziert. Gängige IDE-Platten erreichen nicht mehr als 30 bis 40MB/s, was zur Folge hat, daß ein Gerät allein die Kapazitäten des Interfaces nicht ausschöpfen kann. Will man mehrere Geräte an den IDE-BUS anhängen, erreicht man unmittelbar die Grenzen des Interfaces.
Das SCSI-Interface (Small Computer Systems Interface) spezifiziert im Vergleich zum IDE ein paralleles, technisch allgemeineres und komplexeres Systeminterface zur Ansteuerung von beliebigen Geräten, wobei hier die Ansteuerung des Gerätes ein externer Kontroller übernimmt. Damit muß zum Beispiel jede Bewegung der Leseköpfe einer Festplatte vom Kontroller über das SCSI-Kabel zur Platte transferiert, um dann exekutiert zu werden. Dies zieht zwar teilweise einen Kommunikationsoverhead nach sich, stellt aber nicht wirklich einen Flaschenhals dar, da der BUS mit einem Takt von 80MHz betrieben wird. Die maximal erreichbare BUS-Transferrate beträgt so 160MB/s.
Was die Plattentechnologie selbst betrifft, ist die Qualität der SCSI-Platten in Bezug auf die Lebensdauer und Zugriffszeit besser. Der Grund liegt darin, daß die Plattenhersteller bei SCSI-Geräten neuere und bessere Technologien z.B. beim Positionieren auf den Plat-

ten einsetzten. Im High-Performance-Bereich sind SCSI-Geräte zu bevorzugen, da sie im Multiuser-Multitasking-Betrieb parallel angesprochen werden können. Weitere Vorteile von SCSI, die aber beim Cluster keine wesentliche Rolle spielen, sind die leichte Erweiterbarkeit von anderen internen und externen Geräten und die Möglichkeit, bis zu 15 Geräte an denselben SCSI-Kanal hängen zu können. (Vergleiche dazu [23])
Abschließend kann zusammengefaßt werden, daß IDE eine vor allem günstige und performante Lösung für das Plattensubsystem darstellt, wenn nur eine Platte verwendet wird und nicht massiv im Multi-User-Betrieb zum Einsatz kommt. Für den High-End-Bereich ist SCSI mit seinen preislichen Folgen immer noch dem IDE-Interface vorzuziehen. Für die Rechenleistung ist SCSI ebenfalls besser, da die Input/Output-Arbeit nicht, wie bei IDE, die CPU übernimmt, sondern auf den SCSI-Kontroller ausgelagert wird.

Umgebung Unter Umgebung kann das gesamte Umfeld um den Prozessor, Motherboard und RAM verstanden werden. Es ist von Vorteil, wenn die Gehäuse der Knoten eine gute Verarbeitungsqualität aufweisen. Muß man nun den Cluster hardwaremäßig verändern, so passiert das meistens bei allen Knoten. Dabei ist ein einfaches Handling beim Öffnen der Gehäuse nützlich. Mit dem Gehäuse ist unmittelbar das Netzteil verbunden, das dem Dauerbetrieb standhalten muß. Aus diesem Grund ist ein 400 Watt-Industrie-Netzteil mit guter Durchlüftung angemessen.
Um einen 24-Stunden-Dauerbetrieb überstehen zu können, kommen aktive Lüftungsteile, wie CPU-Kühler und Netzteilbelüftung, zum Einsatz. Dabei wird in der Regel auf kugelgelagerte Qualitätslüfter zurückgegriffen, um der vorzeitigen thermischen Zerstörung der CPUs vorzubeugen.
In diesem Zusammenhang muß auch noch auf Wartungsarbeiten hingewiesen werden, die jährlich durchgeführt werden sollen. Aktive rotierende Teile müssen auf ihre Funktionstüchtigkeit überprüft und bei Bedarf ausgetauscht werden. Eine Reinigung des Inneren der Knoten, z.B. mit Preßluft, sollte alle zwei Jahre durchgeführt werden.
Zum Clusterumfeld gehört auch ein Kontrollterminal, das ein lokales Anschließen eines Monitors und einer Tastatur auf jedem Knoten ermöglicht. Man kann an jeden Kno-

ten ein Monitor- und ein Tastaturkabel anschließen, um über eine zentrale Umschaltbox jeden Rechner direkt erreichen zu können. Allerdings reicht es vollkommen aus, eine Verlängerung für Monitor und Tastatur zu verwenden, um bei Bedarf lokal am Knoten arbeiten zu können. Im Regelfall wird das aber nicht notwendig sein, da sämtliche Knoten übers Netzwerk ferngesteuert werden können. Eine Umschaltbox ist zwar praktisch, aber rechtfertigt nicht die beträchtlichen Kosten (Verbesserung diesbezüglich siehe Seite 50).

Anmerkung Die hier getroffenen Aussagen müssen aufgrund der Schnellebigkeit des PC-Marktes und dessen rasanter Entwicklung relativiert werden. Es gab beispielsweise zum Zeitpunkt des Ankaufs noch keine Dual-Motherboards für Athlon-Prozessoren. Jetzt hingegen sind diese bereits erhältlich. Der Stand der Technologien und die getroffenen Aussagen sind vom Jänner 2001.

2.1.2 Master

Die Hardware des Masters muß möglichst redundant ausgelegt sein, da dessen Versagen den Ausfall des gesamten Systems nach sich ziehen würde. Dabei muß zusätzlich auf die Sicherheit der Userdaten besonderes Augenmerk gelegt werden. Von der Rechenleistung her reicht es aus, wenn der Master wie ein Knoten ausgestattet ist. Allerdings sollte er wegen seines postulierten Dauereinsatzes mit redundanten Netzteilen und zusätzlichen Lüftern ausgestattet werden. Diese Maßnahmen werden deshalb notwendig, da im Master zur Systemplatte zusätzliche Platten eingebaut werden, um genügend Platz für die Benutzerdaten zur Verfügung stellen zu können. Diese produzieren Wärme, die nach außen abgeführt werden muß. Was die Datensicherheit im System- und Userbereich betrifft, ist ein RAID-System (Redundant Array of Indipendent Disks) notwendig, um mit genügend hoher Geschwindigkeit Daten liefern und redundant abspeichern zu können. Diese Redundanz kann sowohl mit Hardware als auch mit Software erzielt werden, wobei in einem derartigen System aufgrund des hohen Datendurchsatzes und der schnelleren Zugriffszeiten eine Hardwarelösung vorzuziehen ist. Ein Kriterium für die Auswahl des Kontrollers ist die mögliche Übertragungsgeschwindigkeit des RAID-Systems und

die optimale Softwareunterstützung der Hardware. Firmen, wie Adaptec^{TM}, Mylex^{TM}, IBM^{TM}und Vortex^{TM}bieten derartige SCSI-RAID-Kontroller an, deren Durchsatzdaten aber beträchtlich variieren. Auf die besondere Netzwerkanbindung des Masters wird in den folgenden Kapiteln noch genauer eingegangen.

2.1.3 Netzwerk

Das Netzwerk spielt eine zentrale Rolle im Clusterverbund und bedarf deshalb besonderer Aufmerksamkeit. Die Aufgaben, welche das Netzwerk erfüllen muß, sind:

- Fileservice: Zentral gehaltene Benutzerdaten müssen auf allen Knoten zur Verfügung gestellt werden, was bei Unix- Systemen meist über das NFS-Protokoll (NetworkFileService) geschieht. Das dafür verwendete Übertragungsprotokoll basiert auf einer verbindungslosen UDP-Kommunikation zwischen Knoten und Fileserver und kann die Bandbreite der Netzwerktechnologien weitgehend auslasten, wenn mehrere Knoten gleichzeitig mit Daten versorgt werden wollen. Deshalb sollte das Fileservice für die großen Datenmengen der Bioinformatik auf einen dezidierten Server oder auf NFS-Filerhardware ausgelagert und auch das dafür verwendete Fileservice-Netzwerk vom Kommunikations-Netzwerk getrennt werden.

- Berechnungskommunikation: Fast jede parallelisierte Software basiert auf einer Kommunikationsbibliothek, die dem Programmierer Arbeit zur Implementierung der knoteninternen Kommunikation abnimmt. Sei es nun MPI oder PVM, die Kommunikation wird ebenfalls über das Netzwerk abgewickelt und belastet es zusätzlich.

- Userverwaltung, Überwachung, Job-/Batch-System: Soll die Überwachung der Knoten und der darauf laufenden Software zentral am Master geschehen, so ist damit zusätzlicher Kommunikationsaufwand verbunden und fällt ebenfalls dem Netzwerk zur Last. Diese Belastung hält sich allerdings im Vergleich zu den ersten zwei Punkten in Grenzen.

Um diese Dienste des Netzwerkes auch anbieten zu können, bedarf es einer sinnvollen Kombination folgender Faktoren:

- Hardware: Je nach Netzwerktechnologie müssen Netzwerkkarten verwendet werden, die vom Betriebssystem in geeigneter Form unterstützt werden. Diese Netzwerkkarten weisen unterschiedliche Betriebsarten auf (full-/halfduplex), wobei die optimale Betriebsart mit der gegebenen Netzwerkhardware erreicht werden soll. Es muß die Entscheidung getroffen werden, wie die Topologie des Netzwerkes ausschauen soll. Eine Möglichkeit ist die sternförmige Verkabelung in einem Switched-Netzwerk (Switch verteilt Pakete je nach Empfängeradresse), die im Gegensatz zum BUS-basierten Netzwerk oder zu einer Ringverkabelung, wie es Tokenring verlangt, steht. In diesem Zusammenhang muß auch festgelegt werden, welche Verkabelung verwendet wird, d.h. welches physikalische Medium die Knoten untereinander verbindet, und wo Engpässe im Netzwerk beim Clusterbetrieb entstehen könnten. Unüblich zum normalen Netzwerkbetrieb wollen alle Knoten aufgrund ihrer identen Konfiguration gleichzeitig kommunizieren. Solche sogenannten Hot-Spots sollten möglichst vermieden werden.

- Betriebssystem: Je nach Betriebssystem und unterschiedlicher Technologie kommt es darauf an, wie die Kommunikation und der Zugriff auf das Übertragungsmedium intern implementiert ist. Schafft es das Betriebssystem nicht, die Daten mit ausreichender Geschwindigkeit auf das physikalische Medium zu übertragen, kann eine eigene Implementierung des Netzwerkzugriffes und der integrierten Cluster-Kommunikation realisiert werden. (Näheres dazu auf Seite 44).

- CPU-Auslastung: Die CPU sollte während der parallelen Berechnungen möglichst gut ausgelastet sein und nicht andauernd auf Daten, die über das Netzwerk geliefert werden, warten müssen. Damit ist die Software gefordert, die vorhandenen Möglichkeiten des Netzwerkes optimal auszunützen. Asynchroner Datentransfer und sinnvolle Größe der Aufgabenverteilung helfen dabei, um nicht andauernd an die Grenzen des Netzwerkes zu stoßen.

- Engpässe: Künstlich erzeugte Engpässe im Netzwerk müssen vermieden werden. Die Cluster-Kommunikation sollte nie über einen einzelnen Gatewayknoten, sondern immer direkt von Knoten zu Knoten geführt werden.

  Die Hardware, z.B. Switches, die die Netzwerkverbindung schafft, sollte bei einer gleichzeitigen Belastung der Knoten nie die eigenen Grenzen erreichen, sondern immer die volle Kommunikation, die das verwendete Übertragungsmedium zuläßt, für alle Knoten zur Verfügung stellen.

Den Abschluß dieser allgemeinen Überlegungen bilden nun meßbare Größen, mit denen Netzwerktechnologien charakterisiert werden können:

- Latenzzeit: Unter Latenzzeit versteht man jene Zeit, die ein Paket braucht, um von einem Ort zum anderen zu kommen. Die Bestimmung dieser Zeiten kann auf zwei Arten erfolgen:

  Die sogenannte *Einweg-Latenz* kann dadurch ermittelt werden, indem ein Knoten einem zweiten Knoten andauernd und so schnell als möglich Datenpakete schickt. Der zweite Knoten ist ausschließlich damit beschäftigt, die vom ersten Knoten gesendeten Pakete entgegenzunehmen und deren Empfang zu bestätigen.

  Die zweite Methode ist die sogenannte *Ping-Pong-Methode*. Hierbei schickt ein Knoten ein leeres Paket zum zweiten, der dasselbe Paket sogleich zurückschickt. Die Zeit, die das Paket braucht, um hin- und hergeschickt zu werden, muß dann noch halbiert werden, um die Latenzzeit für einen Übertragungsweg zu ermitteln [32].

- Bandbreite oder Durchsatz: Die Übertragungskapazität eines Netzwerkes wird als Bandbreite bezeichnet und beschreibt die mögliche Datenmenge, die in einer Sekunde über das Netzwerkkabel geschickt werden kann. Sie wird daher auch in Megabit/Sekunde (Mbps) angegeben. Man kann zum Beispiel mit der Ethernet-Technologie 10Mbps übertragen und mit FastEthernet 100Mbps.

