Power Grid Exist:

Ziele, Architektur und Aufbau

G. Bengel, S. Senst, M. Szkutnik, M. Schmitt,
S. Abu-Rab, D. Defiebre, L. Tauer, B. Alkuns, S. Strack

Informatik-Berichte
Hochschule Mannheim – Fakultät für Informatik

Computer Science Reports
Mannheim University of Applied Sciences – Computer Science Department

CSR 001.10
März 2010

URL: http://www.informatik.hs-mannheim.de/reports

hochschule mannheim

FAKULTÄT FÜR INFORMATIK

Power Grid Exist:

Ziele, Architektur und Aufbau

G. Bengel, S. Senst, M. Szkutnik, M. Schmitt, S. Abu-Rab, D. Defiebre, L. Tauer, B. Alkuns, S. Strack

Stand: Februar 2010

Informatik - Berichte Hochschule Mannheim – Fakultät für Informatik

Hochschule Mannheim
Fakultät für Informatik

Paul-Wittsack-Str. 10
D-68163 Mannheim

Tel.: +49 (621) 292 6223
Fax.: +49 (621) 292 6237

http://www.bts.hs-mannheim.de
http://www.vts.hs-mannheim.de
http://www.pvs.hs-mannheim.de

email:

g.bengel@hs-mannheim.de,
s.senst@hs-mannheim.de,
manuel@szkutnik.de,
kaldaron@gmx.de,
sufian.abu-rab@freenet.de,
daniel.defiebre@freenet,
lemmytower@gmx.net,
baris.alkun@googlemail.com
sylvia.strack@gmx.de

Inhaltsverzeichnis

Abstract

Um den Anforderungen an ein praxisnahes Hochschulstudium gerecht zu werden und den Ablauf von Forschungsprojekten zu vereinfachen, haben wir, basierend auf der Idee des parallelen Rechnens, ein Cluster an der Hochschule Mannheim eingerichtet.

Inhalt dieses Dokuments

In diesem Dokument gehen wir auf die technischen Eigenschaften dieses Clusters ein und darauf, welche Software installiert ist oder noch installiert werden soll. Außerdem stellen wir hier Forschungsprojekte vor, die, in Kooperation mit deren Verantwortlichen, das Cluster nutzen werden.

Weitergehende Informationen

Die Grundlagen für das in diesem Dokument beschriebene Projekt können in „Grundkurs Verteilte Systeme“ [B 04] und „Masterkurs Parallele und Verteilte Systeme“ [BBKS 08] nachgelesen werden.

Hierzu existiert jeweils eine Webseite, auf der weitere Informationen über die jeweiligen Bücher angeboten werden:

Grundkurs Verteilte Systeme: http://www.vts.hs-mannheim.de

Masterkurs Parallele und Verteilte Systeme: http://www.pvs.hs-mannheim.de

Unter anderem werden hier die Autoren vorgestellt und zu jedem Buch der Inhalt vorgestellt. Außerdem können hier Beispielprogramme zu den Beispielen in den Büchern herunter geladen werden.

1. Power Grid Exist

Die folgenden Unterabschnitte gehen auf den Zweck und die Ziele von Power Grid Exist und dessen Bedeutung in der Ausbildung und der Forschung an der Hochschule Mannheim ein.

1.1 Ziele von Power Grid Exist

Power Grid Exist (PGE) ist ein aus Studiengebühren finanziertes Cluster der Fakultät für Informatik an der Hochschule Mannheim. Das Cluster wurde im Wintersemester 08/09 beschafft und im Sommersemester 2009 aufgebaut und in Betrieb genommen.

1.2 Ausbildungsziele Power Grid Exist

PGE ermöglicht eine praxisnahe und gezielte Ausbildung der Studenten im Bereich des parallelen Rechnens. Der Einsatz von parallelen Techniken wie MPI und OpenMP kann direkt am Cluster umgesetzt werden. Dies ermöglicht die Durchführung von Übungen im Rahmen der folgenden Vorlesungen

- Cluster-, Grid- und Cloud-Computing, 6/7 Semester, Bachelor Informatik,

- Parallele Prozesse, 1/2 Semester, Master Informatik,

- Verteilte eingebettete und mobile Anwendungen, 6/7 Semester, Bachelor Informatik (vorläufig vorgesehen, muss noch evaluiert werden).

Die Studierenden erhalten durch die praxisorientierten Erfahrungen ein umfangreiches Fachwissen über parallele Metriken und somit ein umfassendes Wissen über den Einsatz paralleler Architekturen und deren Vor- und Nachteile. Damit einhergehend ist die Bewertung der Performance im Bezug auf den Vergleich von Kommunikationszeiten mit Rechenzeiten.

1.3 E-Science

PGE dient zur Unterstützung des E-Science an der Hochschule Mannheim und zur Ausnutzung der Rechenleistung des PGE.

Bisher sind Anwendungen unterschiedlicher Bereiche in Kooperation mit verschiedenen Fakultäten geplant:

Multiphysics

Zum Einsatz im Lehrbetrieb kommt Multiphysics, ein Programm zur Simulation physikalischer Vorgänge, durch Herrn Prof. Dr. Rasenat von der Fakultät für Informatik.

OpenFOAM

OpenFOAM (Open Field Operation and Manipulation) ist ein numerisches Simulationsoftwarepaket für kontiniumsmechanische Probleme. OpenFOAM ist in C++ geschrieben und frei. OpenFOAM wird von Herrn Prof. Dr. Kniffler von der Fakultät für Elektrotechnik eingesetzt.

Virtual Reality Center

Das dritte Einsatzgebiet ist das Virtual Reality Center von Herrn Prof. Dr. Burbaum von der Fakultät für Maschinenbau. Hier kommt ein Netzwerkrenderer und das COVISE Modul zum Einsatz. Details zur eingesetzten Software und deren Anwendungsgebiete sind nachfolgend in Tabelle 1 aufgeführt.

Applikation	Multiphysics	OpenFOAM	Netzwerk-rendern	COVISE Module
Vernetzungs-voraussetzung	Keine	Keine	Ethernet, Erreichbarkeit der Knoten untereinander	Ethernet, Erreichbarkeit der Knoten untereinander
Betriebs-system-voraussetzung	Keine	Keine	Windows	Windows, Linux
Einzusetzende Software	Multiphysics von Comsol	Matlab, Scilab	Backruner	gcc
Beschreibung der Applikation	Multiphysics ist eine Applikation zur Simulation physikalisch-er Vorgänge, welche mittels Differenzial-gleichungen beschrieben sind.	OpenFOAM ist eine Open Source Finite Elemente Toolbox, welche bisher für Mathlab und Scilab verfügbar ist.	Diese Applikation rendert auf mehrere Rechner verteilt Virtual Reality Szenen und CAD-Geometrie zu Animationen und Bildern.	COVISE steht für Col-laborative Visualization and Simulation Environ-ment. COVISE ist ein flexibles Visualisier-ungssystem, mit dem die Visualisierung wissen-schaftlicher Daten, wie sie bei Virtual Reality Anwendungen vorkom-men, am Virtual Realitiy Center erfolgt. COVISE bietet eine umfangreiche Bibliothek von Pro-grammteilen an, den sog. Modulen, mit der Mög-lichkeit diese zu einem Datenflussnetz interaktiv zu verbinden. Die einzelnen Module sind hierbei auf beliebige im Netzwerk erreichbare Rechner verteilt.

Tabelle 1: Details zur eingesetzten Software

1.3.1 Finite Elemente Methode

Im Bereich der Finite Elemente Methode kommt das Programm Multiphysics von Comsol zum Einsatz. Multiphysics dient zur Simulation physikalischer Vorgänge.

Auf einem 64-bit System muss unter Linux mindestens der Kernel mit der Version 2.4.x verwendet werden. In der 64-bit Version werden offiziel SUSE, RedHat Enterprise und Fedora unterstützt.

1.4 Architekturziele Power Grid Exist

Die Architektur von PGE entspricht der Master-Worker-Architektur, der allgemeinsten Struktur zur Lösung von parallelen Problemen [B 08] [BBKS 08]. Die Grundarchitektur von PGE orientiert sich an der Cell Broadband Engine Architecture (CBEA) [KDH 05], welche das Master-Worker-Schema implementiert. Dementsprechend wurde diese Architektur nachempfunden und mit normalen Intel Server-Prozessoren nachgebildet.

1.5 Forschungs- und Entwicklungsziele

Um auf die Bedeutung des PGE in der Forschung einzugehen, werden diesem Thema hier eigene Abschnitte gewidmet.

