Seminar Hochleistungsrechner: Aktuelle Trends und Entwicklungen
Wintersemester 2016/2017

Verbindungsstrukturen

Moritz Dötterl
Technische Universität München

1.2.2017

Zusammenfassung

In dieser Seminararbeit werden Verbindungsstrukturen in Hochleistungsrechnern behandelt. Es werden verschiedene Ebenen betrachtet, auf denen unterschiedliche Systeme miteinander verbunden werden. Für die verschiedenen Ebenen werden aktuelle Systeme vorgestellt und verglichen. Der Leser soll so ein tieferes Verständnis für aktuelle Verbindungsstrukturen in Hochleistungsrechnern erhalten.

1 Einleitung

Der aktuell schnellste Hochleistungsrechner ist der Sunway TaihuLight in China. Er hat eine theoretische Spitzenleistung von 125 PetaFLOPS und ist damit weit vor dem zweitschnellsten Hochleistungsrechner, der eine theoretische Spitzenleistung von 55 PetaFLOPS erreicht (Stand 11.2016)[5]. In der Zukunft sollen solche Systeme noch leistungsfähiger werden. Firmen versuchen die Ersten zu sein, die es schaffen, ein System zu bauen, das die magische ExaFLOPS Marke überschreitet. Solche Systeme werden als Exascalesysteme bezeichnet. Damit solche Systeme möglich werden, müssen die aktuell verwendeten Technologien weiterentwickelt werden. Dazu gehören auch die Verbindungsstrukturen zwischen den Recheneinheiten.
Diese tragen dabei nicht direkt zur Leistung eines Prozessors bei, so können sie die Performance eines Systems nicht aktiv steigern. Sehr wohl können sie aber die Systemleistung sehr stark negativ beeinflussen und somit gute Prozessoren regelrecht ausbremsen. Deshalb muss sichergestellt werden, dass eine Verbindungsstruktur gut genug funktioniert. Um nun Exascale Performance erreichen zu können, werden sehr viele leistungsstarke Prozessoren parallel benötigt, die optimal miteinander verbunden sind.
In dieser Arbeit werden verschiedene Verbindungsstrukturen auf verschiedenen Ebenen vorgestellt. Zunächst werden mit HyperTransport und QuickPath zwei aktuell genutzte CPU-CPU Verbindungsstrukturen betrachtet und verglichen. Im nächsten Abschnitt wird eine neue Technologie von Nvidia zur Anbindung von Grafikkarten untereinander und zur CPU betrachtet: NVLink. Im letzten Teil der Arbeit wird EXTOLL, eine Verbindungsstruktur zwischen verschiedenen Knoten betrachtet.

2 CPU-CPU Verbindung

In klassischen Hochleistungsrechnern werden für maximale Rechenleistung viele Prozessoren eingesetzt. So ist es keine Seltenheit, dass zwei oder mehr Prozessoren sich ein Mainboard teilen und direkt miteinander zusammenarbeiten. Im Folgenden werden zwei Technologien vorgestellt, die seit ein paar Jahren benutzt werden, um mehrere Prozessoren auf einem Mainboard miteinander zu verbinden.

2.1 HyperTransport

Auf der offiziellen Webseite zu HyperTransport heißt es übersetzt, HyperTransport sei eine hochmoderne, paketbasierte, skalierbare Punkt-zu-Punkt Verbindungsstruktur mit großer Bandbreite und geringer Latenz [7]. Vorgestellt wurde HyperTransport ursprünglich 2001 und seitdem bis 2008 zur Version 3.1 weiterentwickelt. Entworfen wurde es vom HyperTransport-Konsortium, einer non-Profit Organisation, gegründet von vielen bekannten Firmen wie zum Beispiel AMD, Nvidia, Broadcom und Apple[10].

2.1.1 Anwendungsbereich

Angewendet wird HyperTransport besonders in AMD-basierten Systemen zur CPU-CPU Verbindung, sowie zum Chipsatz. Des Weiteren wurde HyperTransport auch in Apples Power Mac G5[17] und einigen MIPS Systemen, wie dem PMC-Sierra RM9000X2 eingesetzt. Somit steht HyperTransport in Konkurrenz zu Intels QuickPath Technologie, die später erläutert wird, und besonders zu PCI Express. Wobei QuickPath im Grunde als indirekter Konkurrent zu betrachten ist, da HyperTransport und QuickPath nur mit unterschiedlichen Prozessoren kompatibel sind. Des Weiteren kann HyperTransport, genau wie PCI Express, auch eingesetzt werden um externe Peripherie sowie E/A Controller anzusteuern[17]. Dafür verwendet HyperTransport sogar denselben Verbinder wie PCI Express. Da allerdings die Pinbelegung anders ist wird der HyperTransport-Verbinder um 180 Grad gedreht verbaut.