Ethernet und FastEthernet

Ethernet ist die weltweit am weitesten verbreitete LAN-Technologie (Local Area Network, auf geographisch kleinem Raum verteiltes Netzwerk), die ein relativ ausgewogenes Verhältnis zwischen Geschwindigkeit, Kosten und Installationsaufwand aufweist. Ihre weltweite Akzeptanz in der Computerbranche und die Möglichkeit unterschiedliche Netzwerkprotokolle, wie z.B. TCP/IP und IPX, gleichzeitig über dasselbe Medium zu transferieren sind die wesentlichen Vorzüge des Ethernets. Das "Institut for Electrical and Electronic Engineers" (IEEE) hat sowohl für die Konfiguration des Ethernet-Netzwerkes als auch für die Zusammenarbeit und Kommunikation einzelner Komponenten eine Spezifikation (IEEE 802.3) definiert, die bei Einhaltung der IEEE-Vorgaben einen äußerst effizienten Betrieb zulassen.

Ethernet kann als BUS-System realisiert werden, d.h. es greifen alle Rechner auf das-

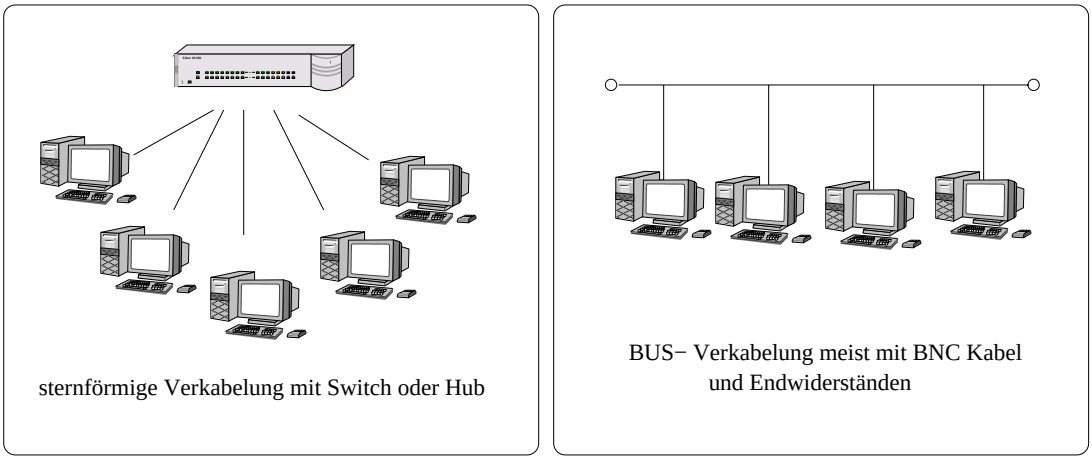


Abbildung 2.1: Verkabelungsmöglichkeiten bei Ethernet

selbe Medium gleichzeitig zu. Senden mehrere Rechner gleichzeitig, kommt es zu einer Kollision, die vom Sender erkannt werden muß. Tritt eine Kollision auf, so muß das bereits gesendete Paket erneut gesendet werden. Diese Art von Kommunikation nennt man *Carrier-Sense-Multiple-Access with Collision-Detection* (CSMA/CD)[1]. Die zweite und bessere Variante, ein Ethernet-Netzwerk zu strukturieren, ist die sternförmige Verkabelung. Dazu braucht es nicht, wie beim BUS-System, ein gemeinsames Kabel, sondern eine

zentrale Hardware, welche die Paketvermittelung übernimmt. Diese zentrale Einheit kann einerseits von einem Hub übernommen werden, der sämtliche Pakete an alle Anschlüsse (Ports) weiterleitet, andererseits von einem Switch, der hardwaremäßig unterscheidet, an welche Adresse im lokalen Netz das geschickte Netzwerkpaket adressiert ist. Damit leitet der Switch das Paket ausschließlich an den entsprechenden Port, an dem der Adressat hängt, weiter. Dies erfordert im Vergleich zum Hub einen höheren Hardwareaufwand bei der Herstellung des Switches, da dieser sämtliche Pakete mit voller Mediumgeschwindigkeit von 10Mbps annehmen, untersuchen und weiterleiten muß.
FastEthernet higegen ist eine Netzwerktechnologie, deren Ursprung beim Ethernet zu finden ist, und wird dieses auch in naher Zukunft ablösen. Auch sie wurde von IEEE (802.3u)[1] spezifiziert und erlaubt eine Übertragungsgeschwindigkeit von 100Mbps. FastEthernet hat den besonderen Vorteil, mit wenig Aufwand ein vorhandenes strukturiertes Ethernet in ein FastEthernet-Netzwerk umzuwandeln. Dafür müssen FastEthernet-fähige Netzwerkkarten verwendet werden, wobei angemerkt werden muß, daß mittlerweile alle gängigen Netzwerkkarten bereits Ethernet und FastEthernet parallel unterstützen. FastEthernet kann nur noch sternförmig aufgebaut werden, wobei alle Knoten über einen Switch verbunden werden sollen.

Verkabelung

Die IEEE-Spezifikation erlaubt bei FastEthernet mehrere Arten von Verkabelungen, wobei grob zwischen Kupferverkabelung und Glasfaserverkabelung unterschieden werden kann. Im Ethernet-Bereich ist hauptsächlich Kupferverkabelung im Einsatz, bei der sich fünf Kabeltypen unterscheiden lassen. Die Daten werden über Kupferlitzen übertragen, die wegen der gegenseitigen magnetischen Beeinflussung verdrillt sein müssen (Twisted-Pair). Je nach Anzahl der Kupferleitungspaare und der Qualität der Litzen können über die telefonkabelartigen Leitungen unterschiedliche Entfernungen überwunden und Bandbreiten erzielt werden.
Für FastEthernet sind allerdings nur zwei der fünf Kabeltypen verwendbar, da mit den anderen der hohe Datendurchsatz von 100Mbps nicht erreichbar ist. Die wichtigste,

aber teurere Variante ist die sogenannte 100Base-TX-Technologie mit UTP-Kategorie-5-Verkabelung (Unshielded Twisted Pair), bei der nur 2 Leitungspaare für die Datenübertragung zuständig sind. Die billigere Variante sind Kategorie-4-UTP-Kabel, die nur 20Mbps übertragen können. Diese Kabel besitzen meist ein weiteres Litzenpaar, das normal nicht verwendet wird. Verwendet man nun Kategorie-4-Kabel als 100Mbps Verbindungen, so werden alle 4 Litzen, wie im Standard 100Base-T4 spezifiziert, verwendet. Damit können 100Mbps gerade noch erreicht werden. Ein weiterer Ausbau, wie bei Kategorie-5-Verkabelung, auf Gigabit-Ethernet ist nicht mehr möglich. Die zweite physikalische Verbindungsmöglichkeit ist die kostspielige Glasfaserverkabelung, die sogenannte 100Base-FX, welche den Vorteil hat, daß damit auch Gigabit-Verkabelungen durch Austausch der Netzwerkkarten realisiert werden können. Als weitere Vorteile, die beim Cluster aber nicht zum Tragen kommen, ist die Unempfindlichkeit gegenüber äußeren magnetischen Einflüssen und die viel größere Reichweite, die bis zu 2 Kilometer betragen kann. Dem hat 100Base-TX nur 200 Meter an Reichweite entgegenzusetzen.
Will man von allen Knoten auf den Master zugreifen, so reicht beim Master bzw. Fileserver eine 100Mbps-Verbindung nicht aus. Aus diesem Grund macht es Sinn, den entstehenden Netzwerkengpaß mit dem Nachfolger von FastEthernet, dem Gigabit-Ethernet, zu beseitigen. Theoretisch sollte er 1000Mbps schaffen, aber de facto stößt man an die Grenzen anderer PC-Komponenten und erreicht 300 bis 400Mbps.

Alternativen

Für das Fileservice kommen zum Gigabit- Ethernet noch Myrinet, SCI-Technologie sowie SAN-Lösungen in Frage. Auf diese Alternativen wird später in der Diskussion auf Seite 44 näher eingegangen.

2.1.4 Designentscheidungen beim TUG-Cluster

PC-Hardware: Für den Beowulf Cluster der TU Graz wurden normale Desktop-Systeme angekauft, die mit wenig Zusatzaufwand zu Arbeitsplatzmaschinen umfunktioniert wer-

den könnten. Falls das Pilotprojekt fehlschlagen würde oder ein Austausch von Knoten in einigen Jahren durchgeführt werden müßte, könnten deshalb die ausgemusterten Maschinen für TU-Institute weiterverwendet werden. Damit war unmittelbar ein erhöhter Platzbedarf verbunden, was aber bei der aktuellen Dimension des Clusters nicht so sehr ins Gewicht fiel. Zum Zeitpunkt der Planung fiel die Wahl der verwendeten Prozessoren

Abbildung 2.2: TU Graz Cluster nach dem Aufbau

auf Intel^{TM}s PIII mit 866MHz, da wegen seiner Dual-Fähigkeit mehr Prozessoren in einem Knoten verwendet werden konnten. Die ausgereifte Technologie und die Stabilität bei diversen Treibern beeinflußten die Entscheidung für diesen Prozessor. Der Athlon schied wegen seiner hohen Temperaturentwicklung und den nicht existenten Dual-Motherboards aus. Gegen den P4-Prozessor sprach der beträchtliche Preisanstieg eines einzelnen Knotens, was unmittelbar die Anzahl der möglichen Knoten im Gesamtsystem reduziert hätte, und seine ebenfalls hohe Temperaturentwicklung. Zum Teil wurde das durch die Verwendung des PIIIs eingesparte Geld in Arbeitsspeicher investiert, wobei ECC-SDRAMs zum Einsatz kamen. Die Größe des RAMs wurde auf 1 Gigabyte festgelegt, um den Anfor-

derungen der Bioinformatik zu entsprechen und möglichst viele Daten in der Nähe des Prozessors halten zu können. Dies gelang zum einen mit genügend RAM, zum anderen mit entsprechend großen Platten, um Datenbanken mit den damit verbundenen Nachteilen des Wartungsaufwandes lokal auf jeden Knoten auszulagern. Dabei mußten die Daten schnell und möglichst ohne Belastung der CPU erreicht werden, was ein SCSI-Plattensubsystem den Vorzug gab. Um die Komponenten möglichst kompakt zu halten, fiel die Wahl des Motherboards auf ein Modell von ASUS^{TM}(CUR-DLS), das zwei PIII-CPUs und bis zu 4GB RAM beherbergen kann, aber auch SCSI-Kontroller, Grafikkarte und Netzwerkkarte in sich vereint. Daher lag die Entscheidung nahe, SCSI als überlegenes Plattensubsystem-Interface zu verwenden, um den integrierten SCSI Kontroller nicht zu vergeuden.

Master Der Master wurde, wie die übrigen Knoten, mit dem ASUS-Motherboard und den PIII-Prozessoren ausgestattet, erhielt aber die notwendigen Zusatzkomponenten, um seinen zusätzlichen Aufgaben gerecht zu werden. Die idente Hardware hat den Vorteil, daß jeder Knoten als Ausfall-Master zur Verfügung steht, wenn Teile des eigentlichen Masters ausfallen würden. Diese Strategie war bereits erfolgreich, da während der intensiven Testphase das Motherboard des Masters, von dem die Funktionsfähigkeit des gesamten Clusters primär abhängt, seinen Dienst versagte. Innerhalb einer halben Stunde war der Master mit dem Motherboard eines Knotens einsatzfähig und konnte wieder benutzt werden.

Die Konfiguration des Masters unterschied sich nur durch die 1,5GB an Arbeitsspeicher und dem zusätzlichen 2-Kanal-Vortexkontroller, der sämtliche Hardware-RAID-Funktionalitäten (RAID 0 bis 10) und 128MB Zwischenspeicher (Cache) zur Verfügung stellte.

Am ersten Kanal des Kontrollers hingen 3 Systemplatten zu je 18GB, wobei zwei von ihnen zu einem gespiegelten Volume (Platte) vereint wurden. Damit wurde sowohl der Ausfallsicherheit Rechnung getragen, da die Platten durch den Spiegel komplett idente Datenbestände aufweisen, als auch der Leseperformance, da beim Lesen beide Platten

mehreren Prozessen gleichzeitig Daten von unterschiedlichen Plattenpositionen liefern können. Dies erhöhte die Geschwindigkeit beim lesenden Zugriff beträchtlich, wobei wegen der Sicherheit noch die dritte Platte als Ausfallplatte zum Einsatz kam. Falls eine der Systemplatten ausfallen würde, könnte der Kontroller den Bestand der funktionierenden Platte verwenden, um automatisch mit der dritten Platte den Systemspiegel aufzubauen. Da die dritte Platte nicht im aktiven Dauereinsatz ist, wird deren Mechanik nicht so abgenutzt, wie die der beiden anderen. Die Wahrscheinlichkeit eines gleichzeitigen Ausfalles aller drei Platten wird damit bedeutend verringert.

Die Benutzerdaten hingegen werden auf fünf 70GB große Platten im RAID-5-Verbund abgelegt, die am zweiten Kanal des SCSI-Kontrollers angehängt wurden. RAID-5 gibt ein hohes Maß an Sicherheit, da dort die Daten stückchenweise zirkulär auf allen Platten verteilt werden. Hierbei werden die Checksummen über die Daten auf unterschiedlichen Festplatten abgelegt, mit denen beim Ausfall einer solchen Platte die Daten wiederhergestellt werden können. Die gesamte Kapazität der Platten beträgt nun aufgrund der Redundanzverluste 280GB und war vorerst auf die Beherbergung der notwendigen Gendatenbanken ausgelegt (mehr dazu auf Seite 25).