1.5.1 Master-Worker-Schema

Das Master-Worker Schema liegt als C-Bibliothek vor [BM 06]. Diese Bibliothek eignet sich für ein Rahmenwerk zur parallelen bzw. Master-Worker-Programmierung. Dieses Rahmenwerk lässt sich dann leicht als Webservice und als „platform as a service“ oder besser gesagt, als „Master Worker as a Service“ anbieten.

Das an der Fakultät für Informatik verwendete Cluster besteht aus insgesamt 14 Knoten, wobei zwei Knoten als Master zum Einsatz kommen. Die restlichen 12 Knoten sind somit die Worker, welche vom Master die Arbeiten aufgetragen bekommen.

1.5.2 Framework MapReduce

Das von Google entwickelte MapReduce ist ein System, welches die zu verarbeitenden Daten in kleine Stücke aufteilt und diese dann parallel auf mehreren Rechnern abarbeiten lässt. Einsatz findet dieses System bevorzugt in großen Cluster Systemen, aber auch in Mehrkernarchitekturen mit einem gemeinsamen Speicher. Dieses Verfahren erzielt eine sehr gute Performanz. Auch skaliert das System sehr gut, indem man einfach weitere Rechner zum Cluster hinzufügt oder aber zusätzliche Kerne verwendet [DG 08].

Die Idee hinter MapReduce basiert auf den Map und Reduce Funktionen von funktionalen Programmiersprachen wie in Lisp. Die Map Funktion erwartet als Input ein key/value Paar

und erzeugt daraus mehrere key/value Paare und gruppiert alle Werte (values) mit dem gleichen Schlüssel (key). Diese Zwischenergebnisse dienen als Übergabewert für die Reduce Funktion. Die Reduce Funktion akzeptiert als Übergabeparameter einen Schlüssel (key) und mehrere dazu assoziierte Werte (values). Anschließend versucht der Reducer aus den erhaltenen Daten eine kleinere Datenmenge zu erzeugen.

Die Verteilung und das Parallelisieren der Funktionen auf dem Cluster verwaltet ein Master Prozess. Mit Hilfe eines Schedulers verteilt er die Mapper oder Reducer Jobs auf dem Cluster. Die Mapper speichern nach Abschluss ihrer Aufgaben die Ergebnisse lokal ab und informieren den Master über den Status des Jobs und den Speicherort der Zwischenergebnisse. Daraufhin erhalten die Reducer die nötigen Informationen und liefern die gewünschten Ergebnisse. Die Implementierung von MapReduce existiert in verschiedenen Programmiersprachen wie Java, Perl, Ruby, Python, PHP, R, oder C++. MapReduce eignet sich besonders gut für Such- und Patternfunktionen in Daten.

Amazon bietet das darauf aufbauende Elastic MapReduce als Web Service an [AMR 09]. Amazon nutzt dabei Elastic Compute Cloud (EC2) [AEC 09] und Simple Storage Service (S3) [AS3 09].

1.5.3 Projekt: Hadoop as a Service

In Kooperation mit dem Steinbuch Center for Computing (SCC) des Karlsruher Instituts für Technologie (KIT) wird von März bis Juni 2010 im Rahmen einer Bachelor-Thesis von Herrn Maximilian Hoecker Hadoop as a Service auf Eucalyptus realisiert. Das Ziel ist hierbei, eine Open Source Version von Amazon Elastic MapReduce zu erhalten. Die Implementierung erfolgt mit Hilfe der Hadoop-Distribution Cloudera [CL 09]. Im Rahmen der Arbeit wird auch die Integration von Hadoop as a Service in PGE evaluiert und gegebenenfalls realisiert. Die Betreuung von Seiten des SCC erfolgt durch Dr. Marcel Kunze und Christian Baun.

Kontaktdaten:

Maximilian Hoecker: maximilian.hoecker@stud.hs-mannheim.de

Christian Baun: baun@kit.edu

Dr. Marcel Kunze: kunze@kit.edu

1.5.4 Projekt: GORBA Benchmarks

Das Institut für Angewandte Informatik (IAI) des Karlsruher Instituts für Technologie (KIT) entwickelt den global optimierenden Grid Resource Broker GORBA weiter zu einem adaptiven System, das sich mit seinen Optimierungsverfahren für das Grid Scheduling über eine Komponente zur Metaoptimierung an den jeweiligen Gridzustand anpasst. Der

Gridzustand wird dabei durch eine Reihe von anwendungs-, ressourcen- und lastbezogenen Kennzahlen beschrieben.

Zum Test der Metaoptimierung sind zahlreiche Benchmarkläufe für unterschiedliche Zustände und Optimierungsverfahren geplant, die parallel ablaufen sollen. Das soll unter anderem auf dem PGE durchgeführt werden. Frau Sylvia Strack von der Hochschule Mannheim wird Ende 2010 ihre Bachelorarbeit auf diesem Gebiet durchführen. Die Betreuung im KIT wird durch Dr. Karl-Uwe Stucky und Dr. Alexander Quinte (beide IAI) erfolgen.

1.5.5 Projekt: Massiv Parallele Berechnungen

Aufbauend auf der Diplomarbeit von Hr. Wolfgang Knobloch „Parallelisierung des Smith Waterman Algorithmus basierend auf der CUDA Entwicklungsplattform“ [K 09] sollen massiv parallele Berechnung von DNA-Proben Spezifitäten auf dem Cluster PGE durchgeführt werden. Diese Arbeiten laufen im Rahmen einer Diplom-/Masterarbeit und in Zusammenarbeit mit Dr. Kai Mohrhagen, devSystems // sharp solutions, Robert Bosch Strasse 7, 64293 Darmstadt.

1.5.6 Cloud Computing

Ein weiteres Entwicklungsziel ist, Scientific Computing as a Service anzubieten. Um dies zu erreichen, soll das vorhandene Cluster PGE als privates Cloud (Internal Cloud bzw. Intra Cloud) fungieren und bei Bedarf an Rechenleistung um ein Public Cloud erweiterbar sein [BKNT 10].

Basis des Cloud Computing sind die Virtualisierung der Rechenressourcen [BKL 09] und die Web-Service-Technologie. Die Realisierung einer Virtualisierungsschicht auf PGE und damit die Überführung von Power Grid Exist in ein Power Cloud Exist ist Gegenstand eines in 2010 gestellten Antrags zur Finanzierung aus Studiengebühren.

1.5.7 Fehlertoleranzkonzept

Die redundante Auslegung von PGE mit zwei Masterknoten, erlaubt Fehlertoleranzkonzepte von Cluster zu testen, erproben und weiterzuentwickeln.

2. Architektur

Eine detaillierte Beschreibung sowohl der Hard- als auch der Softwarearchitektur des PGE wird in den folgenden Abschnitten anhand von existierenden Technologien unter Verwendung von grafischen Abbildungen vorgenommen.

Hierfür werden zuerst Grundlagen über den relevanten, derzeitigen Stand der Technik gegeben, um schließlich auf die konkreten, verwendeten Technologien innerhalb des Clusters einzugehen.

2.1 Allgemeine Architektur des PGE

Die Hardware von PGE orientiert sich an der Verbreitung der beim Supercomputing eingesetzten Hardware. Gemäß der TOP 500-Liste [TOP500 09] eingesetzten Hardware sind derzeit 79,8% aller eingesetzten Prozessoren von Intel. Mit 11% folgen Power-Prozessoren von IBM und mit 8% Prozessoren vom Typ AMD x86_64. Unter den eingesetzten Intel-Prozessoren gibt es eine Reihe verschiedener CPU-Generationen im Einsatz. Im gesamten Kontext stellt die XEON E54xx – Familie (Harpertown) mit 40% den höchsten Anteil dar. Die XEON L54xx – Familie (Harpertown) ist mit 11%, gefolgt von den beiden Vorgängern XEON 51xx (Woodcrest) mit 9,8% und XEON 53xx (Clovertown) mit 8%, die zweithäufigste Prozessorfamilie im Supercomputing. Erst im Anschluss taucht der Opteron Quad Core mit 5,6% Anteil auf.

Bei der Vernetzung kommt das Gigabit Ethernet mit 56,4% und Infiniband mit 30.2% zum Einsatz. Der größte Teil anderer eingesetzter Vernetzungstechnologien ist proprietär [TOP500 09].

2.2 Detaillierte Architektur des PGE

Mit dem aus 28 Quadcore-Prozessoren des Typs Intel L5410 und einem Gigabit Ethernet-Netzwerk folgt PGE dem aktuellen Trend bezüglich des Prozessors und der Vernetzung.