2.1.2 Funktionsweise

HyperTransport besteht aus zwei einzelnen unidirektionalen Punkt-zu-Punkt Verbindungen, die zusammen eine bidirektionale Verbindung ermöglichen. Durch Weiterleitungen können so auch Netze über mehrere Geräte, hier Links genant, gebaut werden. Jede Verbindung besteht aus 2, 4, 8, 16 oder 32 Datenleitungen pro Richtung, wobei beide Richtungen unterschiedliche Breiten besitzen dürfen [13]. Zusätzlich zu den Datenleitungen besitzt HyperTransport für jedes Byte, also pro acht Leitungen, eine eigene Taktleitung sowie, seit Generation 3.0, eine Leitung die angibt, ob es sich um ein Kontrollsignal handelt oder nicht. Im Vergleich dazu gab es bei älteren Versionen dafür lediglich eine Leitung pro Link. Des Weiteren gibt es pro Link eine Leitung, die angibt, ob die Stromversorgung und das Taktsignal stabil sind, sowie eine Resetleitung.
Bei x86-basierten Systemen gibt es zusätzlich die Möglichkeit, einen Link abzuschalten während einer Taktfrequenzänderung, sowie ein weiteres Signal um anzuzeigen, ob ein Link aktiv ist. HyperTransport arbeitet mit verschiedenen Taktraten, die sich je nach Generation und Endgerät unterscheiden und im Bereich von 200 MHz bis 3200 MHz liegen [13]. Daten werden im DDR-Verfahren mit differentiellen Signalpaaren bei einer Spannung von 1,2 V ±5% übertragen. DDR steht für Double Data Rate, dabei werden die zu übertragenden Bits nicht nur bei einer steigenden Flanke des Taktes, sondern sowohl bei steigender als auch bei fallender Flanke auf den Bus gelegt. So wird eine höhere Datenrate erzielt. HyperTransport ist prinzipiell nicht Cache-Kohärent, jedoch hat AMD die proprietären Erweiterungen Kohärent HyperTransport und HyperTransport Assist entwickelt, auf die hier nicht weiter eingegangen werden soll[15].

2.1.3 Features und Vergleich mit anderen Systemen

Im Vergleich zu PCI Express hat HyperTransport eine geringere Latenz und bietet dadurch eine schnellere Übertragung. Grund dafür sind eine effizientere Anbindung an den Prozessor sowie weniger Overhead der einzelnen Pakete. Abbildung 1 zeigt einen Vergleich zwischen dem benötigten Overhead bei HyperTransport und PCI Express. Es ist klar zu sehen, dass HyperTransport pro Paket weniger Overhead benötigt als PCI Express, allerdings kommen diese Informationen vom HyperTransport Konsortium selber und sind deshalb mit Vorsicht zu genießen: Es muss beachtet werden, dass PCI Express mit einem Paket bis zu 4096 Byte Daten

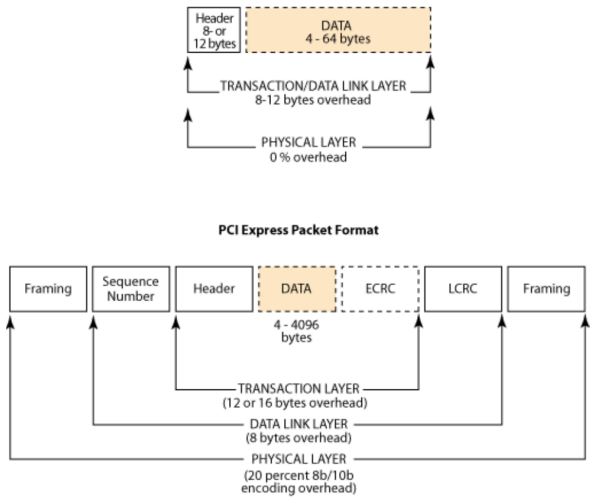

Abbildung 1: HyperTransport Paket Format im Vergleich zum PCI Express Paket Format [12]

übertragen kann, wobei HyperTransport lediglich maximal 64 Byte Daten in einem Paket übertragen kann. Dies resultiert daraus, dass eine Cache line, typischerweise 64 Byte groß, in einem Paket übertragen werden soll. Selbst bei maximaler Paketgröße hat HyperTransport somit $\frac{12}{64} = 18,75\%$ Overhead. Das ist nicht besonders viel weniger als bei PCI Express $\frac{0,2*(4096+8+16)+8+16}{4096} = 20,70\%$. Viel wichtiger dabei ist, dass der Overhead bei kleineren Paketen wesentlich geringer ist als bei PCI Express. Das liegt vor allem daran, dass bei HyperTransport keine spezielle Codierung der Physikalischen Ebene benutzt wird. Bei PCI Express hingegen wandelt diese 8 Bit Nutzdaten in 10 Bit codierte Daten um und erzeugt somit schon einen unumgänglichen Overhead von 20%. Natürlich ist die Leistungsfähigkeit eines Bus-systems nicht nur durch den Overhead der Pakete beschränkt, doch gibt dies einen guten Einblick, wie viele Bits zusätzlich zu den eigentlichen Daten mit übertragen werden müssen und damit die Datenübertragung bremsen. Des Weiteren zeichnet sich HyperTransport besonders durch seine geringe Latenz und hohe Übertragungsrate in einem kostengünstigen System aus[12].

2.2 Intel QuickPath

QickPath ist eine von Intel entwickelte Punkt-zu-Punkt basierte Verbindungsstruktur zur Kommunikation verschiedener Prozessoren untereinander und zum Chipsatz. Es deckt somit im Grunde dieselben Anforderungen ab wie zum Beispiel HyperTransport oder PCI Express, ist im Gegensatz dazu allerdings ein Routing-Mechanismus, der verschiedene Prozessoren in einem Netzwerk verbindet. Dafür wird der kürzeste Weg zwischen zwei Prozessoren ermittelt, gegebenenfalls führt dieser über weitere Prozessoren. QuickPath wurde erstmals 2008 von Intel vorgestellt und ersetzte den bis dahin benutzten Front Side Bus. QickPath soll demnächst durch eine neuere Technologie ersetzt werden, über die allerdings, außer dem Namen UltraPath, noch nichts bekannt ist (Stand Dezember 2016).