Netzwerk Aus Kostengründen konnte eine Gigabit-Verkabelung nicht angekauft werden und so mußte auf eine strukturierte FastEthernet-Verkabelung zurückgegriffen werden. Der zentrale 48-Port-Switch Cisco^{TM}2948G verwaltet den Netzwerkverkehr der mit Kategorie-5-UTP-Kabel verbundenen Rechenknoten. Nur die Fileserver, auf die alle Knoten über das Netzwerk zugreifen, wurden am Switch mit einem Gigabit-Interface verbunden, da dort der Flaschenhals lokalisiert werden konnte. Um mit dem vorhandenen 100-Megabit-Netzwerk trotzdem eine höhere Netzwerkleistung erzielen zu können, wurde eine Trennung des Kommunikations- und des Fileservice-Netzes durchgeführt. Ein zusätzlicher FastEthernet-Adapter zur "onboard"-Netzwerkkarte erhöhte die Kosten eines Knotens nur gering und verhinderte eine vorzeitige Verstopfung des Netzwerkes mit gleichzeitigem File- und Kommunikationstransfer.
Mit dieser Architektur kann nun jeder Knoten auf zwei Wege erreicht werden. Sei es

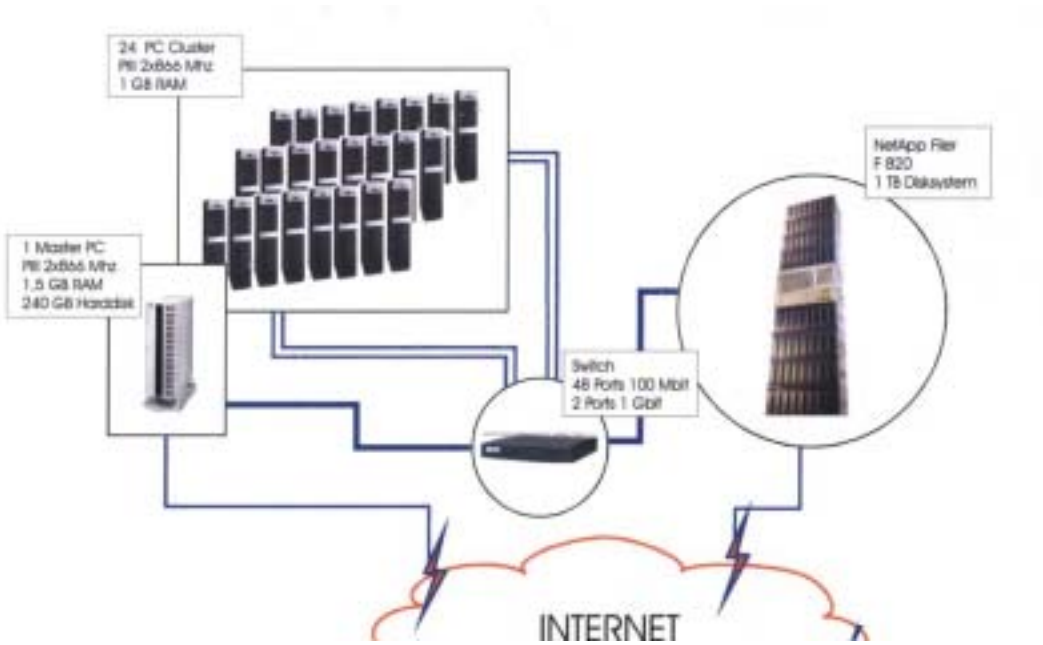


Abbildung 2.3: TU Graz Cluster im Überblick

aufgrund von Treiberproblemen, die zwei unterschiedliche Netzwerkkarten selten gleichzeitig haben, sei es aufgrund von Kabeldefekten, es kann beim Ausfall einer Verbindung über die zweite Karte immer noch am Knoten konfiguriert und repariert werden. Das ist im Falle des Clusters von Vorteil, da man ansonsten zu jedem Knoten hingehen und die Terminal-Kabel anstecken muß.

Als limiterender Faktor für die Anzahl der Clusterknoten erwies sich bei dieser Topologie der Switch, da nur 24 Knoten angehängt werden konnten. Allerdings waren für die erste Ausbauphase des Clusters nur 16 Knoten geplant. Als die Angebote konkret vorlagen und die tatsächliche Finanzierung anstand, konnte doch noch mit zusätzlichen Geldmitteln der Cluster auf die momentane Ausbaustufe von 24 Knoten angekauft werden. Damit ist nun die momentane Ausbaubarkeit mit dem Ausbau der Netzwerkinfrastruktur verbunden (siehe Diskussion Seite 44).

Da der gesamte Netzwerkverkehr über den Switch Cisco^{TM}2948G abgewickelt wird, mußte sichergestellt werden, daß dieser nicht zum Schwachpunkt der Gesamtkonfiguration werden würde. Mit diesem Switch konnte ein sternförmiges FastEthernet-Netzwerk für

die Knoten realisiert werden, in dem jeder der 48 Ports mit seiner vollen Leistung von 100Mbps angesprochen werden kann. Zusätzlich ermöglichten die zwei Gigabit-Ports des Switches die Anbindung der Fileserver mit Gigabit-Ethernet. Daß die Bandbreitenkapazität des Switches mit den 48 mal 100Mbps und 2 mal 1000Mbps nicht erreicht wird, sorgt die auf Durchsatz optimierte sogenannte "switching fabric", die 24Gbps nicht blockierend transferieren kann. Damit wurde sichergestellt, daß alle Ports mit der vollen Bandbreite von 100Mbps bzw. 1000Mbps, die ihnen das Übertragungsmedium zur Verfügung stellt, gleichzeitig ihre Daten senden und empfangen konnten.

Softwaremäßig mußte beim Netzwerk besonders darauf geachtet werden, daß sowohl Switch als auch Netzwerkkarte im fullduplex Mode und mit fix eingestellten 100Mbps miteinander kommunizierten. Dies war deshalb notwendig, da sowohl Switch als auch Netzwerkkarte ansonsten bei der Initialisierung nicht richtig arbeiteten, denn die Ports am Switch wechselten bei einem Neustart in einen nicht definierten Zustand. Zwang man aber die obgenannten Betriebsarten bei der Treiberinitialisierung der Netzwerkkarten und bei der Portkonfiguration am Switch der beteiligten Hardware fix auf, so funktionierte die Hardware problemlos.

2.1.5 Bemerkungen zum Aufbau und zur Konfiguration der Hardware

Der Aufbau des Clusters erfolgte im klimatisierten Rechnerraum, um die nötige Umgebungstemperatur für den Dauerbetrieb gewährleisten zu können. Beim Aufbau war besondere Sorgfalt auf die Verkabelung zu legen. Deshalb wurde sowohl bei der Strom- als auch bei der Netzwerkverkabelung systematisch vorgegangen und jedes Netzwerkkabel entsprechend beschriftet.

2.1.6 NetApp-Filer

Aufgrund der bereits mehrfach erwähnten, riesigen Datenmengen und deren notwendigen raschen Transfers zu den Knoten mußte ein erweiterbares Festplattensystem in der Um-

Abbildung 2.4: strukturierte Strom- und Netzwerkverkabelung

gebung des Clusters positioniert werden.

Im Laufe der Planung und des Aufbaues des Clusters wurde die Notwendigkeit eines zentralen Speichers erkannt, der hardwareoptimiert den Zugriff auf die Daten vornimmt und sie so schnell wie möglich auf das Netz verteilt. Deshalb wurden bereits in der Aufbauphase Möglichkeiten zur Lösung dieses Problems gesucht und bei der Firma NetworkAppliance gefunden. Aus diesem Grund wurde ein Filer angekauft, der mit einem GigabitEthernet-Interface ausgestattet ist und damit 1TB Daten dem Cluster über NFS zur Verfügung stellt. Die beträchtliche Investition in diese Hardware wird sich allerdings auch dahingehend lohnen, daß zusätzlich zum schnellen Zugriff auf die über FibreChannel verbunden Festplatten diese bis auf 2TB erweiterbar sind. Die Zukunft des Clusters sollte damit nicht am Mangel an Festplattenplatz für die Gendatenbanken scheitern.

2.2 Software

Bei der Softwareinstallation eines Linux-Clusters fällt zum einen die Installation des Betriebssystems und dessen Verteilung auf sämtliche Rechenknoten (Cloning), zum anderen die Installation von Cluster- (Kommunikationsbibliotheken etc.) und Ressource-Verwaltungs-Software an. Es muß vorausgeschickt werden, daß diese Abhandlung nicht eine detaillierte Installationsanleitung ist, sondern nur einen groben Überblick über die Vorgehensweise beim Aufbau geben soll. Genauere Informationen und Anleitungen können im Anhang auf Seite 61 bei den Internet-Links gefunden werden.

2.2.1 Grundinstallation

Als Betriebssystem kam Linux zum Einsatz, das sich besonders auf dem Beowulf-Sektor etabliert hat. Für diese Plattform gibt es zahlreiche Programme, die dann im Clusterverbund ihre Berechnungen durchführen können. Linux an sich ist ein Open-Source-Betriebssystem [13], das von Linus Torvalds im Jahre 1991 ins Leben gerufen wurde, und sich aufgrund seiner freien Verfügbarkeit, Flexibilität und damit beliebigen Ausbaubarkeit stetig weiterentwickelt. Vielen freiwilligen Entwicklern ist es zu verdanken, daß Linux heute als ein ausgereifter UNIX-Derivat auftritt, das sowohl im Arbeitsplatzbereich aber noch viel mehr im Multi-User-Serverbereich seinen Platz gefunden hat. Linux ist eigentlich nur der Betriebssystemkern, der sogenannte Kernel, der den Anwenderapplikationen die Hardware nutzbar macht. Aus diesem Grund entstanden Firmen, die sowohl den Betriebssystemkern als auch alle notwendigen Zusatzapplikationen zu einem großen Paket zusammengeschnürt haben. Ein solches Paket wird Distribution genannt, und ist eine Ansammlungen von Programmen, die auf der Betriebssystem-Plattform Linux laufen. Die meisten der mitgelieferten Programme sind unter einer freien Lizenz erhältlich oder als kommerzielle Draufgabe vom Distributor erhältlich. Die Distributoren, z.B. RedHat, SuSE, Debian, Mandrake etc., wurden in den letzten Jahren für die Weiterentwicklung von Linux und dessen Applikationen immer wichtiger, da sie durch hauseigene Entwickler Projekte massiv vorantreiben. Ohne diese Unterstützung wäre Linux besonders im

Normalanwenderbereich bei weitem nicht so weit entwickelt. Mittlerweile unterscheiden sich die Distributionen größtenteils nur noch durch den Installations- und Wartungskomfort. Jede Distribution für sich besitzt eine eigene Administrator-Applikation, die mehr oder weniger die Installation und Konfiguration eines Systems erleichtert. Unterschiede sind auch in der Verbreitung zu erkennen, die sich hauptsächlich auf den Sprachraum zurückführen läßt. Während die amerikanische Distribution Redhat und dessen Derivat Mandrake eher im angloamerikanischen Raum Verbreitung finden, ist im deutschsprachigen Bereich der Distributor SuSE vorherrschend. Jeder von ihnen bemüht sich allerdings, möglichst viele Sprachen mit ihren Produkten abzudecken und entsprechend anzupassen. Da die Bioinformatik seine Wurzeln im amerikanischen Bereich hat und von dort die meiste Software herkommt, die auf diesem Sektor erhältlich ist, haben sich die Software-Hersteller auf RedHat festgelegt und deren Support auf diese Distribution beschränkt. Aus diesem Grund, und weil an der TU Graz RedHat-KnowHow bereits früher angesammelt wurde, fiel die Entscheidung bezüglich der Distribution auf RedHat.

Die Installation des Grundsystems wurde zweigeteilt durchgeführt. Der Master, der den Brückenkopf von außen zum Cluster bildet, mußte als normaler Arbeitsplatzrechner mit RedHat-Linux(Version 7.1) installiert werden. Da der Master bereits einen Netzanschluß hatte, konnte dabei auf die Netzwerkinstallation zurückgegriffen werden, bei der nur eine Bootdiskette benötigt wurde, und sich damit das gesamte System via FTP (FileTransfer-Protokoll) vom nächst gelegenen FTP-Server [31] holen konnte. Auf dieselbe Art wurde ein Knoten ausschließlich mit dem RedHat-Grundsystem ausgestattet, da dort keine Entwicklungsarbeit geleistet, sondern nur die Berechnungen durchgeführt werden. Sämtliche Bibliotheken, die für die Kommunikation und die massiven mathematischen Berechnungen notwendig sind, mußten nachträglich auf dem Knoten installiert werden. Damit konnte es vermieden werden, auf den Knoten unnötiges Programmier- und Kompilierwerkzeug, IDEs (Integrated Development Environment) etc. installiert zu haben.