Das PGE-Cluster besteht aus insgesamt 7 Recheneinheiten mit jeweils zwei Knoten. Jeder Knoten ist hierbei mit einer on-board-Ethernet- (2 Netzwerkanschlüsse) und einer externen Ethernetkarte (4 Netzwerkanschlüsse) ausgestattet. Knoten 1 und Knoten 3 stellen hierbei die Zugangsknoten des Clusters für den Anschluss an das Hochschulnetz dar und sind mit jeweils vier Kabeln über die externe Ethernetkarte mit jeweils einem Switch verbunden. Als Zugang zum Hochschulnetz nutzt der Master den on-board-Anschluss. Die restlichen 12 Knoten des Clusters sind mit jeweils sechs Kabeln gleichmäßig auf die beiden Switches verteilt.

Jeder Knoten enthält zwei XEON L5410-Prozessoren mit jeweils 4 Kernen. Beide Prozessoren arbeiten auf einem 8 GB großen Arbeitsspeicher und teilen sich eine 250 GB große SATA Revision 2.x Festplatte (siehe Abbildung 1).

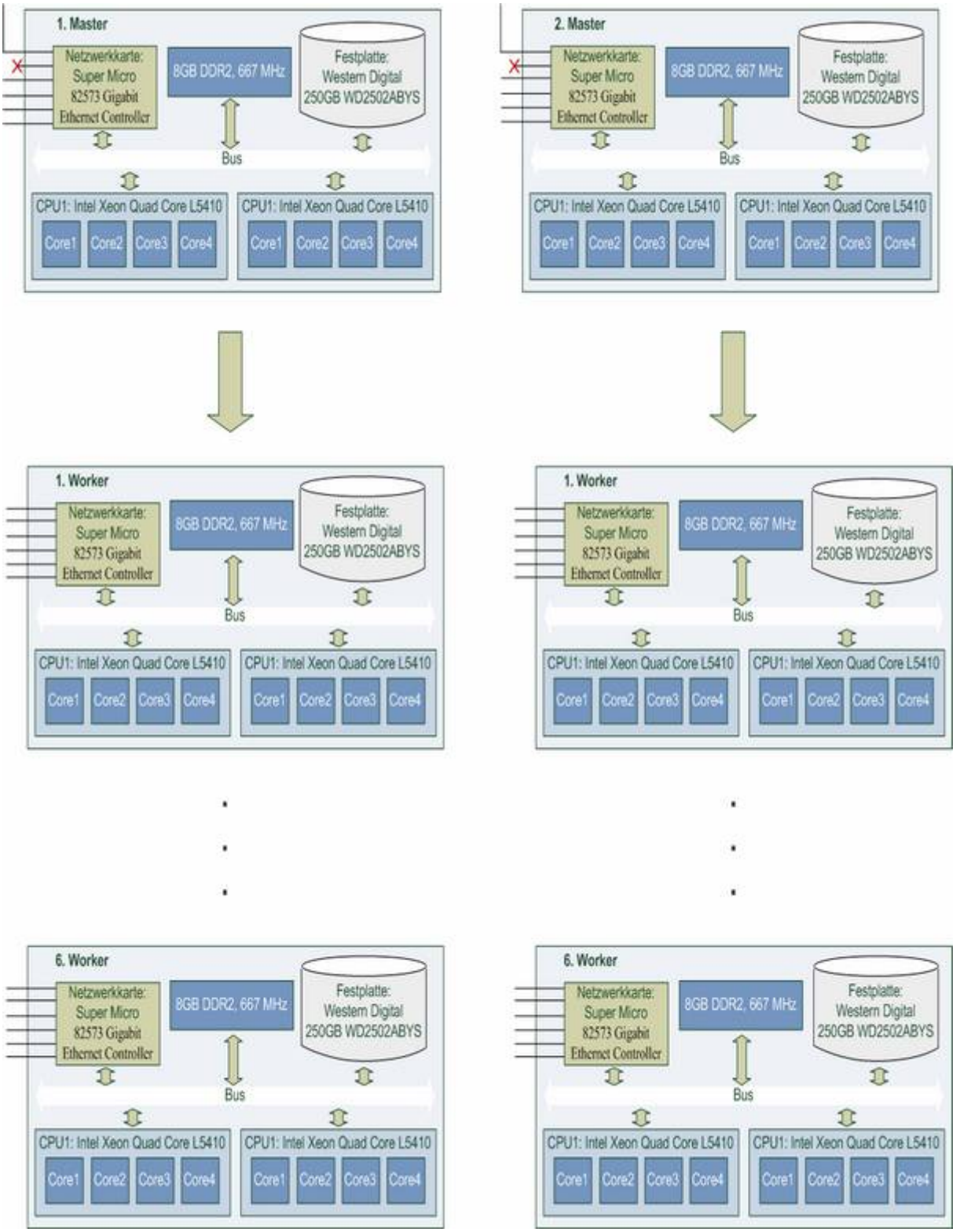

Abbildung 1: Architektur des PGE-Clusters

2.2.1 Master-Worker-Architektur

Der Aufbau des PGE orientiert sich am Master-Worker Schema. Die zwei Zugangsknoten des Clusters, die Header-Nodes, dienen als Master. Sie sind für die Verteilung der Jobs

zuständig und dienen als Softwareentwicklungsplattform zur Entwicklung paralleler Software sowie für alle Clusterdienste. Jedem Master sind sechs weitere Knoten als Worker zur Verfügung gestellt. Diese bekommen die Jobs von den Mastern zugeteilt und sind ausschließlich für die Berechnungen des Clusters verantwortlich.

2.2.2 Nachbildung des Cell-Prozessors

Nach dem Master-Worker Schema arbeitet auch der Cell-Prozessor. Dieser besteht aus einem Power Processing Element (PPE) und acht Synergistic Processing Elements (SPE), welche über einen Hochgeschwindigkeitsbus verbunden sind und die Rechenlast von dem PPE übernehmen. Das PPE und jedes SPE hat einen lokalen Speicher.

Die PGE-Architektur ist der Cell-Architektur nachempfunden. Hierbei stellen die Master-Knoten die PPEs und die jeweiligen sechs Worker die SPEs dar. Ebenso verfügt jeder Knoten über einen eigenen lokalen Speicher. Der Vorteil dieser Architektur liegt speziell im Bereich breitbandiger Berechnungsanwendungen. Der größte Nachteil eines Cell-Prozessors ist die Programmentwicklung, da diese an die hierfür bereitgestellte CellSDK oder CellBE gebunden ist. Durch die aktuelle Architektur, durch Intel-Prozessoren realisiert, nutzen wir den Vorteil des Cell-Prozessors und können unabhängig von einer vordefinierten Programmiersprache Software entwickeln [IBM 09].

2.2.3 Prozessor

Der Quad-Core-Prozessor L5410 „Harpertown“ für den Sockel LGA771 basiert auf Intels 45-nm-Penryn-Architektur. Diese stellt den Nachfolger der Core2-Architektur dar und hat eine verbesserte Mikrostruktur, eine erhöhte Cacheassoziativität und neben weiteren Detailverbesserungen eine gesteigerte Energieeffizienz.

Die CPU hat eine Taktfrequenz von 2,33 GHz und besitzt neben einem 64 kB großen L1-Cache (32 kB instruction Cache und 32 kB write-back-data Cache) einen 12 MB (2 mal 6 MB) großen Level 2 Cache. So teilen sich 2 Kerne einen L2-Cache, was im Bereich der verteilten Anwendungen die Möglichkeit zur gezielten Optimierung des Multithreading auf Gruppen von 2 Kernen bietet. Hierdurch spart man Speicherzugriffe ein und vermeidet lange Wartezyklen der Prozessorkerne bei mathematischen Berechnungen. Die Mikroarchitektur ermöglicht es, den Kernen dynamisch Cache-Bereich zuzuordnen, was gerade bei inaktiven Kernen von Vorteil ist.

Der L5410 arbeitet mit einem Front Side Bus (FSB) von 1333 MHz, um Datentransferraten von bis zu 10,66 GB pro Sekunde zu ermöglichen [INT 09]. Der Aufbau dieses Prozessors ist in Abbildung 2 zu sehen.