2.2.1 Funktionsweise

Genau wie bei HyperTransport auch besteht ein bidirektionaler Port bei QuickPath aus zwei einzelnen unidirektionalen Links. Ein Prozessor kann dabei ein oder mehrere Ports haben. Abbildung 2 zeigt ein Setup in dem vier einzelne Prozessoren mit je vier QuickPath Ports miteinander und mit dem Chipsatz verbunden sind. Es ist auch gezeigt, dass ein bidirektionaler QuickPath Port aus zwei unidirektionalen Links besteht. Zusätzlich dazu besitzen die Prozessoren weitere Busse, über die sie mit dem Speicher verbunden sind. Genauso besitzt der Chipsatz weitere nicht näher spezifizierte Busse zur Verbindung zu Ein/Ausgabe Geräten.
QickPath benutzt 20 differentielle Signalleitungen und zusätzlich eine Taktleitung. Dabei werden immer nur 16 Bits zum gleichzeitigen Datentransfer genutzt, die restlichen 4 Bits werden für den Overhead genutzt. Im Gegensatz zu HyperTransport steht bei QuickPath die Breite des Busses fest und ist nicht flexibel. Gleiches gilt für die Taktfrequenz, diese war ursprünglich 2,4 oder 3,2 GHz, und wurde in späteren Versionen auf bis zu 4,8 GHz erhöht. Des Weiteren ist QuickPath in

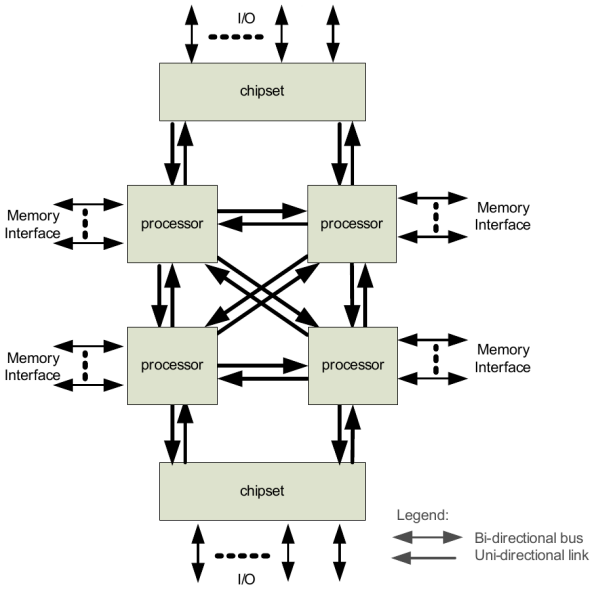

Abbildung 2: QuickPath Architektur mit 4 Prozessoren und je 4 Links pro Prozessor [14]

verschiedene Ebenen unterteilt. Diese sind, von der Hardware zur Abstraktesten: Physikalische Schicht, Link Schicht, Routing Schicht, Transport Schicht und Protokoll Schicht.[14]
Die Physikalische Schicht besteht aus der eigentlichen Hardware, also den Leitungen und Logikbausteinen, die die eigentliche Übertragung vornehmen.
Die nächsthöhere Schicht ist die Link Schicht, die für eine verlässliche Übertragung und den Kontrollfluss zuständig ist.

Eine Abstraktionsebene höher befindet sich die Routing Schicht, die ein Gerüst bietet, mit dessen Hilfe Pakete effizient durch das Netzwerk geleitet werden können.
Die Transport Schicht war anfangs nicht vorhanden und wurde für später reserviert, um erweitertes Routing und verlässliche End-zu-End Kommunikation zu ermöglichen. Die Schicht ist optional und besonders in in großen Systemen wichtig. Außerdem bietet sie erweiterte Funktionen die es zum Beispiel ermöglichen andere als Intels QuickPath Protokolle zu benutzen.[21]
Die abstrakteste Schicht bildet die Protokoll Schicht. Sie besteht aus einigen Regeln zum Austausch von Paketen.

2.2.2 Features und Vergleich mit anderen Systemen

In Intels Whitepaper "An Introduction to the Intel® QuickPath Interconnect"[14] wird ebenfalls der Overhead von QuickPath mit dem von PCI Express verglichen. Da QuickPath für Interprozessorkomunikation entworfen wurde, ist die maximale Datenmenge pro Paket auf die Länge einer Cache line festgelegt worden, also 64 Byte. Bei maximaler Paket-länge kommt QuickPath mit lediglich acht zusätzlichen Bytes also $\frac{8}{64} = 12,5\%$ Overhead aus. Die Auswirkung dieses Overheads kann zusätzlich noch reduziert werden, indem Teile der Steuerdaten parallel mit den Daten selber geschickt werden können. Im Vergleich dazu liegt, wie vorher schon erwähnt, der Overhead von HyperTransport bei 18,75% und von PCI Express im besten Fall bei 20,70%. Wenn über PCI Express nun eine Cache line, also 64 Byte, übertragen werden, liegt der Overhead sogar bei 32%. In Tabelle 1 wird der benötigte Overhead der drei Systeme nochmals übersichtlich zusammengefasst. Sie zeigt besonders, dass der Overhead für kleine Pakete bei PCI Express enorm hoch wird und PCI Express deshalb besser für Ein/Ausgabe-Geräte geeignet ist als um Cache Lines auszutauschen.
QuickPath ist im Gegensatz zu HyperTransport Cache kohärent und sorgt somit dafür, dass alle übertragenen Speicherbereiche und Caches stets konsistente Daten enthalten. Erreicht wird dies durch source oder home snooping in Verbindung mit Cache-zu-Cache Transaktionen[15]. Im Gegensatz zu PCI Express und HyperTransport ist QuickPath nicht als Schnittstelle für externe Peripherie konstruiert.
Um das Routing auf dem Mainboard zu erleichtern, ist es bei QuickPath möglich, einzelne differentielle Signalpaare zu tauschen, oder sogar den gesamten Link umzudrehen. So kann Pin 0 bis 19 auf Pin 19

Tabelle 1: Vergleich des Overhead von QuickPath, HyperTransport und PCI Express für verschiedene Paketlängen.