Was die Installation von Software betrifft, wurde das RPM-Paketmanagement (RedHat Package Management) übernommen, da damit das Installieren, Deinstallieren und Updaten von Software, falls sie im RPM-Format vorlag, vereinfacht wurde. Zusätzliche Kom-

piler, wie z.B. der NAG-Fortran-Compiler [28], und Mathematikpakete, wie z.B. BLAS (Basic Linear Algebra 3.0-9), lapack (Linear Algebra PACKages 3.0-9) oder scalapack, rundeten die standardmäßige GNUC- und GNUFortran77-Entwicklungsumgebung ab.
Da Linux ein Multi-User-Betriebsystem ist, mußte eine einheitliche Userverwaltung über den gesamten Cluster eingerichtet werden, damit sich jeder User auf allen Knoten mit einem Username und einem Passwort "frei bewegen" konnte. Das allgemeinste und zukunftsträchtigste Verfahren dafür stellte das LDAP-Protokoll (Lightweight Directory Access Protocol) [12] zur Verfügung, das sämtliche Daten in einer Baumhierarchie ablegt und als einfaches Textprotokoll implementiert ist. Damit war dieses Protokoll prädestiniert für den plattformübergreifenden und über das Netzwerk verteilten Einsatz, da ausschließlich Textzeichen für die Datenübermittlung verwendet werden und es bereits vorgefertigte Authentifizierungsschemata für die interne LDAP-Baumstruktur gab. Die Authentifizierung an den Knoten erfolgt über das modulare System PAM (PluggableAuthentication-Module), bei dem im laufenden Betrieb jedes beliebige Authentifizierungssystem - über Konfigurationsfiles gesteuert - integriert werden konnte. Da die Entscheidung auf LDAP fiel, wurde nun das LDAP-PAM-Modul [17] eingesetzt, das dynamisch bei jeder Paßwortabfrage eines Programmes eingeklinkt wird und dann gegen den am Master installierten LDAP-Server authentifiziert.
Bei erfolgreichem Login mußte jedem Benutzer der Zugriff auf seine Daten, die im sicheren RAID-Verbund am Master abgespeichert sind (siehe Seite 14 bzw. 22), ermöglicht werden. Dafür wurden Teile des Masterfilesystems, wie in Grafik 2.5 beschrieben, quer über alle Knoten verteilt. Als Übertragungsprotokoll wurde das bereits beschriebene NFS-Protokoll (Network File System) verwendet, wobei es zwecks clientseitiger Optimierung notwendig war, den Betriebssystemkern anzupassen [38]. Mit dieser Optimierung und der Verwendung des serverseitigen, kernelinternen NFS-Servers konnte ein akzeptabler NFS-Zugriff der Knoten auf den Master erreicht werden. Damit das Gigabitinterface des Masters wegen der Standardeinstellungen nicht eingeschränkt wurde, mußten auch noch Anpassungen der Einstellungen auf TCP/IP-Ebene durchgeführt werden.
Für die optimale Nutzung wurde relativ viel Zeit in die Konfiguration der gegebenen

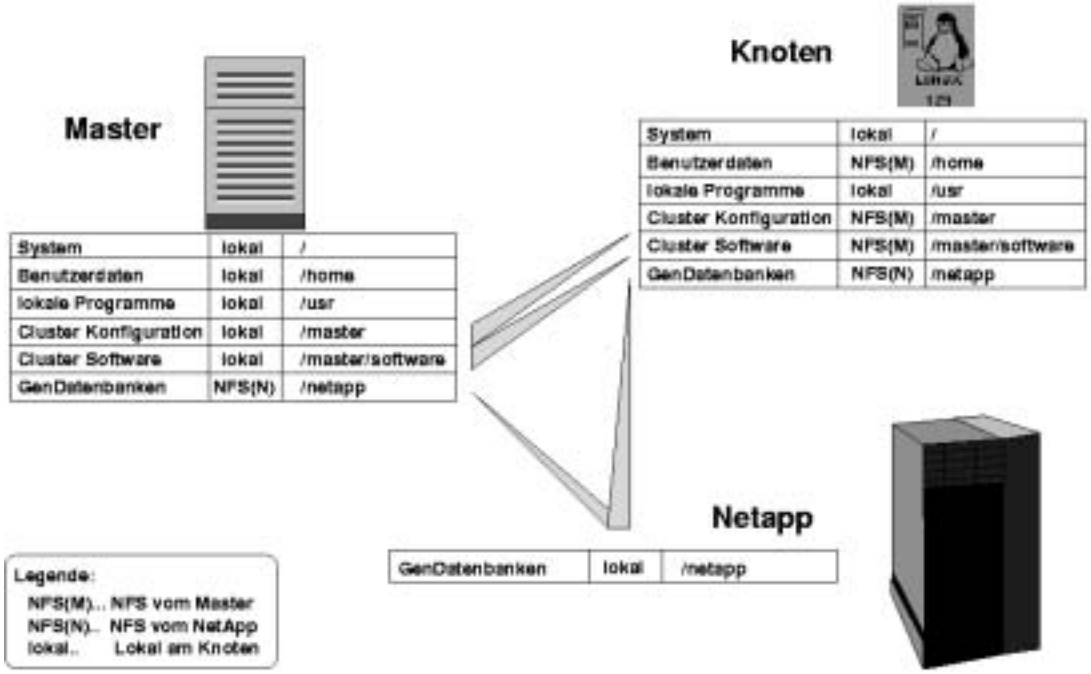


Abbildung 2.5: Filesystemübersicht und dessen Verteilung am gesamten Cluster

Netzwerkinfrastruktur investiert. Dabei war es notwendig, sowohl bei der Switchsoftware als auch bei den Treiber-Parametern des Kernels für die Netzwerkkarte fixe "full-duplex-" und "100MBit-"Einstellungen vorzunehmen. Dadurch konnten Netzwerkpakete mit voller Bandbreite und in beiden Richtungen gleichzeitig verschickt werden.

Als Betriebssystemkern auf dem Master und auf den Knoten wurde ein Kernel der 2.4er Serie installiert [35] [24]. Die Gründe für die Verwendung der 2.4er Kernel-Serie waren die verbesserte Speicherverwaltung, die optimierte Unterstützung von mehreren Prozessoren mit "Symmetric Multi Processing" und die verbesserte Implementation des NFS-Protokolles, die zum Teil massive Ausfälle und Verbindungsprobleme älterer Kernelgenerationen beseitigte. Leider schlichen sich besonders in den letzten Kernelversionen zahlreiche Fehler ein, was zur Folge hatte, daß Knoten zum Teil grundlos bei hoher Belastung ein Reboot auslösten und damit die gesamte Berechnung des laufenden Prozesses vernichteten. Als diese Zeilen verfaßt wurden, war der Kernel 2.4.10 im Einsatz, der mit diversen Zusatzerweiterungen (Patches) im Bereich der Speicherverwaltung versehen

werden mußte, um einen stabilen Betrieb zu garantieren. Obwohl zwei weitere Versionen bereits veröffentlicht wurden, war es unmöglich von diesem funktionstüchtigen Systemkern wegzugehen und auf neuere Versionen upzudaten.

2.2.2 Clonen

Nach der Vollinstallation des Masters und der Minimalinstallation eines Knotens wurde ein Verfahren entwickelt, um die abgeschlossene Konfiguration des Referenzknotens auf alle anderen Knoten verteilen und diese damit in den Clusterverband integrieren zu können.

Bootvorgang

Beim Starten eines Betriebssystems nach der Hardwareinitialisierung des "BasicInputOutputSystem" (BIOS) muß der Betriebssystemkern von einem Datenträger ausgelesen werden. Dafür werden die ersten Sektoren des Datenträgers verwendet, um den sogenannten Bootloader abzuspeichern, der anschließend den eigentlichen Betriebssystemkern oder Kernel nachlädt. Ist dieser Loader erst einmal im Arbeitsspeicher, wird bei Linux der absolut notwendigste Teil des Betriebssystems in den Speicher geholt. Erst bei Bedarf kann der Minimalkernel aufgrund seines modularen Aufbaues dynamisch mit Gerätetreiber und Zusatzfähigkeiten in Form von Kernelmodulen erweitert werden. Dies geschieht automatisiert während des Aufstartens der internen Systemdienste, welche die entsprechenden Erweiterungen benötigen. Es kann z.B. der Systemdienst Netzwerk nicht eingerichtet werden, wenn nicht zuerst der Gerätetreiber der Netzwerkkarte in den Kernel erfolgreich integriert wurde, etc. (vergleiche [40])
Ein besserer Einblick in die Architektur des Betriebssystems und in den modularen Aufbau kann anhand der Grafik 2.6 erzielt werden.

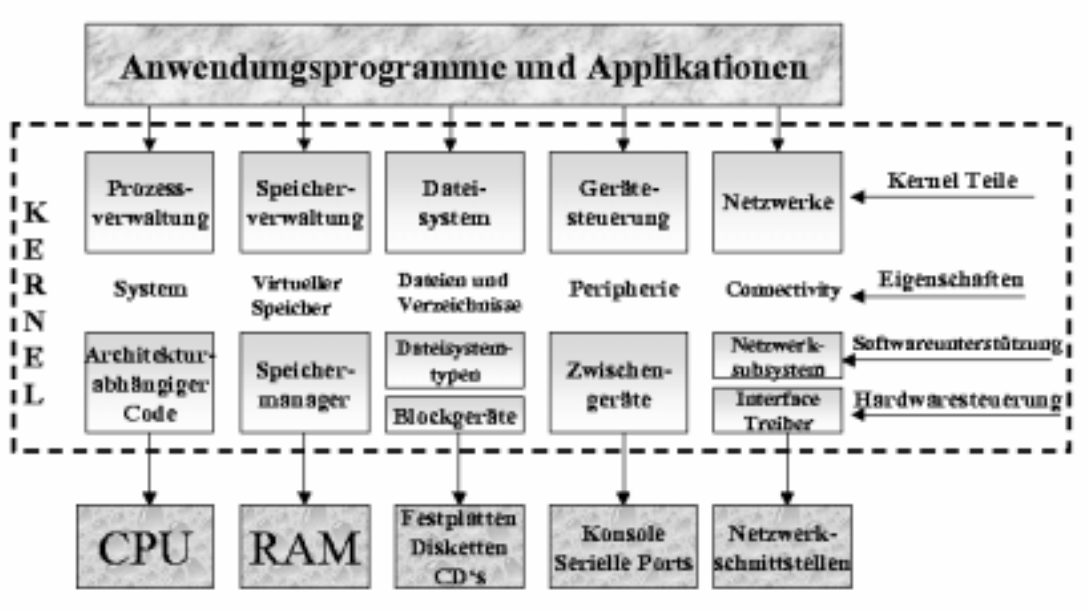


Abbildung 2.6: prinzipieller Aufbau des Linux-Kernels [39]

Installationsvorgang

Um eine problemlose und einfache Installationsmethode zu entwickeln, wurde ein zentralistischer Ansatz gewählt, der sich sämtliche Komponenten inklusive Betriebssystemkern vom Master über das Netz holt. Seinen Ursprung hat diese Methode bei den BootRoms diverser Netzwerkkarten, die einen speziellen Bootvorgang von Rechnern ermöglicht, wobei die benötigten Daten auch dort ausschließlich über das Netz geholt werden. Das Paket Etherboot [41] dient dazu, den Inhalt für dieses BootRom zu generieren, wobei die verwendete Netzwerkkarte von ihm unterstützt werden muß. Da bei der Clusterinstallation dieser Net-Boot-Vorgang nur bei der Installation erwünscht war, war es sinnlos, das tatsächliche BootRom der Netzwerkkarten zu beschreiben. Dieses Verfahren ließ sich auch zum Booten von Diskette bzw. Festplatte adaptieren. Die erzeugte BootRom-Datei mußte mit einem geeigneten Programm auf eine Diskette gespeichert und von dieser gebootet werden. Dabei wurde die im Rechner vorhandene Netzwerkkarte initialisiert und bezog via DHCP (Dynamic Host Configuration Protocoll) eine Internetadresse. Die Netzwerkkarte erhielt diese Adresse von einem am Master konfigurierten DHCP-Server [18],

der mit demselben Verfahren weitere sogenannte BOOTP-Felder dem Knoten übergab. Mit Hilfe dieser Felder, welche die Internetadresse des BootServers und die Position des Betriebssystemkernes am Server beinhalten, konnten alle Knoten den Kernel vom Server herunterladen und ihn dann booten.

Das gesamte Installations-Bootsystem, das sogenannte "root-system", konnte nach der Kernelinitialisierung vom Master via NFS erreicht werden (Root on NFS). Dort wurde das Installationsskript aufgerufen, welches die Formatierung der lokalen Festplatten, das Kopieren eines Abbildes vom Referenzknoten und diverse einmalige Registrierungen am Master durchgeführt hat. Zu diesem Zeitpunkt wurde eine im Cluster eindeutige Internetadresse (IP) für das private Netzwerk im 10.0.0.x-Bereich festgelegt, mit der der Knoten stets erreichbar sein sollte. Selbst nach einer Neuinstallation holten sich die Knoten über ihre Hardwareadresse der Netzkarten die beim ersten Mal erhaltene IP aus einem zentralen Knotenkonfigurationsfile. Wurde der Kopiervorgang abgeschlossen, so konnte der nicht netzwerkfähige Standard-Bootloader LILO lokal auf der Platte installiert werden. Unmittelbar nach einem Neustart war der installierte Knoten einsatzfähig. Um auch den Master von der Existenz bzw. von der Einsatzbereitschaft des Knotens zu informieren, wurde ein Skript beim Bootvorgang eingerichtet, das den Knoten mit IP beim Master anmeldet und beim Reboot wieder abmeldet. Auf diese Anmeldung beruhen sämtliche Shellskripts, bei denen versucht wird, Programme auf allen Knoten auszuführen. Dies geschieht durch die Installation eines RSH-Servers (Remote Shell Servers) auf jedem Knoten, der ohne Paßwortabfrage einen vom Master aus gestarteten Befehl am Knoten ausführt. Aus diesem Grund war die Abschottung des Clusternetzwerkes nach außen notwendig, da derartige Services im Internet aus sicherheitstechnischen Gründen inakzeptabel waren. Die Alternative dazu war der Verbindungsaufbau über einen verschlüsselten Kanal (SSH, Secure Shell), was den Nachteil in sich birgt, daß dieser Vorgang zu jedem Knoten beträchtlich länger gedauert hätte.