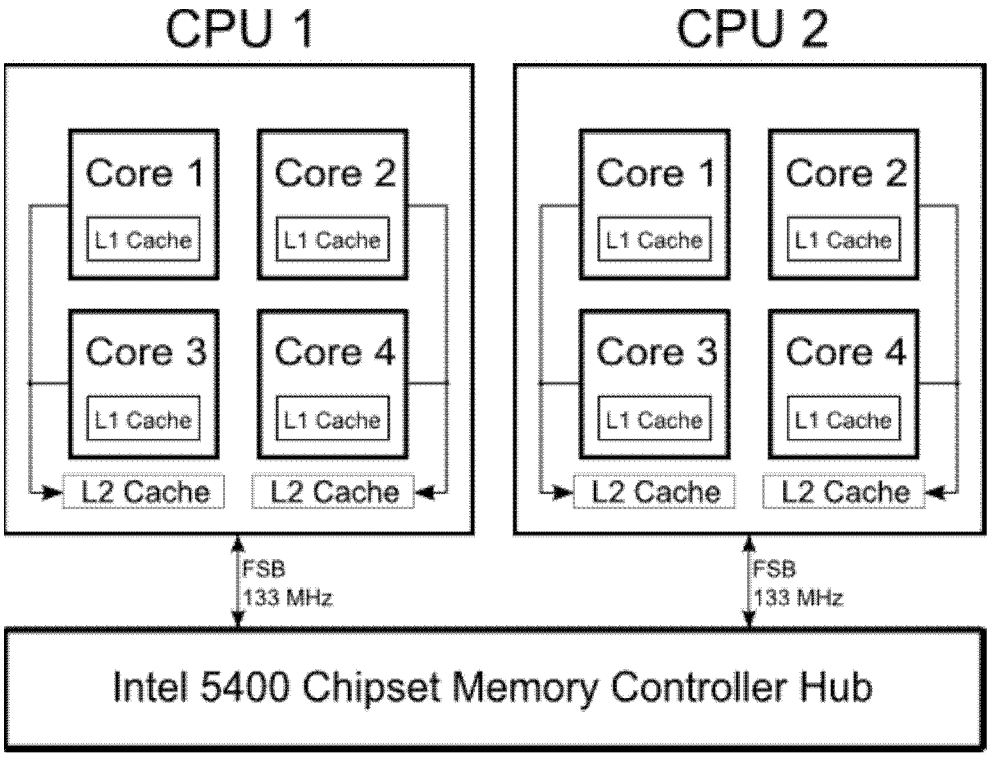

Abbildung 2: Aufbau des L5410

2.2.3.1 Superskalarität

Die Mikroarchitektur des XEON L5410 ist 4-fach superskalar ausgelegt, was Intel als Wide Dynamic Execution bezeichnet. Wide Dynamic Execution ist eine Kombination von Datenflussanalyse, spekulativer, out-of-order (außer der Reihe) und simultaner Ausführung von Befehlen. Dies führt dazu, dass die CPU pro Taktzyklus bis zu vier Instruktionen verarbeiten kann. Hierbei kann die CPU mindestens vier Befehle gleichzeitig holen, dekodieren, ausführen und die Ergebnisse in den L1-Daten-Cache übertragen. Dabei handelt es sich um eine Erweiterung der Vorgängerarchitekturen, welche mit nur 3 Befehlen pro Takt arbeiteten [INT 09].

2.2.3.2 Befehlssatz

Der L5410 verfügt neben den vorangegangenen Streaming SIMD Extensions (SSE) über den neuen SSE4.1 Befehlssatz. Dieser ist der erste Teil des SSE4 Befehlssatzes, in welchem 54 neue Befehle, vorwiegend zur Beschleunigung der Video-Bearbeitung, implementiert sind. SSE4.1 (Version 1) wurde mit dem Penryn-Prozessor eingeführt und umfasst 47 Befehle der neuen Erweiterung [INT 09].

2.2.3.3 Energieeffizienz

Durch die Penryn-Architektur nimmt die Fläche der CPU bei größerem Cache weniger Platz ein als beim Vorgänger E5345 „Clovertown“, welcher in einem 65 nm Fertigungsprozess

hergestellt wurde. Dies führt zu einer Verminderung der Leistungsaufnahme der CPU und zur Steigerung der Energieeffizienz, was im Server- und Clusterbereich wegen der langen Laufzeiten der Systeme und deren hoher Auslastung entscheidende Vorteile in Bezug auf entstehende Abwärme und Energiekosten bietet. Mit 50 Watt verbraucht der L5410 30 Watt weniger Leistung als sein Vorgänger „Clovertown" mit 80 Watt.

Der Prozessor verwendet eine verbesserte Variante der SpeedStep-Technologie und des Demand Based Switching, Intel Intelligent Power Capability genannt. Mit dieser Technologie lässt sich die Leistungsaufnahme bei geringer Prozessorlast dynamisch durch Umschaltung der Taktfrequenzen und Betriebsspannungen minimieren [INT 09].

2.2.3.4 Virtualisierung

Der L5410 unterstützt Intels VT-x Virtualisierungstechnologie, welche die Entwicklung und den Betrieb mehrerer virtueller Maschinen auf einem physikalischen System begünstigt. Die Architektur verfügt über Möglichkeiten, Unterbrechungen unterschiedlichen Betriebssystemen zuzuordnen und erlaubt Gast-Betriebssystemen kontrollierten und direkten Zugriff auf Systemressourcen, wodurch virtuelle Systeme effizienter und leistungsfähiger arbeiten. Die virtuellen Maschinen greifen auf einen virtuellen Maschinen-Monitor zu, der die Rolle eines privilegierten Operators und Verwalters übernimmt und die Zugriffe der weniger privilegierten virtuellen Maschinen steuert. Dieses einheitliche Bindeglied minimiert die Wahrscheinlichkeit unzulässiger Zugriffe auf Hardwareressourcen. Die Realisierung dieser Funktionalität auf Hardwareebene ermöglicht dabei ein hohes Maß an Geschwindigkeit und Effizienz.

2.2.3.5 Vergleichbare Prozessoren und „Nehalem"

Um einen Überblick über andere, gegenwärtige Prozessoren oder Prozessorarchitekturen zu erhalten, gibt dieser Abschnitt Aufschluss über die für das PGE relevanten Alternativen.

Cell-Architektur von IBM

Im Vergleich zu den universellen Intel und AMD Prozessoren bieten IBM´s Cell Prozessoren, bestehend aus einem PowerPC-Kern und acht SPE-Units, Vorteile im Bereich der breitbandigen Datenberechnung. Durch spezielle Optimierung auf diese Multicore CPU lassen sich Leistungsvorteile bei der parallelen Berechnung hoher Datenmengen erzielen. Nachteil ist die Abhängigkeit bei der Softwareentwicklung, welche den Einsatz des Cell Broadband Engine SDK voraussetzt [IBM 09].

Tesla-Architektur von Nvidia

Nvidias Tesla Architektur zielt ebenso auf breitbandige Berechnungen ab. Hier handelt es sich jedoch um keine dedizierte Architektur, sondern um bestehende PC/Server Systeme welche mit bis zu 4 Teslakarten ausgestattet sind. Diese sind mit einer sehr leistungsfähigen

Graphics Processing Unit (GPU) bestück. Durch die hohe Spezialisierung der Prozessoren und den hohen Grad an Parallelität lassen sich im Bereich komplexer, paralleler, wissenschaftlicher Berechnungen deutlich Leistungssteigerungen erzielen. Die Tesla Technologie eignet sich besonders als potentielle Erweiterung für bestehende XEON-Systeme, um entsprechende Ressourcen bereitzustellen. Zur Entwicklung von Anwendungen steht die von Nvidia entwickelte Compute Unified Device Architecture (CUDA)-Schnittstelle zur Verfügung [NVID 09].

„Nehalem" von Intel

Der Trend im Prozessor-Bereich geht Richtung Intel XEON 5500 „Nehalem", eine von Intel entwickelte Architektur. Diese verfügt über einen neuen Level 3-Cache, den sich alle Kerne teilen. Durch die veränderte 3-stufige Cacheverwaltung verbessert sich die Cacheeffizienz. Anstelle eines FSB verfügt der neue Prozessor über ein Punkt-zu-Punkt-Interface für Verbindungen (QuickPath Technology). Des Weiteren steigert sich die Performanz dieser Systeme durch einen separaten Memory-Controller und den vollständigen SSE4 Befehlssatz [INT 09].