	QPI	HT	PCIe
64 Byte			
Anzahl Pakete	1	1	1
Overhead in Byte	8	12	30
Overhead	12,5%	18,75%	46,88%
4096 Byte			
Anzahl Pakete	64	64	1
Overhead in Byte	512	768	848
Overhead	12,5%	18,75%	20,70%

bis 0 verbunden werden. Dadurch müssen, je nach Mainboard, die Leitungen nicht überkreuzt werden und dem Design des Mainboards wird mehr Freiraum gegeben. Während der Initialisierung lernt die Physikalische Ebene dann automatisch, welche Leitungen getauscht wurden[14].

3 GPU-GPU Verbindung

In modernen Supercomputern werden längst nicht mehr nur Prozessoren zur Berechnung eingesetzt, es wird vermehrt auf den Einsatz von Beschleunigern gesetzt. Die dabei am Häufigsten verwendete Art von Beschleuniger ist eine Grafikkarte. Durch ihre höhere Parallelität lässt sich insgesamt eine höhere Rechenleistung erzielen und gleichzeitig durch die weniger komplexen Rechenkerne der Energieverbrauch drastisch reduzieren. 2012 wurde Titan vorgestellt, der damals rechenstärkste Supercomputer weltweit. Mit einer theoretischen Spitzenleistung von 27 PetaFLOPS befindet sich Titan auch heute noch auf dem 3. Platz der top500 Liste (Stand 11.2016)[5]. Dieser besitzt insgesamt 18.688 GPUs, pro Prozessor eine Grafikkarte, und generiert so 90% seiner Peak Performance allein durch seine Grafikkarten [1][19]. Nachdem dieser Trend anhält und neuere Systeme teilweise zwei oder mehr Grafikkarten pro Prozessor besitzen, müssen nun neue Verbindungsstrukturen entwickelt werden, die dieser Belastung gewachsen sind. Die alten Strukturen waren vor allem darauf ausgelegt, dass die Grafikkarte benutzt wurde, um den Bildschirminhalt zu rendern. Wenn die Grafikkarte nun eingesetzt wird, um gemeinsam mit dem Prozessor Berechnungen durchzuführen, müssen GPU und CPU offensichtlich weit mehr kommunizieren und somit weit besser angebunden sein. Eines dieser Systeme, das für solch eine Aufgabe entwickelt wurde ist NVLink, das im Folgenden vorgestellt wird.

3.1 Nvidia NVLink

NVLink wurde von Nvidia entwickelt, um Grafikkarten besser miteinander und mit dem Prozessor zu verbinden. Nvidia behauptet dabei, dass NVLink ein hoch effizientes System mit hoher Bandbreite und Datenraten von mindestens 80 Gigabyte pro Sekunde ist. Das wäre mindestens 5 mal schneller als der Hauptkonkurrent: Ein PCI Express Gen3 x16 Bus[19]. Auch wenn NVLink erst 2016 vorgestellt wurde, soll im laufe des Jahres 2017 bereits eine weiterentwickelte Form vorgestellt werden: NVLink 2.0[4]. Dabei kann NVLink den PCI Express Bus in einem System komplett ersetzen, es ist aber auch möglich, dass PCI Express gemeinsam mit NVLink eingesetzt wird.

3.1.1 Funktionsweise

Da die Technologie ziemlich neu ist - erste Grafikkarten, die die Technologie unterstützen sind dieses Jahr vorgestellt worden - sind genauere Informationen dazu, wie NVLink aufgebaut ist, nicht zu finden.
NVLink basiert auf einem High-Speed Bus, der aus 8 Datenleitungen besteht. Diese arbeiten wie bei HyperTransport oder QuickPath auch mit Differentiellen Signalen in einem dual Simplex bidirektionalen Link. Die aktuellen Pascal GPUs unterstützen mehrere solcher Links. Die Grafikkarte kann dann entweder mit mehreren Links mit einer anderen Grafikkarte verbunden werden, oder mit einem Link zum Prozessor und mit den restlichen Links zu anderen Grafikkarten. Jeder einzelne Link bietet eine Übertragungsrate von 20 GB/s. Werden nun meh-

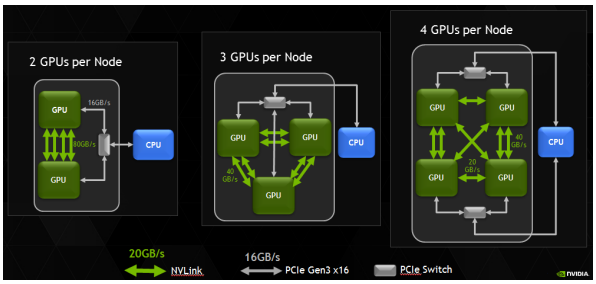

Abbildung 3: NVLink Architektur mit zwei, drei und vier GPUs mit je vier Links pro GPU [2]

rere Links zwischen zwei Endgeräten verwendet, addieren sich die einzelnen Übertragungsraten. So können mit vier Links bis zu 80 GB/s übertragen werden. Abbildung 3 zeigt verschiedene Konfigurationsmöglichkeiten, wie zwei, drei oder gar vier Grafikkarten über NVLink miteinander verbunden werden. Die Anbindung zum Prozessor wird hier weiterhin durch PCI Express realisiert. Allgemein betrachtet Nvidia in ihren Whitepapers bisher nur Konfigurationen, in denen lediglich die GPUs untereinander mit NVLink verbunden sind und die Anbindung zur CPU weiterhin über PCI Express läuft. Wahrscheinlich liegt das an der noch fehlenden Unterstützung von NVLink in aktuellen Prozessoren. Aktuell (stand Januar 2017) wird NVlink lediglich von IBMs POWER Architektur unterstützt, ob andere Prozessoren folgen werden bleibt ungewiss. Somit wird in den Benchmarks noch nicht das gesamte Potential durch NVLink ausgeschöpft. [3]