Um bei nachfolgenden Installationen nicht wieder eine Bootdiskette benutzen zu müssen, wurde der Bootloader LILO so installiert, daß auch die BootRom-Datei der Diskette als Bootoption zur Verfügung gestellt wurde. Für nachfolgende Neuinstallationen mußte die-

se Bootoption als Standard-Bootoption definiert werden und schon konnte der Knoten mit demselben System automatisch und ohne Überwachung neu installiert werden.

Anmerkungen

Auch im Normalbetrieb wäre DHCP das ideale Verfahren, um den Knoten dynamisch die IPs zuzuweisen. Nach zahlreichen Versuchen, diesen Lösungsansatz zu realisieren, mußte doch auf die oben beschriebene Vergabe fixer IPs bei der Erstinstallation zurückgegriffen werden. Dafür gab es zwei Gründe:

- Der Switch hatte bei einem gleichzeitigen Reboot aller Knoten Probleme, seine Ports zu initialiseren. Diese punktuelle Überlastung des Switches wurde hauptsächlich dadurch verursacht, daß alle 48 Netzwerkkarten gleichzeitig eine IP beim DHCP-Server beantragen wollten. Der Switch ist aber auf IP-Switching spezialisiert und kam teilweise in einen undefinierten Zustand, da zu diesem Zeitpunkt keine Netzwerkkarte an den Ports eine IP hatte.

- Gleichzeitig mit dem ersten Problem kam noch ein Problem mit dem DHCP-Client hinzu, weil in jedem Knoten zwei Netzwerkkarten in Betrieb waren. Der Client konnt aber via DHCP nur ein Interface zuverlässig bedienen.

2.2.3 Clustersoftware

Um nicht ständig für jede Applikation, die am Cluster ausgeführt werden soll, die Interprozeßkommunikation zwischen den Knoten implementieren zu müssen, gab es verschiedene Möglichkeiten, eine einheitliche Schnittstelle zu finden.

Parallel Virtual Machine (PVM) Der PVM-Technologie liegt eine virtuelle Maschine zugrunde, die sich über mehrere auch heterogene Computersysteme erstreckt. Dabei übernimmt PVM nach der Initialisierung des Systems - für den Programmierer transparent - das Verteilen von Nachrichten über das Netz, die Datenkonversion innerhalb inkompatibler Rechner- und Betriebssystemarchitekturen (z.B. "endian conversion"), aber

auch die Aufgabenverteilung innerhalb der virtuellen Maschine.
Die PVM-Bibliothek bietet ein einheitliches System, um unterschiedlichste Programmstrukturen parallelisiert zu implementieren. Ein Benutzer kann zum Beispiel sein Programm als Ansammlung von kooperierenden Prozessen, sogenannte Jobs, programmieren, die über Interface-Funktionen auf gemeinsame virtuelle Ressourcen, wie Speicherbereiche etc., zugreifen. Durch diese Funktionen können neue Jobs über das Netz gestartet, beendet und untereinander synchronisiert werden, wobei typisierte Speicherstrukturen ausgetauscht werden. Sogar generelle Kommunikationsprinzipien, wie "Broadcasting", das Schicken einer Nachricht an alle Jobs, werden intern unterstützt (vergleiche [14], Kapitel Overview).
Die Kontrolle und Überwachung der einzelnen Prozesse können auf spezielle Kontrolljobs verteilt werden, wobei diese zu jedem Zeitpunkt der gemeinsamen Parallelberechnung den Ablauf der anderen Prozesse abbrechen oder auf andere Knoten verlagern können. Damit sind zum Beispiel Auswertungsprobleme mit dreiwertiger Logik dahingehend möglich, daß eine "Wenn"-Abfrage mit mehreren Kriterien getrennt und damit parallel ausgewertet werden kann. Das Ergebnis derartiger Auswertungen kann "wahr", "falsch" oder "nicht terminierend" sein.

Zusammenfassend kann jeder Prozeß mit jedem anderen Prozeß über eine definierte Programmier-Schnittstelle, deren Methoden einfach aber vollständig sind, in einem heterogenen Netzwerk direkt kommunizieren. Dies schlägt sich besonders in der weiten Verbreitung der PVM-Technologie im wissenschaftlichen Bereich und der existierenden Applikationen nieder. Sogar graphische Überwachungstools, z.B. xpvm, mit dem die Kommunikation laufend verfolgt und überwacht werden kann, und Debug-Werkzeuge stehen frei zur Verfügung. Diese Technologie ist mittlerweile veraltet und wird von Implementierung des Message-Passing-Interface-Standards abgelöst.

Message Passing Interface (MPI) Message Passing ist ein Programmierschema ("Design Pattern"), das bei parallelisierten Programmen zur Synchronisierung von Programmteilen durch Nachrichtenaustauch zum Einsatz kommt. Da Firmen mit der Zeit unter-

schiedliche Interpretationen und Implementierung dieses Prinzips prägten, wurde 1992 die Definition und Spezifikation [9] einer einfachen Bibliothek begonnen, die die besten Eigenschaften damals bestehender Projekte auf dem Message-Passing-Bereich in sich vereinen sollte. Ziel dieses Standards war nicht die Schaffung einer betriebssystemartigen Umgebung wie bei PVM, sondern dezidiert die Definition einer praktischen, portierbaren, effizienten und flexiblen Bibliothek zum Schreiben von parallelen Anwendungen. Es wurde dabei vorausgesetzt, daß MPI die Leistung eines Systems mit möglichst wenig Overhead nutzen kann, Modularität innerhalb von Prozessen gegeben (keine Referenzen auf externe Bereiche etc.) und die Möglichkeit zukünftiger Erweiterung durch objektorientierte Kapselung sichergestellt ist.
Dieser Standard wurde von mehreren Gruppen [36] [26] und Firmen implementiert und teilweise zwecks Leistungsoptimierung sogar in Hardware umgesetzt. Durch die klare und einfache Definition der Methoden und deren Funktionalität war dies mit wenig Aufwand möglich. Auf dem TU Graz-Cluster wurde die Software-Implementierung MPICH (siehe [15] , [36], [26]) installiert, da sie zum Zeitpunkt der Installation als einzige den gesamten Standard der MPI-Spezifikation Version 1.1 und Teile der Version 2.0 erfüllte.

2.2.4 Queueingsysteme

Linux ist ein Multiusersystem und stellt am Cluster mehreren Benutzer gleichzeitig die Rechenressourcen zur Verfügung. Geschieht dies aber auf denselben Knoten, so stören sie sich gegenseitig bei der Berechnung, da der Prozessor laufend zwischen den verschiedenen Prozessen hin- und herwechseln muß. Auch steht je Knoten nur eine beschränkte Netzwerkkapazität zur Verfügung, was zur Folge hat, daß, wenn mehrere kommunikationsintensive Prozesse auf demselben Rechner laufen, diese sich gegenseitig die Bandbreite streitig machen. Die unmittelbare Auswirkung auf die konkurrierenden Prozesse ist, daß alle Berechnungen verzögert werden und keiner die Ressourcen wirklich nutzen kann. Um trotzdem eine optimale Nutzung des Systems zu erzielen, muß eine Serialisierung der Jobs durchgeführt werden. D.h. die zu startenden Aufgaben aller Benutzer wer-

den mit den Angaben zu den benötigten Ressourcen an ein zentrales Verwaltungssystem geschickt und dort in Warteschlangen gestellt (Queueingsystem). Werden genügend Ressourcen am System frei, um einen Job in der Warteschlange abarbeiten zu können, so wird dieser gestartet und hat damit exklusiv die angeforderte Kapazität an Rechen- und Netzwerk-Leistung zur Verfügung. Für die Vergabe der Prioritäten im System sind bei allen bekannten Queueingsystemen unterschiedliche Verfahren implementiert, wobei diese je nach Warteschlange variiert werden können.
Der Berechnungsschlüssel setzt sich aus dem Produkt der benötigten Knoten und der Anzahl der erlaubten Jobs zusammen. Will ein Benutzer viele Jobs am System absetzen, so darf er nur wenige Knoten damit beschäftigen. Will er aber viele Knoten für eine Berechnung einsetzen, so darf er nur wenige Jobs absetzen. Dieses Gleichgewicht sorgt damit für eine gerechte Verteilung der Clusterkapazitäten.
Die Auswahl des Queueingsystems erfolgte aufgrund der freien Verfügbarkeit, der Verbreitung im akademischen Bereich und vor allem aufgrund der Fähigkleiten, die ein Queueingsystem zur Verfügung stellt. Die Entscheidung fiel dabei auf die im Bioinformatikbereich sehr weit verbreitete Software OpenPBS [37] und fundiert auf einer Studie der amerikanischen Behörde NASA [22] . Diese Studie vergleicht die gängigsten Vertreter der Queueingssysteme, wie z.B. OpenPBS, Codine, DQS, Condor, LSF etc. und spricht dabei allen Systemen Unzulänglichkeiten zu. Sie bestätigt aber, daß OpenPBS am besten für den Clusterbetrieb geeignet ist.

2.2.5 Services

Neben den bereits genannten Services, die für die prinzipielle Funktionalität des Clusters zur Verfügung stehen müssen, werden noch zusätzliche Dienste bzw. Applikationen am Master zur Verfügung gestellt:

- CVS ("Concurrent Versioncontrol System") [11] ist ein Serverdienst, der die Verwaltung und die Integration von Programmcode übernimmt, wenn mehrere Personen gleichzeitig am selben Projekt arbeiten. Dieses Systems hat aber den großen Nach-

teil, daß es nicht möglich ist, die Authentifizierung via LDAP zu implementieren. Aus diesem Grund mußten Skripts zur Verwaltung der CVS-Accounts angefertigt werden.

- APACHE, PHP, LDAP: Um am Cluster diverse Webfrontends anbieten zu können, wurde der am weitesten verbreitete Opensource Webserver *Apache* installiert. Dieser kann sowohl die Scriptsprache *PHP* interpretieren, als auch LDAP-Erweiterungen benützen, um mit der Kombination aus beiden Technologien ein Webportal mit lokalem Usermanagement implementieren zu können. Auch hier bleiben Username und Passwort dieselben wie im Cluster-System.

- PHP-CControl: Besonderes Augenmerk wurde auf ein übersichtliches Monitoringtool gelegt, mit dem der Gesamtstatus des Clusters und dessen Knoten laufend kontrolliert werden kann. Dieses Programm sollte einfach, übersichtlich und plattformunabhängig sein, um keinen Benutzer von dieser Möglichkeit der Job-Überwachung auszuschließen. Nach intensiven Internet-Recherchen erfüllte keines der bestehenden Monitoring-Programme diese Anforderungen. Einige von ihnen verbrauchten so viel Systemressourcen, daß der normale Clusterbetrieb beeinträchtigt wurde.
  Aus diesem Grund wurde eine eigene ClientServer-Architektur entwickelt, die es ermöglicht, auf Anfrage, den aktuellen Status eines jeden Knotens zu ermitteln. Dafür wurde in der Programmiersprache C ein flexibler Server entwickelt, der kontinuierlich auf jedem Knoten Systemdaten sammelt. Weiters implementiert dieser Server ein Textprotokoll, über das man die gesammelten Daten mittels bestimmter Schlüsselwörter abfragen kann. In der ersten Phase ist es damit möglich, CPU-, RAM- und Festplattenauslastung sowie Uptime über diesen Server abzufragen. Mit geringem Aufwand können zusätzliche, für den Clusterbetrieb interessante Daten zur Verfügung gestellt werden.
  Clientseitig ist aufgrund des verwendeten Textprotokolls keine Einschränkung bezüglich der Plattform gegeben. Allerdings muß der Client Zugriff auf das lokale Clusternetz besitzen. Deshalb wurde am Master ein Webinterface programmiert, das allen in der

LDAP-Userverwaltung existierenden Benutzern ermöglicht, Statusabfragen durchzuführen. Es wurde ein aus PHP-Klassen bestehendes Framework entwickelt, das durch einfaches Verbinden zu den auf allen Knoten laufenden Monitoringservern den aktuellen Status jedes einzelnen Knoten abruft. Aus diesen Rohdaten werden dynamisch Anzeigebalken generiert, welche die Zahlen in ansehnlicher und übersichtlicher Form darstellen. Es mußte auch auf eine geeignete Textversion Rücksicht genommen werden, da die Cluster-Übersicht auch für Sehbehinderte zugänglich sein mußte.
Dieses Monitoringtool mit dem Namen *PHP-CControl* wurde noch um eine Zusatzfunktionalität erweitert, mit der über das Webinterface Jobs in das OpenPBS-Queueing-System hineingestellt werden können. Das Framework verwaltet für jeden Benutzer getrennt sämtliche Ein- und Ausgabedateien der laufenden Jobs in einer eigenen Verzeichnisstruktur, auf die ebenfalls über das Webinterface zugegriffen werden kann. Damit ist es möglich, die Ergebnisse der Berechnungen über jeden beliebigen Browser abzufragen.
Der einzige erkennbare Nachteil dieses Systems ist, daß die Applikationen, die am Cluster via Webinterface zur Verfügung gestellt werden sollen, als Module integriert werden müssen. Ein dezidiertes Entwickeln dieser Applikationsmodule ist daher nicht zu vermeiden.