2.2.4 Speicher

Bei einem Multicore-Prozessor verwenden die Prozessorkerne den Speicher gemeinsam, was bedeutet, dass die Kerne sich den Arbeitsspeicher selbst einteilen. Die Größe des Arbeitsspeichers korreliert hierbei stark mit den Anforderungen an die Parallelität der Prozesse und der Größe der Datensätze. Falls mehrere Prozesse parallel arbeiten, sollten diese für schnellen Kontextwechsel im Arbeitsspeicher vorliegen. Um eine schnelle Bearbeitung einzelner Prozesse zu gewährleisten, welche immer wieder auf größere Datenmengen zugreifen, sollten diese Daten sich ebenso vollständig im Arbeitsspeicher befinden. Hierbei zeichnet sich Ubuntu aus, indem es weitaus ressourcenschonender arbeitet, als zum Beispiel Windows, und bei der Speicherverwaltung den physikalischen Speicher um einen virtuellen Speicher erweitert. Falls der virtuelle Speicher bis zu einer bestimmten Prozentzahl gefüllt ist, erfolgt eine Auslagerung bestimmter Seiten auf die Festplatte. Dadurch kommen hohe Lese- und Schreibzugriffe zustande. Im PGE verfügt jeder Knoten über acht Gigabyte (GB) Arbeitsspeicher. Damit ist ausreichend Arbeitsspeicher zum Einlagern von Seiten vorhanden, ohne das Gesamtsystem zu verlangsamen.

Die Stabilität des Gesamtsystems hängt unter anderem von der Wahl der Arbeitspeichertechnologie ab. Das im PGE-Cluster eingesetzte Board von Intel besitzt einen Error Correction Code (ECC)- Controller. Bei Serversystemen, welche im Normalfall auf dauerhaften Betrieb ausgelegt sind, ist dies angebracht. Im Dauerbetrieb treten mit der Zeit Fehler in den einzelnen Bits der Speicheradressen auf. Diese können zu kritischen Systemabstürzen (Befehlssatz) führen oder die Daten verfälschen (Datensatz). Solche Fehler können zum Beispiel durch elektromagnetische Strahlung (Soft Errors) oder defekte

Hardware auftreten. Die ECC-Technologie bietet ein Prüfbit zur Korrektur von 1-Bit-Speicherfehlern und zur Überprüfung auf 2-Bit-Speicherfehlern an. PGE verwendet ECC-fähigen Speicher, um solche Fehler zu korrigieren und dadurch einen dauerhaft stabilen Betrieb zu gewährleisten.

Das sich im Einsatz befindliche Mainboard unterstützt Busgeschwindigkeiten von 533 Megahertz (MHz) und 667 MHz. Durch den Einsatz von 667 MHz erreicht man eine Datenübertragungsfrequenz mit einer Speicherzugriffszeit von 1,5 ns.

2.2.5 Platte

Die in den einzelnen Knoten verwendeten Festplatten sind von der Firma Western Digital, mit 250 GB Speicherkapazität und vom Typ Serial Advanced Technology Attachement Revision 2.x (SATA 3Gb/s). Der Datendurchsatz liegt bei bis zu 300 MB/s und kann somit mit dem wesentlich teureren Small Computer System Interface (SCSI) 320 Standard (320 MB/s) mithalten. Ebenso unterstützt der SATA Revision 2.x Standard HotSwap, was einen Austausch der Festplatten im laufenden Betrieb ermöglicht und Staggered Spinup, welches ein zeitverzögertes Einschalten mehrerer Laufwerke unterstützt, um zum Beispiel das Netzteil nicht zu überlasten.

Da jeder Knoten mit einer eigenen Festplatte ausgerüstet ist und nicht alle Daten über einen Speicher-Server erhalten, wird das Netzwerk entlastet. Dadurch können die Prozesse ihre Zwischenergebnisse zur Weiterverarbeitung lokal speichern und sparen somit Zugriffe über das Netzwerk zum globalen Speicher-Server ein.

Um die Performance und Ausfallsicherheit bezüglich der Festplatten zu optimieren, ist eine Erweiterbarkeit auf SATA Raids im späteren Verlauf des Projekts möglich. Der Raid Verbund mehrerer Festplatten unterstützt zum einen erhöhte Lese- und Schreibgeschwindigkeiten und erhöht zum anderen die Ausfallsicherheit.

2.2.6 Filesystem

In der Konfiguration von Power Grid Exist sind der Datenhaushalt und der Datenaustausch durch den Master-Knoten gewährleistet. Der Master fungiert als Dateiserver und stellt somit Daten und Programme für alle Worker-Knoten bereit. Durch den Einsatz eines Netzwerk-Dateisystems (Network File System – NFS) stellt der Master-Knoten einen Ordner im Netzwerk bereit. Auf diesen Ordner können sich die Worker-Knoten verbinden. Dies ermöglicht den Worker-Knoten den Zugriff auf Daten des Masters, gleich einem Zugriff auf Daten einer lokalen Festplatte.

Durch diese Konfiguration eines verteilten Dateisystems (Distributed File System – DFS) ist eine Synchronisation der Knoten nicht notwendig.

Von den Worker-Knoten benötigte Daten werden vom Master bereitgestellt, bevor die Abarbeitung eines Prozesses auf Power Grid Exist erfolgt. Daten, welche vom Worker zu bearbeiten sind, speichert dieser lokal und kopiert sie im Anschluss in einen vom Master bereitgestellten Ordner. Daten-Backups funktionieren dadurch einfach, da nur die Daten auf dem Master-Knoten zu sichern sind. Die Knoten verfügen nur temporär, während des Abarbeitens eines Prozesses über Daten und sind daher nicht vom Backup-Prozess zu betrachten.

2.3 Externe Vernetzung – Anbindung an das Hochschulnetz

In den nächsten Abschnitten wird erläutert, auf welche Weise das Cluster nach außen verfügbar gemacht wird und wie der Zugriff darauf erfolgen kann.

2.3.1 Anbindung

Das PGE verfügt über zwei Gigabit-Ports, welche für die externe Anbindung konfiguriert wurden. Jeder Master-Knoten stellt einen Netzwerkanschluss für die Anbindung an das Hochschulnetz bereit. Durch diese Konfiguration besteht die Möglichkeit, über zwei verschiedene Internetprotokoll-Adressen (IP-Adressen) auf das PGE zuzugreifen.

2.3.2 Ausfallsicherheit

Durch die zwei voneinander unabhängigen Anschlüsse des PGE ist im Falle eines Ausfalles eines Master-Knotens die Arbeitsfähigkeit auf dem Cluster weiterhin gewährleistet. Durch diese Konfiguration wurde das Single Point of Failure Problem im Bezug auf die Verfügbarkeit des PGE beseitigt. So ist das Cluster auch bei Ausfall eines Master-Knotens noch erreichbar.

Durch die direkte Verbindung der Master-Knoten mit dem Hochschulnetzwerk bleiben nach einem Ausfall eines oder beider Switche die Verbindungen zu den Master-Knoten weiter bestehen.

2.3.3 Zugriff auf das PGE

Der Zugriff auf PGE erfolgt über einen der beiden Master-Knoten. Da nur diese an das Hochschulnetzwerk angebunden sind, muss man sich erst mit einem Master verbinden. Im Anschluss sind Berechnungen oder Programmausführungen auf anderen Workern möglich.

Die Ausführung aller Zugriffe ist durch das Secure Shell (SSH) Protokoll verschlüsselt. Die Verbindung zum Cluster ist aus dem Hochschulnetzwerk heraus möglich. Außerhalb des Hochschulnetzwerks erlangt man nur per VPN-Verbindung Zugriff auf einen Master-Knoten.

2.4 Interne Vernetzung

Der Aufbau der internen Vernetzung, einschließlich der verwendeten Hardware, wird in den Folgeabschnitten beschrieben.

2.4.1 Trunking

Trunking, auch bekannt als „Link Aggregation“, „Port-Trunking“ oder „Channel Bundling“, ist unter dem Standard IEEE 802.3ad standardisiert.

Bei diesem Verfahren arbeiten mehrere physikalische Leitungen zusammengefasst als eine logische Leitung. Das Verfahren funktioniert nur auf der Full-Duplex-Ebene. Bei Ausfall einer Leitung verteilt sich die Last gleichmäßig auf die verbliebenen Leitungen.

Zunächst findet auf Schicht 2 des Open Systems Interconnection (OSI) Schichtenmodell´s eine Zerlegung des zu sendenden Pakets in Frames statt. Danach teilt die Link Aggregation Control Layer (LACL) die Frames in kleinere Teilframes auf. Die Übertragung dieser Teilframes erfolgt auf unterschiedlichen Netzwerkports. Das Link Aggregation Control Protocol (LACP) stellt die Kommunikation zwischen den beiden Endpunkten dar. Diese Informationen sind im Frame enthalten. Außer dem IEEE 802.3ad Standard gibt es noch herstellerspezifische Protokolle. Alle Endpunkte müssen diesen Standard unterstützen.