3.1.2 Features

NVLink bietet durch seine höhere Datenrate auch weitere Vorteile. So kann zum Beispiel ein quasi gemeinsamer Speicher zwischen Prozessor und Grafikkarte erreicht werden. Mit diesem ist es möglich, dass ein Programm, das auf der GPU läuft, mit beliebigen Pointern arbeitet, ohne zu wissen, wo genau sich diese Daten befinden. Die Hardware findet dann den Speicherort heraus und lädt bei Bedarf den Wert aus dem Speicher der CPU. Bei aktuellen Hochleistungssystemen ist solch etwas noch nicht möglich, es müssen die Daten manuell in den Speicher der GPU übertragen werden und diese kann nur auf Daten, die in ihrem Speicher liegen, zugreifen. Da aber auch bei NVLink GPU und CPU getrennte Speichercontroller besitzen, können diese auf die jeweiligen Anforderungen optimiert werden, für die GPU auf Bandbreite und für die CPU auf Latenz.
NVLink ist nicht Cache-Kohärent, dieses Feature soll mit NVLink 2.0 und der Volta GPU Architektur im laufe des Jahres 2017 eingeführt werden[4]. Des Weiteren braucht NVLink dabei bis zu drei mal weniger Energie als PCI Express Gen3 x16.
NVLink kann zusammen mit x86, ARM und POWER Prozessoren genutzt werden. Für Entwickler gibt es drei Möglichkeiten, die GPU über NVLink anzusteuern: GPU-beschleunigte Bibliotheken, OpenACC Compiler Directives und CUDA[2].

3.1.3 Einsatz in realen Systemen

NVLink wird für Privatkunden ab den aktuellen Nvidia Pascal Grafikkarten, seit 2016, unterstützt. Nvidia bietet zusätzlich den DGX-1, einen NVLink basierten Server, für Firmen an.
Abgesehen davon werden aktuell bereits neue Pre-Exascale Hochleistungsrechner mit NVLink entwickelt. Im Folgenden werden zwei Supercomputer, sowie das DGX-1 System vorgestellt.

Summit und Sierra Summit und Sierra sind zwei neue Hochleistungsrechner des U.S. Department of Energy. Summit wird am Oak Ridge National Laboratory und Sierra am Lawrence Livermore National Laboratory eingesetzt werden. Summit wird damit ab 2017 den bekannten Titan Hochleistungsrechner ersetzen. Sierra wird das Hauptsystem zum Management und zur Sicherheit der U.S. Kernwaffen und zur Überwachung des Kernwaffensperrvertrages. Außerdem wird er für Antiterrorismus Programme eingesetzt werden. Beide Systeme setzen dabei auf IBM POWER9 Prozessoren und Nvidia Volta Grafikkarten, die über NVLink miteinander verbunden sind. Auf diesem Modell und mit den Erfahrungen sollen später die ersten Exascale Hochleistungsrechner gebaut werden. Summit wird voraussichtlich zwischen 150 und 300 PetaFLOPS

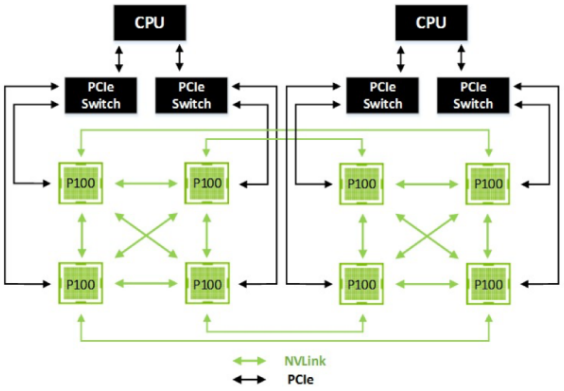

Abbildung 4: Architektur von DGX-1 [8]

Peak Leistung haben, während Sierra mehr als 100 PetaFLOPS leisten soll. Dafür wird Summit mehr als 3400 Knoten mit je mehreren CPUs und mehreren GPUs besitzen, womit jeder knoten auf mehr als 40 TerraFLOPS kommen soll. Dank NVLink besitzt jeder knoten 512 GB kohärenten Speicher, der sowohl von den CPUs als auch von den GPUs benutzt werden kann. Zusätzlich können die GPUs noch auf 800 GB NVRAM zurückgreifen. Nvidia behauptet, dass Summit und Sierra zwei wichtige Zwischenschritte auf dem Weg zu Exascale Hochleistungsrechnern sind.
Laut einer aktuellen Information der Webseite der Top500 Liste plant das U.S. Department of Energy, den ersten Exascale Hochleistungsrechner, basierend auf den Erfahrungen von Summit und Sierra, bis 2021 zu bauen [6].