- SRS: Einfache Gendatenbankzugriffe können ebenfalls via Webinterface durchgeführt werden, da am Master zusätzlich noch das SRS-Modul der LIONBioscience AG installiert ist. Dabei wird über die Textdateien der Gendatenbanken ein Index gelegt, der schnelles Suchen in den Datenbanken ermöglicht und Verknüpfungen innerhalb der Datenbanken herstellt. Die Erstellung dieses Suchindizes ist bei den riesigen Datenmengen, z.B. der Genbank etc., mit einer einzigen Maschine sehr zeitaufwändig und kann mittlerweile ebenfalls auf den Cluster ausgelagert und verteilt berechnet werden. Die Zeitersparnisse liegen im Bereich von Tagen, je nachdem wieviele Datenbanken neu indiziert werden müssen.

Abbildung 2.7: PHP-CControl

2.2.6 Messungen

Um den Cluster international vergleichen [34] zu können, wurden Benchmark-Programme verwendet, die als Standard für Performancemessungen in Clustersystemen gelten. Zum einen wurde der standardisierte Linpack-Test [29] für parallele Architekturen verwendet, der ein zufälliges lineares Gleichungssystem mit doppelter Genauigkeit (64bits) auf Clustersystemen berechnet. Mit den daraus resultierenden 14 Gigaflops liegt der Cluster nur knapp unter der Wertungsgrenze der Top500-Rechner in der Welt.

Zum anderen wurden Zeitmessungen für diverse Applikationen wie z.B. das Rendern eines bestimmten 3D-Modelles mit einer parallelisierten Version der Grafik-Software PovRay [5] [8] durchgeführt und mit Ergebnissen anderer Clusterprojekte verglichen [16]. Bei diesen Tests liegen die Leistungen des Clusters unter den ersten 10 registrierten Bewerbern.

Kapitel 3

Diskussion

Aus dieser Diplomarbeit resultiert ein voll funktionstüchtiger Beowulf Cluster, der aus 24 Knoten mit insgesamt 48 Prozessoren und einem Master mit weiteren 2 Prozessoren besteht. Die Knoten untereinander sind mit zwei 100Mbps FastEthernet-Karten verbunden und greifen damit auf das 240GB große Plattensystem des zentralen Masters zu, der mit Gigabit-Ethernet integriert ist.
Die Installation der Knoten erfolgt dynamisch über das Netz, wobei innerhalb von wenigen Minuten ein neuer Knoten in den Rechnerverband integriert werden kann. Es ist nur eine Netzwerkverbindung und ein Diskettenlaufwerk notwendig, um eine Knoteninstallation der Distribution RedHat 7.1 durchzuführen. Die zentrale LDAP-Userverwaltung ermöglicht jedem Benutzer Zugriff auf alle Knoten des Clusters, um parallelisierte Applikationen berechnen zu lassen. Die gängigsten Kommunikationsbibliotheken wurden in das System über diverse Anpassungen und Umgebungsvariablen integriert. Bei Bedarf wurden Skripts, wie z.B. zur Einrichtung der virtuellen Maschinen, zusätzlich implementiert.
Zur permanenten und übersichtlichen Überwachung der Knoten wurde ein Webinterface programmiert, das den aktuellen Zustand der Knoten in ansehnlicher Form darstellt. Weiters stellt dieses Webinterface ein modulares Framework für Applikationen aus der Bioinformatik zur Verfügung, welche das Absetzen von Jobs über das Ressourcemanagement OpenPBS ermöglicht.

Um einen einfachen Zugang zu den zentral gespeicherten Datenbanken zu ermöglichen, wurde das Software-Paket SRS installiert, das nach einer Indizierungsphase schnelles Suchen in den Datenbanken zuläßt. Die riesige Menge an Daten, deren Ausmaß sich im Bereich von 200 bis 300GB bewegt, werden von einem zentralen NetApp-Filer mit einer Kapazität von einem Terabyte gehalten, der bei dem exponentiellen Anstieg der Datenbankgrößen für zukünftige Erweiterungen gerüstet ist.

3.1 Vor- und Nachteile des Beowulf Clusters

Es hat sich herausgestellt, daß ein Linux-Cluster eine preisgünstige Alternative zu den heutigen Großrechnern ist, da ausschließlich Standard-PC-Komponenten verwendet werden. Er ist auf einfache Weise erweiterbar, da zusätzliche Knoten mit wenig Aufwand in das System integriert werden können, und Applikationen mit nicht exzessiven Datenverkehr gut skalieren. Die einfache Bildung von Subclustern kann ebenfalls als Vorteil gewertet werden, um für kurzfristige Projekte dezidiert Rechenleistung zur Verfügung stellen zu können.

Durch die Verwendung von Linux als Betriebssystem entstand eine Plattform, die besonders im Bereich Bioinformatik weite Verbreitung findet, und daher existierende Software bereits parallelisiert einsatzbereit ist. Dafür werden Kommunikationsbibliotheken, wie z.B. MPI, verwendet, wobei durch die Verwendung der standardisierten Methoden plattformunabhängige und portierbare Versionen von Programmen zur Verfügung stehen.

Zusätzlich stehen durch die Verwendung dieses frei verfügbaren Betriebssystems [13] ein großer Pool an Software zur Verfügung, der das Arbeiten in einer Unixumgebung erleichtert. Damit bietet Linux versierten Unixbenutzern die gewohnte Umgebung und GUI(Graphical User Interface)-Benutzern brauchbare Anwendungen. Die Stabilitätsanforderungen an das Betriebssystem werden weitgehend erfüllt, um nicht durch das Abstürzen einzelner Rechenknoten Ergebnisse von Langzeitberechnungen zu verlieren.

Weiters steht für Linux ein frei zugänglicher und fast unerschöpflicher Pool an Informationen im Internet zur Verfügung, da die meisten Probleme bereits vorher gelöst und

dokumentiert wurden [4].

Als Nachteil kann der hohe Arbeitsaufwand gewertet werden, wobei Erfahrung und KnowHow sowohl im Bereich von Hardware als auch von Linux-Software Voraussetzung für das Gelingen eines derartigen Projektes ist. Wird die tatsächlich investierte Arbeitszeit in die Kosten einkalkuliert, ist es zum Teil fraglich, ob die Rentabilität des Selbstbaues gegeben ist. Im Falle des TU Graz-Clusters waren die Personalkosten jedoch vergleichsweise gering, da der Aufbau Teil dieser Diplomarbeit war. Allerdings überwiegt bei einem Selbstbau der hohe Grad an Flexibilität und Freiheit in der Wahl der Komponenten gegenüber der Abhängigkeit, wenn ein deutlich teureres "Out-Of-The-Box-System" angekauft wird. Die Preisunterschiede bewegen sich bei gleicher Leistung teilweise im Bereich des Doppelten. Darüberhinaus konnten besonders durch den Selbstbau wertvolle Erfahrungen gesammelt werden, die für den Aufbau und die Erweiterung zukünftiger Projekte auf diesem Gebiet von großem Nutzen sein werden. Im Vergleich dazu wäre es beim Ankauf eines vorgefertigten Systems nicht möglich gewesen KnowHow und Praxis im Umgang mit der Clusterhard- und software zu erlangen.

Wagt man nun den Vergleich mit Großrechnern, wie sie von der Firma Compaq^{TM}/HP^{TM}oder Sun^{TM}hergestellt werden, so kommt es hauptsächlich auf die Implementierung der Applikation und deren Skalierbarkeit an. Läßt sich eine Aufgabe in unabhängige Subprozesse aufspalten, die auf einem Cluster verteilt mit wenig Kommunikation auskommen, stellt der Cluster eine effiziente und günstige Lösung dar. Wird bei der Berechnung der Aufgabe sehr viel auf gemeinsame Daten zugegriffen, die möglichst im selben Arbeitsspeicher liegen sollten, so sind die Großrechner einem Cluster überlegen. Zudem hängt der Ankauf eines Großrechners hauptsächlich von der Finanzierbarkeit ab, denn die Preise für diesen bewegen sich im Bereich des vier- bis zehnfachen Anschaffungspreises eines Linux-Clusters. Hier wurden bewußt keine absoluten Zahlen verwendet, denn die Preise ändern sich ständig, wobei der Faktor dazwischen weitgehend konstant bleibt.

3.2 Erweiterungs- und Verbesserungsvorschläge

3.2.1 Netzwerk-Hardware

Die wichtigste Verbesserung des Clusters, die im nächsten Schritt durchgeführt werden sollte, ist der Ausbau der Netzwerkinfrastruktur in Richtung Gigabit-Technologien . Dabei kommt es nicht nur auf die beträchtliche Erhöhung der Bandbreite an, sondern viel mehr auf die Verringerung der Latenzzeiten, die unmittelbar in die erzielbare Rechenleistung einfließt. Diese Latenzzeiten sind jene Verzögerungszeiten bei der Interprozeßkommunikation, die den Austausch interner Statusinformationen zwischen den unterschiedlichen Knoten und die Synchronisation hemmt. Dabei spielt der TCP/IP-Stack, welcher die Verbindung zwischen Netzwerkkarte und Betriebssystemkern auf Internetprotokollebene implementiert, aber auch die verwendete Hardware ein zentrale Rolle. Kann man mit FastEthernet Latenzzeiten im Bereich von 100 bis 200 Mikrosekunden erzielen, so liegen Speziallösungen, wie es im High-Performance-Forschungsbereich einige Firmen, z.B. Myricom und IBM, anbieten, im Bereich zwischen 0.5 bis 20 Mikrosekunden.

GigabitEthernet: Mit dem Netzwerkausbau auf GigabitEthernet wird zwar die Bandbreite auf 1000Mbps erhöht, allerdings bleibt die Zugriffstechnologie auf das GigabitEthernet-Medium wie bei FastEthernet CSMA/CD (siehe Seite 18). Damit kann es wie bei FastEthernet Latenzzeiten von 100 Mikrosekunden nicht unterbieten. Diese Technologie stellt daher für die HighPerformance-Interprozeßkommunikation keine wesentliche Verbesserung dar, wohl aber beim Datenaustausch via NFS[20]. Hinsichtlich der Verkabelung können sowohl Kathegorie-5-UTP-Kabel, die bereits an Ihre Grenzen stoßen, als auch Glasfaserkabel verwendet werden, bei denen die Reichweite und Erweiterbarkeit bedeutend besser ist.

Myrinet Die erste "Hochgeschwindigkeits-Netzwerktechnologie" für Berechnungen im Forschungsbereich wird von der Firma Myricom [27] zur Verfügung gestellt und schafft bidirektional einen Durchsatz von bis zu 2Gbps. Die Verbindung zwischen den Myrinet-

Karten und den Switches kann mit ihrer proprietären Kupferverkabelung oder über Glasfaserverbindungen hergestellt werden. Diese Kupferkabel lassen mit Fullduplex-Übertragung nur eine Distanz von drei Metern zu, mit Halfduplex-Übertragung über 18 Meter. Glasfaserkabel hingegen haben je nach Art (multimode oder singlemode) eine Reichweite zwischen 100 Meter und 10 Kilometer. Bei einer Erweiterung des Clusters könnten Kupferkabel im Fullduplex-Betrieb je nach Aufbau zu kurz werden.
Topologisch betrachtet sind mit Myrinet Punkt-zu-Punkt-Verbindungen, Hub-basierte oder Switch-basierte Aufbauformen möglich, wobei die Anzahl der Switching- bzw. Hub-Einheiten zwischen zwei Rechnern nicht wie bei Ethernet einen limitierenden Faktor darstellt. Einzige Auswirkung ist die geringfügige Erhöhung der Latenzzeiten, die bei einer normalen Verbindung zwischen 5 und 18 Mikrosekunden schwankt. Außerdem ist mit diesen Netzwerkkarten DMA-Zugriff (Direct Memory Access) über das Netz möglich, wie es bisher nur für einen schnellen Transfer vom RAM zu internen Geräten verwendet wurde. Damit wird clusterweit ein verteiltes "shared memory system" erzeugt, auf das sämtliche Prozessoren direkt zugreifen können.
Diese Technologie ist derzeit bei Clustersystemen im HighEnd-Bereich weit verbreitet, unter anderem auch deshalb, weil eine eigene Implementierung des MPI-Standard-Interfaces optimiert auf dieses Netzwerk existiert. Damit kann jedes Programm, das mit MPI-Bibliotheken programmiert wurde, sämtliche Vorteile dieser Technologie ohne Zusatzaufwand nutzen.

Giganet Giganet ist die Antwort von Intel auf den Bedarf eines schnellen Netzwerks in der Clusterbranche und basiert auf der "Virtual Interface Architecture(VIA)"-Technologie. Dabei bildet diese VI-Architektur ein plattformneutrales Software- und Hardwaresystem, das direkt über ein eigenes Kommunikationsprotokoll - nicht IP - Daten zwischen den Knoten transferiert. Dieser SAN-Standard beseitigt den Kommunikationsoverhead, der bei IP hauptsächlich für große Netzwerke im WAN (Wide Are Network) vertretbar ist. Diese Technologie ist relativ neu und wird sich nur dann durchsetzen, wenn Hardwarehersteller eine bessere Integration in ihr Input/Ouput-System vornehmen werden. Erste Daten der

cLAN-Karten und -Switches sind vielversprechend und liegen während unidirektionaler Kommunikation bei 1Gbps Bandbreite und 7 Mikrosekunden Latenzzeit.