Vorteil dieses Verfahrens ist die dynamische Einteilung der Frames durch das LACP, welche eine automatische Lastverteilung auf verbliebene Leitungen zur Folge hat, siehe Abbildung 3.

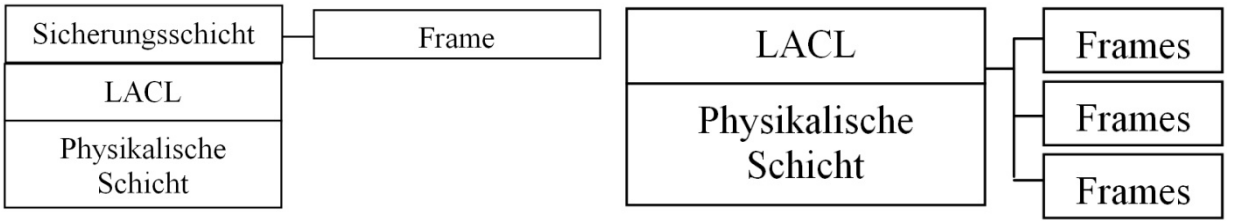

Abbildung 3: Arbeitsweise des Trunking (Elektronik-Kompendium)

2.4.2 Aufbau der Vernetzung

Zwei 48-Port-Gigabit Switches sind jeweils mit acht Gigabit-Ports über einen Trunk miteinander verbunden. Dies gewährleitstet ausreichend hohe Datenübertragungsraten zwischen den Workern von bis zu 8 Gb/s. Von den 14 Knoten sind die zwei Master mit jeweils einem Trunk, bestehend aus 4 Gigabit-Verbindungen, an die Switches angebunden. Hierbei wurde, bezogen auf die Ausfallsicherheit, jeweils einem Master ein Switch

zugeordnet. Mit jeweils 6 Gigabit Leitungen pro Knoten sind die Worker auf die zwei Switches verteilt angeschlossen.

Durch die Link Aggregation beider Switches ist ein ausreichend schneller Datentransfer zwischen beiden Switches gewährleistet. Die Knoten wurden so konfiguriert, dass alle sechs Ports (bzw. vier Ports bei den Mastern) mittels Bonding zusammengeschlossen sind. Dies hat den Vorteil, dass jeder Knoten nur über eine einzige statt über sechs IP-Adressen erreichbar ist und somit alle Netzwerkanschlüsse wie ein einziger Netzwerkanschluss agieren. Diese Konfiguration stellt eine schnelle Kommunikation zwischen den Knoten sicher. Den detaillierten Vernetzungsplan zeigt nachfolgende Abbildung 4 und die Konfiguration der Switches zeigt Anhang 1.

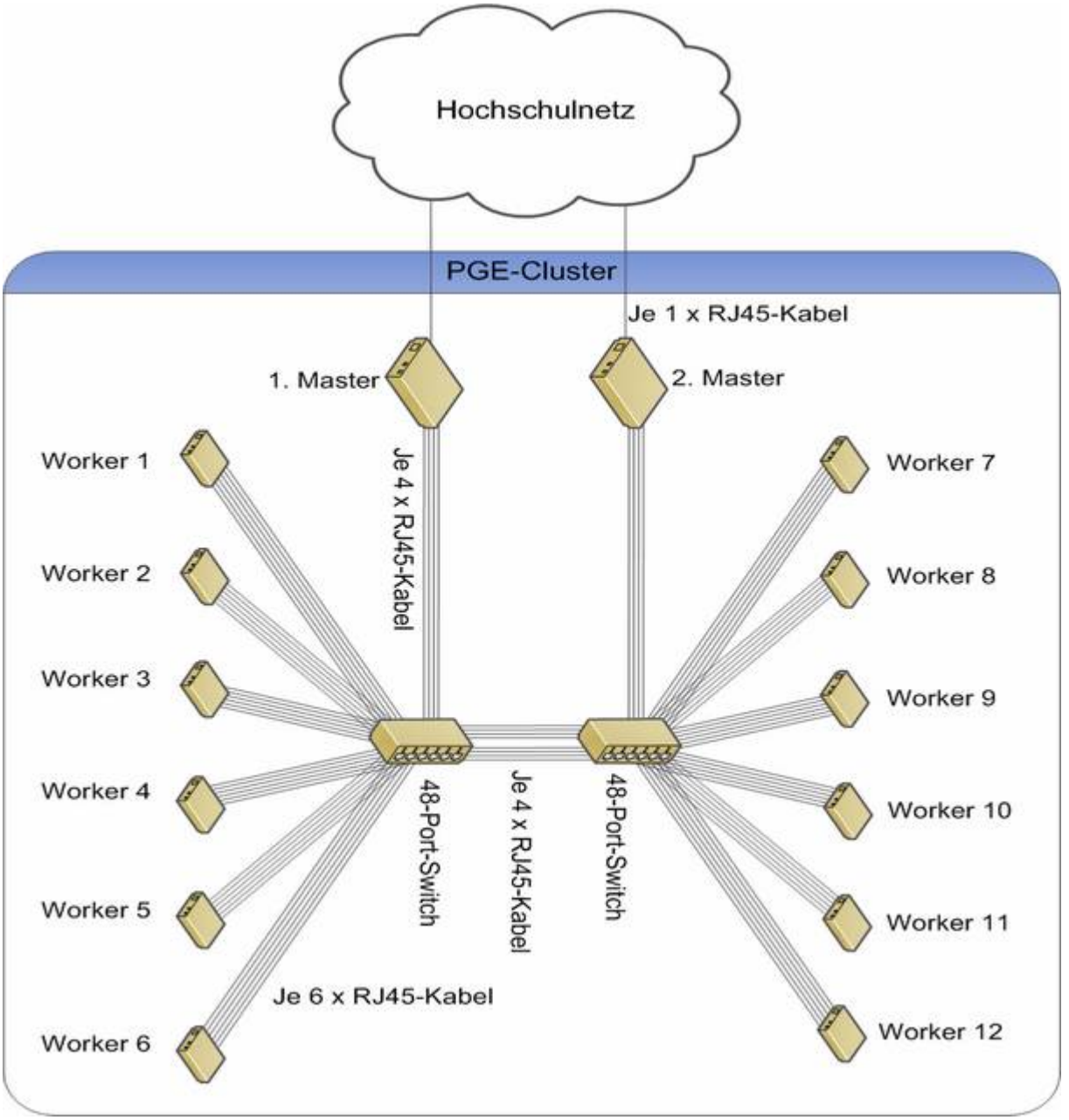

Abbildung 4: Vernetzung des PGE

2.4.3 Optimale Vernetzung und Ausfallsicherheit

Um Komplettausfälle zu vermeiden ist der Cluster fehlertolerant ausgelegt. Hierbei ist eine Reihe von Fehlerquellen zu beachten. Es können Master- oder Worker-Knoten ausfallen, Switches, Netzwerkkarten, -kabel oder die Anbindung an das Hochschulnetz. Durch die Anbindung beider Master-Knoten ist das Cluster bei einem Ausfall einer Zuleitung weiter ansprechbar. Fällt ein Switch aus, sind die diesem Switch zugeordneten sechs Knoten nicht mehr erreichbar. Die andere Hälfte kann aber normal weiterarbeiten. Bei Ausfall eines Master-Knotens übernimmt der andere Master die Aufgabenverteilung.

Diese Konfiguration stellt im Falle eines Ausfalles eines der Switches nicht sicher, dass alle Knoten weiterhin erreichbar sind. Um dies zu verhindern müsste jeder Knoten mit beiden Switches verbunden sein. Durch die Trunks ist dies nicht ohne weiteres möglich, da eine Trunkbildung über mehrere Switches hinweg nicht möglich ist.

2.5 Betriebssystem und clusterrelevante Software

Im Folgenden werden das eingesetzte Betriebssystem und die eingesetzte Software, inklusive deren Vor- und Nachteile, aufgeführt.

2.5.1 Eingesetztes Betriebssystem

Das auf dem Cluster eingesetzte Betriebssystem ist Ubuntu in der Version 8.04.3. Da es ein frei zugängliches Open-Source-Projekt darstellt, bietet Ubuntu eine kostengünstige Alternative zu anderen Linux Distributionen. Die Entwickler verfolgen mit Ubuntu das Ziel, ein einfach zu installierendes und leicht zu bedienendes Betriebssystem mit abgestimmter Software zu schaffen. Die aktuell eingesetzte Version kommt mit einer verlängerten Betreuung (LTS-Version, Long Term Support) und bietet somit voraussichtlich bis April 2011 vollen Support [UBU 09].