DGX-1 Ein weiteres System, dass für Firmen, die sich keinen Hochleistungsrechner leisten können oder wollen, interessanter ist, ist Nvidias DGX-1. Es ist besonders für Deep Learning und künstliche Intelligenz entwickelt worden und bereits heute für ca. 150.000 $ erhältlich [8]. DGX-1 basiert auf zwei Intel Xeon E5-2698v4 mit jeweils 20 Kernen, die von acht Tesla P100 GPUs unterstützt werden. Als Verbindungsstruktur kommt NVLink zum Einsatz. Wie in Abbildung 4 zu sehen, wird auch in diesem System lediglich die Verbindung der GPUs untereinander mit NVLink realisiert, die Anbindung an die CPUs erfolgt weiterhin über PCI Express. DGX-1 soll die Trainingsdauer für künstliche Intelligenz, z.B. Neurale Netzwerke, drastisch reduzieren und die Performance wesentlich steigern. Nvidia behauptet, dass DGX-1 75 mal schnelleres Training ermöglichen würde. Jedoch ist dieser Vergleich nicht besonders aussagekräftig, da er zu einem Server mit zwei CPUs einer älteren Generation (Xeon E5-2697v3) ohne GPU Unterstützung gemessen wurde. [11]

3.1.4 Performance-Vergleich zu anderen Systemen

Um die Performance von NVLink abschätzen zu können, hat Nvidia ein Testsystem aufgebaut. Dieses verwendet nicht näher spezifizierte Grafikkarten der nächsten Generation und vergleicht die Performance, wenn die Grafikkarten rein über PCI Express angeschlossen sind gegen ein Setup, bei dem die Grafikkarten untereinander mit NVLink verbunden sind und die Anbindung zur CPU weiterhin über PCI Express gestaltet wird. Da NVLink gerade erst auf den Markt gekommen ist, gibt es leider noch keine unabhängigen Benchmark Tests und es bleibt nichts anderes übrig, als den Nvidia Zahlen zu vertrauen.
Die Performance unterscheidet sich natürlich je nach Anwendung. Berechnungsintensive Anwendungen profitieren weniger von NVLink als Anwendungen, die viel Austausch zwischen den GPUs erfordern. Abbildung 5 zeigt den Speedup, der sich durch einen Wechsel von PCI Express zu NVLink für verschiedene Anwendungen ergibt. 3D FFT und ANSYS wurden dabei in einer Konfiguration mit zwei Grafikkarten getestet, die anderen Beispiele mit vier Grafikkarten. Es ist deutlich zu sehen, dass eine 3D FFT, die viel Kommunikation erfordert, mit einem Speedup von fast 2,25 sehr stark von NVLink profitiert, wohingegen Anwendungen wie ANSYS mit einem Speedup von 1,25 weniger stark profitieren. Eine Verschlechterung ist jedoch nirgendwo festzustellen. Da die Daten allerdings von Nvidia stammen, ist das auch nicht zu erwarten. Für die meisten Anwendungen scheint ein Speedup von

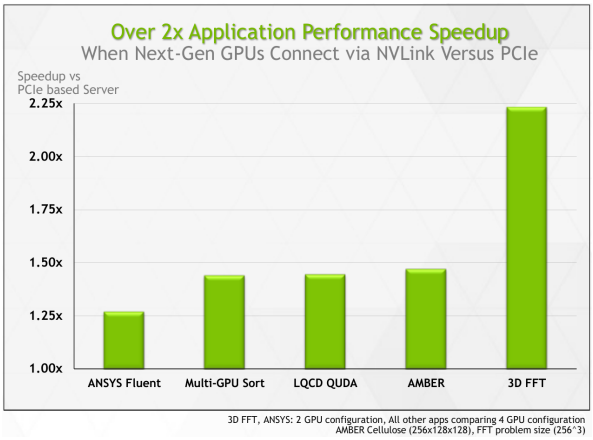

Abbildung 5: Performance eines NVLink Systems verglichen mit einem identischen PCI Express basierten System [20]

knapp 1,5 realistisch zu sein. Ob sich das in der Praxis auch so bemerkbar macht, muss sich erst noch zeigen.

4 Node-Node Verbindung

Bisher wurden Verbindungsstrukturen zwischen mehreren CPUs, mehreren GPUs und von CPUs zu GPUs betrachtet. Diese Strukturen werden innerhalb eines Knotens verwendet. Um allerdings die Rechenleistung eines aktuellen Hochleistungsrechners erreichen zu können, werden mehrere tausend Knoten benötigt. Um die Leistung der Hochleistungsrechner weiter zu steigern, wird in neuen Designs gerne die Anzahl an Knoten noch weiter erhöht. Um diese enorme, stetig wachsende Anzahl an Knoten miteinander zu verbinden, werden Netzwerke benötigt, die gut skalieren und einen schnellen Datenaustausch gewährleisten können. Ein solches neues Netzwerk ist EXTOLL, das im Folgenden vorgestellt wird.

4.1 EXTOLL

EXTOLL geht aus einem Forschungsprojekt der Universität Heidelberg hervor, das 2005 begonnen wurde. 2011 wurde die EXTOLL GmbH gegründet, die seitdem die Technik weiterentwickelt. EXTOLL ist ein neues Netzwerk auf Basis von FPGAs, das besonders für kleine Pakete optimiert ist. Es wurde speziell zur Vernetzung von Knoten in Hochleistungsrechnern entwickelt. Eines der wichtigsten Augenmerke während der Entwicklung war eine möglichst geringe Latenz sowie eine gute Skalierbarkeit, weshalb einige Konzepte, die bei klassischen Netzwerken durch das Betriebssystem geregelt werden, bei EXTOLL direkt in Hardware implementiert sind. Somit wird die CPU entlastet und gleichzeitig die Latenz reduziert. So benutzt die Schnittstelle zum Beispiel Hardwarevirtualisierung und kann somit von bis zu 64.000 Threads gleichzeitig benutzt werden, ohne dass das Betriebssystem diese Zugriffe regeln muss [18].