Scalable Coherent Interface (SCI) Als Alternative zur proprietären Lösung von Myricom kann der IEEE Standard SCI (IEEE Std 1596-1992) angesehen werden, der bei einer Übertragungsrate von 3,2Gbps in beide Richtungen Latenzzeiten unter 2,5 Mikrosekunden erzielt. Um diese Eigenschaften erfüllen zu können, kommen mehrere unidirektionale Punkt-zu-Punkt-Verbindungen zum Einsatz, die in einer Ring- bzw. Torustopologie organisiert werden und einen hohen Durchsatz garantieren. Je nach Anzahl der zu verbindenden Knoten (bis zu 64000) werden zwei- oder dreidimensionale Torusringe errichtet, die auf jeder Schnittebene und in jede Richtung einen eigenständigen Ring bilden. Damit skaliert diese Technologie bei einer Erweiterung sehr gut, da bei der Kommunikation stets der schnellste Pfad entlang der räumlichen Struktur zwischen zwei Knoten verwendet wird.
Mit der Hardwarespezifikation der Verbindungen werden bei SCI auch Paketprotokolle definiert, wobei eine geringe Paketgröße charakteristisch ist. Diese Pakete werden sowohl über parallele Kupferleitungen als auch über Glasfaserkabel transferiert. Hierbei benötigt jedes Interface zwei unidirektionale Verbindungen, die parallel angesprochen werden können.
Die Paketprotokolle selbst unterstützen "shared memory"-Zugriff, wobei die dafür nötigen Anfragen und Antworten in SCI-Anfragen und -Antworten gekapselt werden. Der korrekte Datentransfer wird über Handshakeprotokolle, wie sie bei LVD-SCSI-Festplatten verwendet werden, garantiert und durch diverse Cachingverfahren zusätzlich optimiert.
Für diese Technologie gibt es ähnlich wie bei Myrinet eine angepaßte Version der MPI-Implementierung, die spezielle Eigenschaften von SCI zur Steigerung der Kommunikationsleistung ausnützt.

Wertung Während der ersten Recherchen über Netzwerktechnologien war SCI eher ein Außenseiter und Myrinet die einzige sinnvolle Lösung im Bereich "HighPerformanceCom-

puting". Sowohl die Hardware- als auch die Softwareunterstützung waren bei Myrinet in einem ausgereifteren Zustand. Mittlerweile kann SCI als ernster Konkurrent für Myrinet betrachtet werden, da zahlreiche Projekte spezielle Softwareanpassungen für diesen nicht proprietären Standard aufgenommen haben. Die Tatsache, daß erstens nicht nur eine einzelne Firma hinter SCI steht, zweitens Spezifikationen dafür verfügbar sind und drittens eine Weiterentwicklung garantiert ist, läßt den Schluß zu, daß SCI auch für die Zukunft die sinnvollere Lösung sein wird. Myrinet und SCI liegen bei vergleichbarer Leistung in derselben Preisklasse, steigern aber die Kosten je Knoten beträchtlich.
Um die Diskussion über mögliche Netzwerktechnologien abzuschließen, muß noch erwähnt werden, daß bei all diesen Technologien, nicht das Netzwerk zum beschränkenden Faktor wird, sondern bereits der 32-Bit-PCI-Bus. Aus diesem Grund werden diese Netzwerkinterfaces hauptsächlich mit 64-Bit-PCI-BUS ausgestattet, was aber für den TU Graz-Cluster kein Problem ist. Sämtliche Knoten besitzen 2 zur Zeit unbenutzte 64-Bit-PCI-Schächte und sind damit für einen "Netzwerkupgrade" gerüstet.

3.2.2 Computing-Hardware

Eines der größten Vorteile eines Clusters ist, daß er auf einfache Weise durch zusätzliche Rechenknoten erweiterbar ist. Da es in nächster Zeit einige Projekte an der TU Graz geben wird, die auf dem Cluster ihre Berechnungen durchführen werden, muß die Anzahl der Knoten verdoppelt bzw. verdreifacht werden. Allerdings steigt die Leistung des Clusters nicht linear mit der Anzahl der Knoten. Zusätzliche Knoten bringen einen erhöhten Kommunikationsoverhead im Netzwerk mit sich und korrigieren damit die Leistungskurve nach unten.
Bleibt man bei 32-Bit-Rechenknoten so muß die Wahl zwischen AMDs Athlon und Intels P4 getroffen werden, wobei der Intel momentan eine höhere Prozessor-Taktrate erreichen kann. Eine detailliertere Empfehlung würde diesbezüglich, wie im Kapitel Methoden beschrieben, in ein Abwägen der Vor- und Nachteile spezieller Prozessoreigenschaften und dessen Umfeld enden.

Bei der Auswahl einer zukünftigen 64-Bit-Rechner-Architektur hingegen wurden die Möglichkeiten in den letzten Jahren beträchtlich eingeschränkt. Nachdem die Firma Compaq^{TM}die Alphatechnologie^{TM}an Intel^{TM}verkauft hat, Compaq die Fertigung von Alphaprozessoren in Kürze einstellen wird und selbst von HP^{TM}übernommen wurde, bleiben eigentlich nur noch HP, Intel und Sun als Alternativen. Allerdings hat Sun seinen Schwerpunkt bei der Prozessorentwicklung nicht auf Rechenleistung gesetzt, wodurch die SPARC-Prozessoren für einen Rechencluster nicht von Bedeutung sind. Daraus resultieren nur noch zwei Möglichkeiten:

- Alphatechnologie
- Intels 64-Bit-Itanium, dessen Entwicklung nur langsam voranschreitet und die Leistung der Alphaprozessoren nicht erreicht.

Auch hier ist daher eine pauschale Empfehlung zum jetzigen Zeitpunkt schwierig.

3.2.3 Software

Installationssystem Das Installationssystem könnte dahingehend verbessert werden, daß nur noch knotenspezifische Daten zentral in einer Datenbank abgespeichert werden. Mit Hilfe dieser Daten könnte bei Bedarf eine normale Komplettinstallation des laufend aktualisierten Systems durchgeführt werden. Dies hätte den Vorteil, daß nicht ein Abbild eines Knotens laufend gewartet und geclont, sondern eine normale Installation des Betriebssystems auf jedem Knoten einzeln durchgeführt werden müßte (siehe [3]).

Kompileroptimierung Beträchtliche Optimierungen könnten beim Übersetzen von Programmen in Maschinencode durchgeführt werden, wenn man nicht den freien Standard-GNU-Kompiler, sondern kommerzielle Konkurrenten verwenden würde. Je nach verwendeten Kompiler [30] könnte durch automatische Prozessoroptimierungen während der Übersetzungsphase die benötigte Laufzeit von Programmen um bis zu 50 Prozent reduziert werden. Allerdings sind teilweise die beträchtlichen Kosten für die Verwendung auf allen Knoten weit überzogen und können den Ankauf nicht rechtfertigen.

Betriebssystemoptimierung Die Optimierung im Bereich des Betriebssystems ist eine weitere Möglichkeit, eine höhere Leistung des Clustersystems zu erzielen. Für die Interprozeßkommunikation wird dabei nicht das TCP/IP-Internet-Protokoll verwendet, sondern ein eigenes, das mit gängiger EthernetHardware [19] implementiert wird. Damit fällt der für den WAN-Betrieb benötigte Kommunikationsoverhead weg und erlaubt über dieselbe Hardware eine schnellere Kommunikation. Diese Variante ist jedoch für Speziallösungen und nicht für eine allgemeine Recheninfrastruktur geeignet, die unterschiedlichen Anforderungen von Instituten in Graz entsprechen muß.

Clusterprojekte Bei der Clustersoftware wurde eine Auswahl der gängigsten Vertreter getroffen, deren Stabilität und Einsatzbereitschaft überzeugt hat. Im Laufe des Aufbaus wurden aber auch andere Clusterprojekte auf ihre Tauglichkeit hin überprüft.
Beispiele:

- LAM [36] stellt eine durchaus denkbare Alternative zur aktuell verwendeten MPI-Implementierung dar.
- MOSIX [2] sorgt für eine benutzertransparente Lastverteilung am Cluster.
- OpenMP [21] ist ein zukunftsträchtiges Projekt für einfache portierbare Programmierung im skalierbaren SMP-Bereich.

Allerdings konnten sie teilweise aufgrund ihres instabilen Zustandes nicht eingesetzt werden. Für die zukünftige Administration des Clusters ist es wichtig, diese Projekte laufend zu beobachten, um sie bei Bedarf heranziehen und integrieren zu können.
Für die Bioinformatik in Graz sind Java-Projekte auf diesem Bereich besonders interessant, da die bereits entwickelte Software ausschließlich in Java vorliegt. Diese Softwarepakete könnten zu einer parallelisierten Version erweitert werden, damit z.B. aufwendige Clusteranalysen größeren Ausmaßes auf den Cluster ausgelagert werden können. Im Zuge des Aufbaus einer zentralisierten Auswertungsinfrastruktur für Biologen in Graz sollten derartige Überlegungen angestrengt werden, um aufgrund der beschränkten Skalierbarkeit einem Engpaß bei der lokalen Rechenkapazität der dafür notwendigen Applikationsserver

vorzubeugen.
Die Varianten dafür gehen von Implementierung der MPI-Schnittstellen [7] in Java über eine Java-OpenMP-Integration [6] bis zum Kernansatz einer erweiterten virtuellen Maschine [10], bei der in der Applikation abgespaltene Threads transparent über die Knoten verteilt werden. Eine genauere Untersuchung über den Entwicklungsstand dieser Technologien und deren Skalierbarkeit würde aber den Rahmen dieser Diskussion sprengen.

SRS Nachdem die Grundinstallation von SRS bereits abgeschlossen ist, kommt bei der Erweiterung dieses Systems ein beträchtlicher Arbeitsaufwand für die laufende Wartung und Aktualisierung der Datenbanken und deren Index auf die Administration zu. Diesbezüglich gibt es bereits kommerzielle Produkte, z.B. SRS-Prisma^{TM}, die allerdings den finanziellen Aufwand nicht rechtfertigen. Deshalb sollte eine Automatisierung der Wartungs- und Download-Arbeit implementiert werden, die in Opensource-Manier auch anderen Projekten zur Verfügung gestellt werden könnte. Ein derartiges Projekt ist bereits am Institut in Planung und wird hoffentlich bald den gewünschten Erfolg bringen.

NetConsole Zum Schluß sollte noch auf eine für die Clusteradministration äußerst hilfreiche Erweiterung des Betriebssystemkerns hingewießen werden. Sie wurde erst im September 2001 ins Leben gerufen und nennt sich *NetConsole*. Sie beseitigt das Problem der Rekonstruktion eines Kernelabsturzes dahingehend, daß die letzten Zeichen, die ein noch von sich gibt, in ein Netzwerkpaket verpackt und zu einem zentralen Logging-Rechner geschickt werden. Bisher war es bei einem Absturz nicht möglich, diese Meldung, die normalerweise auf die lokale Konsole geschrieben wird, anzeigen zu lassen, da zu diesem Zeitpunkt der Rechner bereits in einem nicht mehr funktionstüchtigen Zustand war. Wenn diese Erweiterung ein einsatzfähiges Stadium erreicht, wird sie bei der Fehlersuche im Clusterbetrieb äußerst hilfreich sein.

3.3 Danksagung

Einen herzlichen Dank möchte ich an alle Beteiligten dieses Projektes aussprechen, insbesonders Z. Trajanoski, der die politische Vorarbeit und die Betreuung übernommen hat. Weiters möchte ich mich bei D. Rieder, M. Hanscho, G. Thallinger und A. Sturn für die tatkräftige Unterstützung in der Planungsphase bedanken, die mit mir die Idee des Clusters konkretisiert haben. Dank gebührt auch dem Zentralen Informatikdienst der TU Graz, in Vertretung von I. Kamrat, M. Stepponat und U. Chararas, die der Finanzierung und des Ankaufs Sorge trugen, sowie G. Prem und M. Lang, die konkret beim technischen Aufbau mitwirkten. Zuguterletzt möchte ich noch S. Platzer , M. Hanscho und G. Schwann für die Geduld und Mühen bei der Korrektur dieser Arbeit danken.

Literaturverzeichnis

[1] F. Abbas. *Part 3: Carrier sense multiple access with collission detection (CSMA/CD) access method an physical layer specifications.* The Institute of Electrical and Electronics Engineers, Inc., 2000 edition, March 2000.

[2] A. Barak. Mosix, scalable cluster computing for linux. WWW, Oktober 2001. http://www.mosix.org/.

[3] M. A. Bentley. Redhat kickstart. WWW, July 2000. http://www-users.cs.umn.edu/ bentlema/projects/kickstart/.

[4] A. Blücher, H. Weiß, et al. Quantenchemie auf einem linux-cluster: Der struktur von molekülen auf der spur. *Linux-Magazin*, 1, Jänner 1999. http://www.linux-magazin.de/ausgabe/1999/01/Quantenchemie/quantenchemie.html.