Der momentane Trend bezüglich des eingesetzten Betriebssystems auf Hochleistungsrechnern geht deutlich Richtung Linux. Laut der Liste der Top 500 Superrechner läuft auf 88% der Rechner Linux und gerade mal auf 1 % Windows [TOP500 09].

2.5.2 Installierte Software

Eine Beschreibung bereits installierter Software kann in den nächsten Abschnitten nachgelesen werden.

2.5.2.1 OpenSSH 4.7

OpenSSH ist ein freies Netzwerkverbindungs-Tool. OpenSSH verschlüsselt den gesamten Verkehr (inklusive Passwörtern), um Mithören (eavesdropping), Entführen von Verbindungen (connection hijacking) und andere Angriffe auf Netzwerkebene effektiv zu

eliminieren. Die momentan neuste Version 5.2 von OpenSSH unterscheidet sich von der OpenSSH-Version des Clusters hauptsächlich dahingehend, dass Fehler bei der Benutzung von X11-Forwarding behoben wurden und neue Features wie z. B. eine „Fingerprint"-Funktion oder neue Testmodi hinzugefügt wurden. Ein Upgrade auf die neuste Version ist nicht nötig, da die Version 4.7 als stabil und sicher gilt [OpenSSH 09].

2.5.2.2 C-Compiler gcc/g++/c++ 4.2.4

Die auf dem Cluster verwendete Version des GCC-Compilers ist 4.2.4. GCC steht für GNU Compiler Collection und ist der Name der Compiler-Suite des GNU-Projekts. Die Sammlung enthält Compiler für die Programmiersprachen C, C++, Java, Objective-C, Fortran 95 und Ada. Die Version 4.2.4 ist momentan die Aktuellste. Diese wurde gegenüber den letzten Releases dahingehend verbessert, dass openMP-basierende Programme besser kompilierbar sind, z.B. durch frühzeitige Erkennung von Endlosschleifen (Loops) und Bufferoverflows [GNU 09].

2.5.2.3 MPICH2 1.1.1p1

Das Message Passing Interface (MPI) erlaubt den Nachrichtenaustauch bei parallelen Berechnungen auf verteilten Computersystemen. Das Message Passing Interface (MPI) ist eine von IEEE standardisierte Kommunikationsbibliothek für die Programmiersprachen C, C++, Fortran77 und Fortran90. Das MPI Forum entwickelt MPI seit 1992. 1995 erschien MPI 1, die erste Version des Standards, 1997 erschien mit MPI 2 die zweite Version. Die erste Implementierung des MPI-1.x-Standards war MPICH. Mittlerweile stellt das Konsortium Open MPI MPICH2 zur Verfügung, welches den MPI-2.1-Standard implementiert [MPI 09].

2.5.2.4 OpenMP 2.5

Diese API dient zur Shared-Memory-Programmierung in C, C++ und Fortran auf Multiprozessor-Computern. Bei der Parallelisierung von Programmen, z. B. durch MPICH2, arbeiten diese auf Prozessebene, während bei OpenMP die Parallelisierung auf Thread- bzw. Schleifenebene stattfindet. Eine Eigenschaft von OpenMP ist, dass die Programme auch korrekt laufen, wenn der Compiler die OpenMP-Anweisungen nicht kennt und als Kommentar bewertet. Dies resultiert aus der Möglichkeit, mit OpenMP in mehrere Threads aufgeteilte for-Schleifen mittels eines einzigen Threads sequenziell abzuarbeiten. Die zum Zeitpunkt des Verfassens dieser Dokumentation aktuellste Version trägt die Versionsnummer 3.0. OpenMP installiert sich automatisch mit dem GNU C-Compiler 4.3.1. Die aktuell installierte C-Compiler-Version hat die Release-Nummer 4.2.4. Diese beinhaltet die OpenMP-Version 2.5. Zu den Verbesserungen der neusten Version seit 2.5 gehören zum einen die Beseitigung mehrerer Bugs und zum anderen die Verbesserung des Windows-Supports [OpenMP 09].

2.5.2.5 Nagios 3.2.0

Durch die Software Nagios (Network + Hagios), welche früher unter dem Namen NetSaint bekannt war, ist es möglich, komplexe IT-Strukturen zu überwachen. Nagios bietet eine Sammlung von Modulen zur Netzwerk-, Host- und speziell zur Serviceüberwachung, sowie ein Webinterface zum Abfragen der gesammelten Daten. Nagios kann den Status verschiedener Dienste, wie zum Beispiel SSH, FTP und HHTP, sowie den Festplattenplatz, die Speicher- und CPU-Auslastung oder die Uptime über diverse Module (Plug-ins) abfragen und auswerten. Sobald ein Dienst oder ein Host einen (teilweise einstellbaren) kritischen Wert erreicht oder nicht mehr erreichbar ist, alarmiert Nagios die Kontaktpersonen über beliebige Kanäle wie zum Beispiel E-Mail, SMS, Pager, IM-Messages und Telefonanrufe. Die installierte Version trägt die Release-Nummer 3.2.0. Diese hat sich seit den letzten Releases in folgenden Bereichen verbessert: Der Diagrammdarstellung, der Installation, der Kompatibilität mit verschiedenen Betriebssystemen sowie im Bereich der Update-Funktionalität [NAG 10].

2.5.3 Alternative Software

Um die Wahl mancher installierten Software zu begründen, dient dieser Abschnitt dazu, aufzuzeigen, welche Alternativen zur Wahl stehen, seien es andere Versionen ein und derselben Software, oder auch ganz andere Software, indem unter anderem Vergleiche mit anderen existenten Lösungen angestellt werden.

2.5.3.1 MPICH – Unterschiedliche Versionen

Dieser Unterabschnitt behandelt alternative Versionen von MPICH zur bereits installierten Version MPICH2 1.1.1p1.

2.5.3.1.1 Unterschiede zwischen MPICH1 und MPICH2

Ein wesentlicher Unterschied zwischen MPICH1 und MPICH2 besteht darin, dass MPICH1 die MPI 1 und die MPI-IO-Bibliothek und MPICH2 die MPI 2 Bibliothek implementiert. Bei MPI 2 kam zusätzlich die dynamische Prozesserzeugung und -verwaltung sowie die parallele Ein- und Ausgabe hinzu. Die MPI 2 Bibliothek hat im Gegensatz zur MPI 1 Bibliothek eine verbesserte Datentypverarbeitung. Zudem ist MPICH2 bei der Verarbeitung von einseitigen Operationen (one-sided-operations) und Punkt-zu-Punkt-Nachrichten (point-to-point messages) schneller als MPICH1. Im Vergleich zu MPICH1 liegt der Geschwindigkeitsgewinn von MPICH2 bei 28% pro 8 MB Nachrichten. Der Nachteil von MPICH2 besteht darin, dass es keine heterogenen Plattformen unterstützt, was wiederum bei MPICH1 der Fall ist. Bei MPI 1 muss die Anzahl von Prozessen einer MPI Anwendung konstant sein. Beim Start der Anwendung steht die Anzahl fest und ist im weiteren Verlauf nicht mehr änderbar. MPI 2 unterstützt eine dynamische Prozesserzeugung. Des Weiteren

können getrennt gestartete, unabhängige MPI Programme miteinander kommunizieren. Dies ermöglicht, Client Server-Anwendungen mit dynamischer Anzahl von Clients mit MPI 2 zu realisieren [MPI 09].

2.5.3.1.2 Verbesserungen der MPICH2 Version 1.1 gegenüber 1.0

Die aktuellste Version von MPICH2 trägt zum jetzigen Zeitpunkt die Versionsnummer 1.1.1p1. Diese ist gegenüber der Vorgängerversion MPICH2 1.0 stabiler im Umgang mit Schleifen, in denen Null-Werte (zero count types) vorkommen. Zudem wurden mehrere funktionelle Fehler, wie z.B. das Anlegen von doppelten Verlinkungen, in einigen Bibliotheken behoben. Des Weiteren wurde die Kompatibilität zwischen MPD und Python 2.3 verbessert. MPD ist notwendig, um einen Cluster-Node-Ring zu erzeugen. Ebenso wurden Speicher-Löcher (memory leaks) beseitigt und der Codereinigungsassistent verbessert [MPI 09].

2.5.3.1.3 Verbesserungen der MPICH2 Version 1.0 gegenüber 0.93

Eine wichtige Verbesserung von MPICH2 Version 1.0 besteht darin, dass die Funktionsbibliothek auch unter Solaris verfügbar ist. Zudem wurde die Installation vereinfacht, indem Routinen zur Überprüfung falscher Konfigurationseinstellungen, blockierender Firewalls oder auftretender Kommunikationsprobleme integriert wurden [MPIWIK 09].