4.1.1 Funktionsweise

Die EXTOLL Architektur ist auf einem FPGA oder ASIC implementiert und wird als Erweiterungskarte, ähnlich wie eine Grafikkarte, über PCI Express mit dem Mainboard verbunden. Die Karten unterschiedlicher Knoten sind untereinander über Optische Kabel verbunden. Abbildung 6 zeigt ein Blockdiagramm der EXTOLL Architektur. Dort ist auf der linken Seite der PCI Express Bus zu sehen und auf der rechten Seite die Verbindung ins Netzwerk. Über ein On-Chip-Netzwerk sind mehrere Einheiten mit dem PCI Express Bus verbunden. VELO ist die Kommunikations-Engine, die besonders für kurze Pakete optimiert ist. Sie besteht aus einem Requester, der Pakete verschickt und einem Completer der Pakete empfängt und in den Hauptspeicher der CPU schreibt. Für längere Pakete wird die Remote Memory Access (RMA) Engine verwendet. Diese entlastet die CPU mehr und verwendet das Put/Get Model. Unterstützt wird sie von der Address Translation Unit (ATU), die virtuelle Speicheradressen in Physikalische Adressen umrechnet. Bei jeder Übertragung entscheidet entweder die verwendete Bibliothek, oder der Programmierer selbst, welche Engine benutzt werden soll[22].
Mit dieser effizienten, gepipelineten Hardware bietet EXTOLL einen schnellen Nachrichtenaustausch

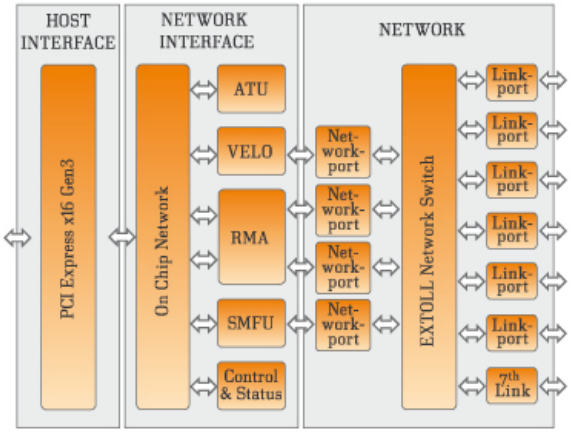

Abbildung 6: Blockdiagramm der EXTOLL Architektur [9]

mit geringer Latenz und hoher Nachrichtenrate sowohl für lange als auch kurze Pakete.

4.1.2 Features

EXTOLL bietet ein flexibles Design, das bis zu 64.000 Knoten in einer beliebigen direkten Topologie unterstützt. Dabei wird garantiert, dass Pakete immer in der Reihenfolge geliefert werden, in der sie abgeschickt wurden. EXTOLL verfügt außerdem über drei virtuelle Chanel (VCs) und vier unabhängige Übertragungsklassen.
Dank der VELO-Engine sind sehr schnelle Nachrichtenaustausche möglich, zugleich ist der Zugriff auf Speicher anderer Knoten dank der RMA-Engine ebenfalls optimiert. Durch die integrierte Hardwarevirtualisierung können verschiedene Prozesse, oder sogar virtuelle Maschinen, die beiden Kommunikations-Engines verwenden, ohne dass das Betriebssystem den Zugriff auf die Hardware manuell regeln muss.
EXTOLL bietet durch die Hardwareimplementierung zusätzlich auch noch Sicherheitsfeatures. So werden die Kommunikationsgruppen durch die Hardware separat gehalten und die Prozesse eines Knotens bleiben getrennt. Alle internen Speicher sind durch ECC abgesichert, und die Richtigkeit der Übertragungen wird durch CRC Checks sichergestellt.
Um EXTOLL ansteuern zu können gibt es Treiber für den Linux Kernel (ab 2.6), sowie eine Userspace-Bibliothek. Des Weiteren wird das Message Passing Interface OpenMPI und das Network Management Protokoll EMP unterstützt. Der Kernel Treiber wird dabei zur Einrichtung der EXTOLL Hardware benutzt, danach können Nachrichten auch direkt aus dem Userspace über die entsprechenden Bibliotheken versendet werden[22]. Dadurch werden Kontextwechsel in den Kernel vermieden und die Performance weiter gesteigert. Der größte Vorteil ist jedoch, dass das Design sehr flexibel ist und mit relativ geringen Aufwand an die jeweiligen Bedürfnisse angepasst werden kann.

4.1.3 Einsatz in realen Systemen

EXTOLL ist momentan auf zwei Erweiterungskarten erhältlich, Galibier und Tourmalet, die im Folgenden kurz vorgestellt werden.

Galibier Galibier ist die erste Implementierung von EXTOLL, die 2011 vorgestellt wurde. Die Karte basiert auf einem Xilinx FPGA und unterstützt bis zu 4 EXTOLL Links mit je 16 GB/s pro Richtung, aus denen eine beliebige Topologie aufgebaut wird. Zur Anbindung an das Mainboard wird ein PCI Express x8 v2.0 benutzt

Tourmalet Tourmalet ist die erste ASIC Implementation von EXTOLL und seit Q3 2015 erhältlich. Im Gegensatz zu Galibier wird hier ein PCI Express x16 v3.0 verwendet. Außerdem verfügt Tourmalet über 6 Links, um damit das Netzwerk aufzubauen und einen weiteren Link, um einen Beschleuniger anzusteuern. Jeder Link erreicht Datenraten von bis zu 100 GB/s pro Richtung [16] Dadurch dass ein ASIC schneller ist als ein FPGA, erreicht Tourmalet eine typische Latenz von 600-800 ns und bis zu 120 Millionen MPI Nachrichten pro Sekunde. [9]