[5] R. Bono. Pondermatic iv, home supercomputing with linux. WWW, Dezember 1999. http://www.cris.com/ rjbono/html/pondermatic.html.

[6] M. Bull. Jomp home page. WWW, 2001. http://www.epcc.ed.ac.uk/research/jomp/.

[7] B. Carpenter. The hpjava project, mpijava home page. WWW, Mai 2000. http://aspen.csit.fsu.edu/pss/HPJava/mpiJava.html.

[8] A. Dilger, B. Kline, et al. Pvm patch for pov-ray. WWW, Jänner 2000. http://www.haveland.com/povbench/.

[9] J. Dongarra, D. Walker, et al. Mpi: A message-passing interface standard. WWW, Juni 1995. http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html.

[10] M. Factor. Cluster virtual machine for java. WWW, Oktober 2001. http://www.haifa.il.ibm.com/projects/systems/cjvm/index.html.

[11] K. Fogel. *Open Source Development With CVS*. The Coriolis Group, 2 edition, 2000. http://cvsbook.red-bean.com/cvsbook.html.

[12] OpenLDAP Foundation. A quick-start guide. WWW, May 2000. http://www.OpenLDAP.org/doc/admin/quickstart.html.

[13] Gnu general public license. WWW, Juni 1991. http://www.gnu.org/copyleft/gpl.html.

[14] A. Geist, A. Beguelin, et al. *PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing*, chapter Overview. MIT Press, Scientific and Engineering Computation, September 1994. http://www.netlib.org/pvm3/book/pvm-book.html.

[15] W. Gropp, E. Lusk, et al. A high-pxerformance, portable implementation of the mpi message passing interface standard. *Parallel Computing*, 22(6):"789–828", sep 1996. ftp://ftp.mcs.anl.gov/pub/mpi/mpicharticle.ps.

[16] A. Haveland-Robinson. Povbench - the official home page. WWW, Jänner 2001. http://www.haveland.com/povbench/.

[17] L. Howard. Pam ldap module. WWW, Oktober 2001. http://www.padl.com/pam_ldap.html.

[18] Isc dynamic host configuration protocol (dhcp). WWW, Oktober 2000. http://www.isc.org/products/DHCP/.

[19] Y. Ishikawa. Rwcp parallel distributed system software tsukuba laboratory. WWW, Juli 2001. http://pdswww.rwcp.or.jp/.

[20] R. Jain. *GigaBit Ethernet*. Ohio State University, 1 edition, March 1998. ftp://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-97/h_8gbe.pdf.

[21] Openmp simple, portable, scalable smp programming. WWW, Oktober 2001. http://www.openmp.org/.

[22] J. Kaplan, M. L. Nelson, et al. A comparison of queueing, cluster and distributed computing systems. WWW, Juni 1994. http://techreports.larc.nasa.gov/ltrs/94/tm109025.tex.refer.html.

[23] C. M. Kozierok. Ide/ata vs. scsi: Interface comparison. WWW, April 2001. http://www.pcguide.com/ref/hdd/if/comp.htm.

[24] Linux headquarters. WWW, Oktober 2001. http://www.linuxhq.com/.

[25] Peter Lukas. *Softwareparadigmen*, chapter 10, page 112 ff. Ordinariat für Softwaretechnologie, 2 edition, März 1999. ftp://ftp.ist.tugraz.at/pub/courses/old/swp00/vo/oldscriptum/ld_scriptum.ps.gz.

[26] Mpich-a portable implementation of mpi. WWW, August 2001. http://www-unix.mcs.anl.gov/mpi/mpich/.

[27] Myrinet. WWW, August 2001. http://www.myricom.com/myrinet.

[28] Fortran compilers. WWW, Oktober 2001. http://www.nag.com/nagware/NP.asp.

[29] A. Petitet, R. C. Whaley, et al. Hpl - a portable implementation of the high-performance linpack benchmark for distributed-memory computers. WWW, September 2000. http://www.netlib.org/benchmark/hpl/.

[30] Pgi cdk [tm] cluster development kit [tm] software for linux. WWW, Dezember 2000. http://www.pgroup.com/prodcdk.htm.

[31] Inc. Red Hat. How to download red hat linux. WWW, Mai 2001. http://www.redhat.com/download/howto_download.html.

[32] B. Saphir, P. Bozeman, et al. Production linux cluster, sc 99 tutorial. WWW, November 1999. http://www.nersc.gov/research/FTG/tribble/production_linux_clusters_v1.pdf.

[33] C. Shirky. Whatis?com: Beowulf. WWW, Juli 2001. http://whatis.techtarget.com/definition/0,,sid9_gci211652,00.html.

[34] The linpack benchmark. WWW, September 2000. http://www.top500.org/lists/linpack.html.

[35] L. Torvalds et al. The linux kernel archives. WWW, Oktober 2001. http://www.kernel.org.

[36] LAM Team / UND. Lam / mpi parallel computing. WWW, Oktober 2001. http://www.lam-mpi.org/.

[37] Portable batch system. WWW, August 2000. http://www.openpbs.com.

[38] S. Vidal, T. Myklebust, et al. Nfs links, trond's nfsv3 client patches. WWW, October 2001. http://nfs.sourceforge.net/.

[39] www.bs welt.de, editor. *Einführung in das Betriebssystem Linux*, page 8. www.bs-welt.de, 2000. www.bs-welt.de.

[40] Wie ist ein kernel aufgebaut? WWW, May 2000. http://www.bs-welt.de/linhtml/kernelaufbau.html.

[41] K. Yap. Etherboot. WWW, May 2001. http://etherboot.sourceforge.net/.

Index

Anhang A

Parallele Architekturen

A.1 Allgemeine Überlegungen

Die Parallellisierung von Arbeitsabläufen dient der Reduktion der Dauer von Prozessen, sowie zur Einsparung und Entlastung kritischer Ressourcen. In der Computerbranche werden zum einen parallele Architekturen dazu verwendet, beschränkte Ressourcen mehreren Benutzern gleichzeitg und damit parallel zur Verfügung zu stellen. Zum anderen besteht das starke Bestreben, immer schnellere und bessere Computer herzustellen, deren Leistung durch einfaches Dazuhängen neuer Geräte möglichst linear zu steigern und um dadurch eine optimale Skalierung von Aufgabenstellungen zu erreichen. Ein weiterer Grund ist die Vereinfachung komplexer Vorgänge durch Aufteilung in kleinere, überschaubarere Unteraufgaben, die dann möglichst parallel abgearbeitet werden können.
Die Anwendung paralleler Architekturen birgt aber auch Nachteile in sich, wie zum Beispiel die zeitliche Abhängigkeit und Notwendigkeit der Synchronisation einzelner Prozesse untereinander. Wenn nur ein Prozeß eine Aufgabe übernimmt, so hat er stets den inneren Zustand des gesamten Systems in sich integriert. Bei parallelen Prozessen müssen untereinander Statusinformationen ausgetauscht werden, um ihren lokalen Fortschritt und dessen Ergebnisse in der Abarbeitung der Unteraufgaben anderen Prozessen mitzuteilen. Damit ist also ein gewisser Kommunikationsaufwand und -verzögerung verbunden, die die

Rechendauer nicht überschreiten darf.
Aufgrund der Parallelität läuft jeder Prozeß auf unterschiedlichen Rechnern und mit seiner lokalen Geschwindigkeit. Damit entstehen zwischen den Prozessen relative Geschwindigkeiten, die beim Programmieren und der Fehlersuche schwer nachvollziehbar sind. Man erreicht zeitliche und statusbedingte Abhängigkeiten zwischen den Prozessen, die ein reproduzierbares Testen nur schwer und mit hohem Aufwand möglich macht. (Vergleiche [14] und [25])

A.2 Möglichkeiten der Parallelisierung

Die Parallelisierung von Aufgaben kann auf unterschiedliche Arten durchgeführt werden, wobei nicht jede Methode den gezielten Erfolg im Vergleich zum benötigten Aufwand erbringt.

Abstrakte virtuelle Maschine Hierbei wird über mehrere Rechner verteilt eine virtuelle Von-Neumann-Maschine gelegt, die konkurrierende Programme parallel im Verbund abarbeiten kann. Dabei stellt sie die aktuellen Ressourcen zur Verfügung und funktioniert wie ein großer Rechner mit einem gemeinsamen Speicher und mit mehreren Prozessoren. Damit kann theoretisch auf mehreren kleineren Rechnern, deren Anschaffungskosten meist bedeutend geringer sind als die eines Großrechners, vergleichbare umfangreiche Aufgaben gelöst werden. Das hat den Nachteil, daß, falls beim bekannten Vertreter PVM [14] ein Rechenknoten ausfällt, die gesamte Berechnung abbricht, da die virtuelle Maschine nicht mehr vollständig ist. Die Fehleranfälligkeit ist aufgrund der hohen Anzahl beteiligter Komponenten größer und hat den Verlust der gesamten Berechnung zur Folge.

Swarming Anders ist das bei den sogenannten "Cluster of Working Nodes"(CLOWNS), die auf dem Prinzip des Swarmings basieren. Dabei werden nicht parallelisierte Programme mit unterschiedlichen Eingabedaten gespeist, die dann unabhängig voneinander auf jedem Knoten getrennt berechnet werden und die gewünschten Ergebnisse liefern. Hierbei

wird die Parallelisierung nicht auf programmiertechnischer Basis durchgeführt, sondern aufgrund der Aufteilung der Aufgabenstellung bzw. der Eingabedaten. Diese Methode hat den Vorteil, daß jede Applikation auf dem Cluster ohne Zusatzaufwand gerechnet werden kann. Es gibt zum Teil sogar Mischformen, die mit Hilfe anderer Parallelisierungsverfahren, wie z.B. des Message-Passings, das "Swarming" überwachen.

Message-Passing Das Prinzip des Message-Passing beschreibt die Initialisierung, Synchronisierung und Terminierung von Prozessen mittels Nachrichten, die zwischen den Knoten über definierte Schnittstellen ausgetauscht werden. Diese Form von Parallelisierung ist bei der wissenschaftlichen Applikationsprogrammierung weit verbreitet. Näheres dazu kann im Kapitel Message-Passing auf Seite 35 nachgelesen werden.

Repository Architectures Als interessante Alternative zur prozeßorienterten Parallelisierung kann das Prinzip der "Blackboard- oder Repository-Architectures" angesehen werden. Dieser Aufbau beruht auf einen zentralen Datenpool, auf den mehrere unabhängige Prozesse gleichzeitig zugreifen können. Ein Vertreter dieses Systems ist das sogenannte LINDA-Prinzip, wobei dort Tupel von Daten und dazugehörigen Markierungen (TAGS) in einen zentralen Pool gegeben werden. Diese Daten können von komplett unabhängigen Prozessen aufgearbeitet werden und die Ergebnisse wieder mit dem entsprechenden Tags in den Pool zurückgeschrieben werden. Ein Masterprozeß überwacht die gesamte Berechnung und terminiert nach erfolgreichem Abschluß die gestarteten Prozesse.
Mit dieser Methode wird ein äußerst flexibles System von Komponenten ermöglicht, die entkoppelt arbeiten und ausschließlich über definierte Funktionen auf den Datenpool kommunizieren. Dieses Verfahren hat allerdings den Nachteil, daß das Suchen im Datenpool aufwendig ist, besonders bei großen Systemen.

Anhang B

Internet-Links

Allgemein:

- Allgemeine Sammlung von Informationen, Links Bücher:
  http://www.csse.monash.edu.au/ rajkumar/cluster/
- Allgemeine Sammlung von ClusterLinks:
  http://www.cbc.umn.edu/ mwd/linux-info.html

Aufbau:

- Cookbook for LinuxCluster:
  http://www.scl.ameslab.gov/Projects/ClusterCookbook
- How to build a Linux Cluster: http://ha.redhat.com/whitepapers/clus-design
- Linux-Cluster Howto: http://www.ram.org/computing/linux/linux_cluster.html

Parallelprogrammierung:

- http://mombasa.anthro.utah.edu/wooding/Linux/Parallel.html
- PVM Basics:
  http://athene.riv.csu.edu.au/ ialtas/module3/

- PVM Buch:
  http://www.netlib.org/pvm3/book/pvm-book.html
- MPI:
  http://WWW.ERC.MsState.Edu/labs/hpcl/projects/mpi/

Hardware:

- 32/64-bit Systeme: http://www.bernd-leitenberger.de/neubeginn-bei-64-bit.htm
- Intel: http://www.intel.com/mobile/processors
- AMD: http://www.amd.com:7246/us-en/Processors/TechnicalResources
- RAM: http://www.hardtecs4u.com/reviews/2001/ddr-sdram/
- RAM: http://www.pcguide.com/ref/ram/
- Festplattensysteme: http://www.pcguide.com/ref/hdd/if/index.htm
- Netzwerk: http://www.wildpackets.com/compendium
- FastEthernet: http://www.wildpackets.com/compendium/FE
- Netzwerk: http://standards.ieee.org/getieee802/

Software:

- Linux Basics: http://www.bs-welt.de/betriebssysteme.htm
- RedHat Installationsanleitung:
  http://www.europe.redhat.com/documentation/rhl7.1/rhl-ig-x86-de-7.1/
- PVM: http://www.netlib.org/pvm3
- MPI-LAM: http://www.lam-mpi.org/
- MPI-MPICH: http://www-unix.mcs.anl.gov/mpi/mpich/