2.5.3.2 LAM/MPI

LAM steht für Local Area Multicomputer und implementiert den Standard MPI 1 vollständig und den Standard MPI 2 fast vollständig. Es unterstützt TCP/IP, Infiniband und Myrinet-Netze. Zum Starten von MPI Programmen, und zum Teil auch für die Kommunikation, verwendet LAM/MPI einen Dämon. LAM/MPI unterstützt heterogene Cluster und ist gridfähig. Bei Grids kommt das Globus-Toolkit zum Einsatz. LAM/MPI stellt zur Kompilation und Ausführung eine komfortable Laufzeitumgebung zur Verfügung. Diese MPI-Funktionsbibliothek ist momentan nicht auf dem Cluster installiert, da OpenMP und MPICH2 für unsere Bedürfnisse völlig ausreichen [BBKS 08].

Literatur

[ACK 02] Anderson D. P., Cobb J., Horpela E., Lebofsky M., Werthimer D.: Seti@home: An Experiment in Public-Resource Computing. Communications of the ACM, Vol. 45, No. 11, Nov. 2002.

[AEC 09] Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2/, 2009.

[AMR 09] Amazon Elastic MapReduce, http://aws.amazon.com/elasticmapreduce/, 2009.

[AS3 09] Amazon Simple Storage Service (Amazon S3), http://aws.amazon.com/s3/, 2009.

[B 04] Bengel G.: Grundkurs – Verteilte Systeme: Grundlagen und Praxis des Client-Server-Computing – Inklusive aktueller Technologie wie Web-Services u.a. – Für Studenten und Praktiker. Vieweg+Teubner, 2004.

[B 08] Bengel G.: Power Grid Exist: A Dynamic Cluster with Hierarchic Master Worker Architecture. Informatik-Berichte Hochschule Mannheim – Fakultät für Informatik, 2008.

[BBKS 08] Bengel G., Baun C., Kunze M., Stucky K-U.: Masterkurs Parallele und Verteilte Systeme. Vieweg+Teubner Verlag 2008.

[BKL 09] Baun C., Kunze M., Ludwig T.: Servervirtualisierung. Informatik Spektrum, Band 32, Heft 3, Juni 2009.

[BKNT 10] Baun C., Kunze M., Nimis J., Tai S.: Cloud Computing, Web-basierte dynamische IT-Services. Springer Verlag 2010.

[BM 06] Bauke H., Mertens S.: Cluster Computing. Springer Verlag, 2006.

[CL 09] Cloudera Hadoop-Distribution, http://www.cloudera.com/, 2009.

[DG 08] Dean J., Ghemawat, S: MapReduce: simplified data processing on large clusters. Communications of the ACM, Vol. 51, No. 1, January 2008.

[GEM 07] Gschwind M., Erb D., Manning S., Nutter M.; An Open Source Environment for Cell Broadband Engine System Software. IEEE Computer, Vol. 40, No. 6, 2007.

[GHF 06] Gschwind M., Hofstee H.P., Flachs B., et. al.: Synergistic Processing in Cell's Multicore Architecture. IEEE Micro, Vol. 26, No. 2, March/April 2006.

[GNU 09] GCC 4.2 Release Series, http://gcc.gnu.org/gcc-4.2/changes.html, 17.08.2009.

[IBM 09] IBM: http://www.ibm.com/developerworks/power/library/pa-cellperf/, 17.08.2009.

[INT 09] Intel: http://www.intel.com, 17.08.2009.

[KDH 05] Kahle J. A., Day M. N., Hofstee H.P. et. al.: Introduction to the Cell Multiprocessors. IBM J. Research and Development, Vol. 49, No. 4/5, 2005.

[K 09] Knobloch Wolfgang: Parallelisierung des Smith Waterman Algorithmus basierend auf der CUDA Entwicklungsplattform. Diplomarbeit, Fakultät für Informatik, Hochschule Mannheim, Okt. 2009.

[KPP 06] Kistler M., Perrone M., Petrini F.: Cell Multiprocessor Communication Network: Built for Speed. IEEE Micro Vol. 26, No. 3, May/June 2006.

[MPI 09] MPI Release 1.1.1p1 Changes: https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.1.1p1/CHANGES, 17.08.2009.

[MPIWIK 09] Message Passing Interface: http://de.wikipedia.org/wiki/Message_Passing_Interface, 17.08.2009.

[NVID 09] Nvidia: http://www.nvidia.de/page/tesla_computing_solutions.html, 17.08.2009.

[OpenMP 09] OpenMP: http://openmp.org/wp/, 17.08.2009.

[OpenSSH 09] OpenSSH: http://openssh.com/txt/, 17.08.2009.

[TOP500 09] Top 500 List: http://www.top500.org/list/, 17.08.2009.

[UBU 09] Ubuntu: http://de.wikipedia.org/wiki/Ubuntu, 17.08.2009.

Anhang

Switch 1

Uplink to Switch 2				Node 14			Node 12			Node 10			Node 8			Node 6			Node 4			Node 3 (HN 2)	
P 1	P 3	P 5	P 7	P 9	P 11	P 13	P 15	P 17	P 19	P 21	P 23	P 25	P 27	P 29	P 31	P 33	P 35	P 37	P 39	P 41	P 43	P 45	P 47
S 2 Port 1	S 2 Port 3	S 2 Port 5	S 2 Port 7	N 14 eth 2	N 14 eth 3	N 14 eth 0	N 12 eth 5	N 12 eth 0	N 12 eth 4	N 10 eth 0	N 10 eth 2	N 10 eth 1	N 8 eth 1	N 8 eth 3	N 8 eth 2	N 6 eth 4	N 6 eth 2	N 6 eth 1	N 4 eth 4	N 4 eth 1	N 4 eth 5	N 3 eth 3	N 3 eth 1
P 2	P 4	P 6	P 8	P 10	P 12	P 14	P 16	P 18	P 20	P 22	P 24	P 26	P 28	P 30	P 32	P 34	P 36	P 38	P 40	P 42	P 44	P 46	P 48
S 2 Port 2	S 2 Port 4	S 2 Port 6	S 2 Port 8	N 14 eth 1	N 14 eth 5	N 14 eth 4	N 12 eth 2	N 12 eth 1	N 12 eth 3	N 10 eth 4	N 10 eth 5	N 10 eth 3	N 8 eth 0	N 8 eth 4	N 8 eth 5	N 6 eth 3	N 6 eth 5	N 6 eth 0	N 4 eth 2	N 4 eth 0	N 4 eth 3	N 3 eth 0	N 3 eth 2

Switch 2

Uplink to Switch 1				Node 13			Node 11			Node 9			Node 7			Node 5			Node 2			Node 1 (HN 1)	
P 1	P 3	P 5	P 7	P 9	P 11	P 13	P 15	P 17	P 19	P 21	P 23	P 25	P 27	P 29	P 31	P 33	P 35	P 37	P 39	P 41	P 43	P 45	P 47
S 1 Port 1	S 1 Port 3	S 1 Port 5	S 1 Port 7	N 13 eth 0	N 13 eth 4	N 13 eth 2	N 11 eth 1	N 11 eth 2	N 11 eth 3	N 9 eth 5	N 9 eth 2	N 9 eth 1	N 7 eth 0	N 7 eth 4	N 7 eth 1	N 5 eth 0	N 5 eth 4	N 5 eth 2	N 2 eth 2	N 2 eth 3	N 2 eth 4	N 1 eth 0	N 1 eth 1
P 2	P 4	P 6	P 8	P 10	P 12	P 14	P 16	P 18	P 20	P 22	P 24	P 26	P 28	P 30	P 32	P 34	P 36	P 38	P 40	P 42	P 44	P 46	P 48
S 1 Port 2	S 1 Port 4	S 1 Port 6	S 1 Port 8	N 13 eth 5	N 13 eth 3	N 13 eth 1	N 11 eth 0	N 11 eth 5	N 11 eth 4	N 9 eth 3	N 9 eth 0	N 9 eth 4	N 7 eth 3	N 7 eth 2	N 7 eth 5	N 5 eth 3	N 5 eth 1	N 5 eth 5	N 2 eth 5	N 2 eth 0	N 2 eth 1	N 1 eth 3	N 1 eth 2

Legende:
P = Switch-Port
S = Switch
eth = Netzwerkkarten-Port (Linuxbeschreibung)
N = Knoten
Rote Gruppierung = Trunking der Ports

Anhang 1: Aufteilung der Nodes auf die zwei Switches