5 Schluss

Es wurden verschiedene Verbindungsstrukturen vorgestellt, die auf unterschiedlichen Ebenen arbeiten, um gemeinsam dafür zu sorgen, dass Hochleistungsrechner überhaupt möglich sind. Zwischen mehreren CPUs wurden HyperTransport und QuickPath vorgestellt, die beide schon etwas älter sind und wahrscheinlich nicht mehr in einem Exascale System eingesetzt werden. Intel entwickelt gegenwärtig einen Nachfolger namens UltraPath, der dann höchstwahrscheinlich in einem Exascale System mit Intel Prozessoren eingesetzt wird. Bevor AMD eine neue Verbindungsstruktur entwickelt die potentiell in einem Exascale System eingesetzt wird müsste AMD zunächst erst einmal eine neue Prozessor Architektur vorstellen, die in Supercomputern verwendet werden kann. Des Weiteren wurde zur Verbindung unter Grafikkarten und von Prozessoren zu Grafikkarten NVLink vorgestellt. Diese Technik ist zum Zeitpunkt dieser Arbeit gerade erhältlich geworden und ich halte es für vorstellbar dass diese, oder eine weiterentwickelte Form davon, im ersten Exascale Hochleistungsrechner zu finden sein wird. Zuletzt wurde EXTOLL betrachtet, das verschiedene Knoten eines Hochleistungsrechners miteinander vernetzt. Laut Aussage der EXTOLL GmbH ist diese Technik bereit für Exascale Systeme. Die Limitierung von maximal 64.000 Knoten ist dafür ausreichend. Nimmt man als Berechnungsgrundlage den zuvor vorgestellten Summit Supercomputer: Dieser soll über 3400 Knoten verfügen und zwischen 150 und 300 PetaFLOPS Rechenleistung erreichen. So erhält man eine pro Knoten Leistung zwischen $\frac{150}{3400} = 0,044$ und $\frac{150}{3400} = 0,088$ PetaFLOPS. Bei 64.000 Knoten würde man so auf eine theoretische Leistung zwischen 2,8 und 5,6 ExaFLOPS kommen. Da zu erwarten ist, dass die Knoten auch zunehmend leistungsstärker werden, sind Exascale Systeme mit EXTOLL durchaus wahrscheinlich. Ob diese Technik jedoch im ersten Exacale System auch eingesetzt wird, oder ob doch ein anderes Netzwerk verwendet wird, bleibt abzuwarten. Mit den hier vorgestellten Verbindungsstrukturen kommen wir Exascale Systemen einen großen Schritt näher, nun liegt es an den Prozessoren und Grafikkarten, die nötige Rechenleistung aufzubringen. Die Verbindungsstrukturen sind bereit für Exascale Systeme.

Literatur

[1] Introducing Titan, 2012. https://www.olcf.ornl.gov/titan/.

[2] How NVLink Will Enable Faster, Easier Multi-GPU Computing, 2014. https://devblogs.nvidia.com/parallelforall/how-nvlink-will-enable-faster-easier-multi-gpu-computing/.

[3] NVLink, Pascal and Stacked Memory: Feeding the Appetite for Big Data, 2014. https://devblogs.nvidia.com/parallelforall/nvlink-pascal-stacked-memory-feeding-appetite-big-data/.

[4] Optimizing Data-Centric IT Environments, 2015.

[5] Die Top500 Liste November 2016, 2016. https://www.top500.org/lists/2016/11/.

[6] First US Exascale Supercomputer Now On Track for 2021, 2016. https://www.top500.org/news/first-us-exascale-supercomputer-now-on-track-for-2021.

[7] HyperTransport Overview, 2016. http://www.hypertransport.org/ht-overview.

[8] NVIDIA DGX-1 Deep Learning System, 2016. https://www.microway.com/preconfiguredsystems/nvidia-dgx-1-deep-learning-system/.

[9] Technology Overview, 2016. http://www.extoll.de/technology.

[10] The HyperTransport Consortium, 2016. http://www.hypertransport.org/consortium-1.

[11] The World's First AI Supercomputer in a Box, 2016. http://www.nvidia.com/object/deep-learning-system.html.

[12] Mario Cavalli. The Future of High-Performance Computing: Direct Low Latency CPU-to-Subsystem Interconnect.

[13] HyperTransport Consortium. HyperTrasnport I/O Link Specification. 6 2010.

[14] Intel Corporation. An Introduction to the Intel QuickPath Interconnect. 1 2009.

[15] David A. Wood Daniel J. Sorin, Mark D. Hill. A Primer on Memory Consistency and Cache Coherence. 2011.

[16] EXTOLL. EXTOLL TOURMALET brochure. 2016.

[17] Steve Jobs. Apple WWDC 2003 Keynote - The Power Mac G5 introduction. `https://youtu.be/iwsn27J_tlo?t=10m37s https://youtu.be/YJlO37wN9-w`.

[18] Heiner Litz. VELO: A Novel Communication Engine for Ultra-low Latency Message Transfers. 9 2008.

[19] Nvidia. NVIDIA NVLink High-Speed Interconnect: Application Performance. 11 2014.

[20] Nvidia. Summit and Sierra Supercomputers: An Inside Look at the U.S. Department of Energy's New Pre-Exascale Systems. 11 2014.

[21] Robert J. Safranek, Michelle J. Moravan. QuickPath Interconnect: Rules of the Revolution. 11 2009.

[22] Khaled Benkrid Wim Vanderbauwhede. *High-Performance Computing Using FPGAs*. 2013.