Hochleistungsrechner: Aktuelle Trends und Entwicklungen
Wintersemester 2016/17
Manycore Architekturen

Johannes Offner
Technische Universität München

02.02.2017

Zusammenfassung

In dieser Seminararbeit werden Grundlagen der Manycore Architekturen behandelt. Dabei wird zunächst auf die Entwicklung von Mehrkernprozessoren eingegangen und anschließend wichtige Merkmale einer solchen Architektur, wie Verbindungsstrukturen, Bandbreite, Energieverbrauch oder die Organisation von Cache erläutert. Zudem werden Schwierigkeiten des Chipdesigns aufgrund physikalischer und theoretischer Grenzen, wie Dark Silicon, der immer kleiner werdenden Strukturgröße sowie Amdahls Gesetz aufgezeigt. Abgeschlossen wird die Arbeit, indem mit Intels Xeon Phi Knights Landing (KNL) Prozessor und TILE von Tilera zwei Beispiele gegeben werden.

1 Entwicklung von Mehrkernprozessoren

Im Jahre 1965 entdeckte Gordon Earle Moore den Zusammenhang, dass sich die Komplexität von Prozessoren, gemessen an der Anzahl ihrer Transistoren, bei gleichbleibender Chipfläche, in regelmäßigen Zeitabständen verdoppelt. Aus diesen Beobachtungen leitete er die Vorhersage ab, dass dies auch bei zukünftigen Prozessoren gelten werde. Diese Regel beweist ihre Gültigkeit bis heute, wird als Mooresches Gesetz bezeichnet und gilt als wichtige Leitlinie der Prozessorindustrie. Der Zeitraum für eine Verdopplung beträgt dabei erfahrungsgemäß zwischen 12 und 24 Monaten [18, 11]. In den 90er Jahren konnten Leistungssteigerungen neben der wachsenden Anzahl an Transistoren vor allem auch durch exponentiell wachsende Frequenzraten der Prozessoren erreicht werden. Dieser Trend sollte jedoch etwa im Jahre 2004 enden, als die Taktraten der neuesten Prozessoren die 4-GHz-Schwelle erreichten.[13, 5, 18]

Mit einer stark steigenden Taktfrequenz nimmt auch die Verlustleistung des Chips exponentiell zu. Durch die immer kleiner werdenden Transistoren erhöhen sich zudem die Leckströme an den Gate- und Source-Elektroden. Dies führt durch die verstärkte Verlustwärme zu einer abfallenden Effizienz und mehr Kühlaufwand [13]. Aus wirtschaftlicher Sicht ist dies nicht tragbar. Trotz der anfänglichen Bemühungen durch stetig komplexere Prozessoren und niedrigere Betriebsspannungen dem entgegenzuwirken, war bald klar, dass man, um den Trend des exponentiellen Leistungszugewinns beizubehalten, künftig grundlegend umdenken musste [18, 9]. Dies bedeutete das beginnende Ende der Singlecore-Prozessoren und gleichzeitig den Anfang für eine schnell wachsende Entwicklung von Mehrkernprozessoren (CMPs). Anstatt die Frequenz eines Prozessors bei steigender Transistorzahl mit zu erhöhen, lässt sich die gleiche Rechenleistung mit weniger Verlustleistung wesentlich effizienter erreichen, indem man die Berechnungen des Programmcodes auf die im selben Chip befindlichen

Kerne aufteilt und Taktfrequenz sowie Betriebsspannung senkt.[18]
Heutzutage kommen vor allem im Bereich des HPC sog. Manycore-Prozessoren zum Einsatz. Ein Manycore-Prozessor zeichnet sich durch eine hohe Anzahl sowie gleichzeitig oft geringe Komplexität seiner einzelnen Kerne aus, was diesen somit auf paralleles Rechnen spezialisiert. Während Multicore-Prozessoren teils auch noch in sequenziellen Aufgaben ausreichend Leistung bieten, ist dies bei Manycore-Prozessoren meist nicht der Fall. [18, 10]

2 Multi-Core Grundlagen

Um im späteren Verlauf der Arbeit Architektur und Entwicklung der Manycore-Prozessoren besser zu verstehen, werden im Folgenden die wichtigsten Grundlagen und Begriffe eines CMP erläutert, indem dabei besonders auf die technischen und physikalischen Aspekte eingegangen wird.

2.1 Homogenität und Heterogenität

Mehrkernprozessoren sind homogen oder heterogen. Homogenität bedeutet, dass jeder der Kerne des Prozessors identisch ist. Im Gegensatz zur Heterogenität findet keine Spezialisierung auf bestimmte Aufgabenbereiche statt. Identische Aufgaben werden auf allen Kernen gleichermaßen bearbeitet. Das Betreiben der einzelnen Kerne mit unterschiedlichen Taktfrequenzen ist dabei aber möglich. Heterogene Manycore-Prozessoren weisen hingegen fundamentale Unterschiede in Funktionalität und Leistung ihrer Kerne auf. Zudem unterscheiden sich die Kerne heterogener Chips untereinander in ihrer Befehlssatzarchitektur. Dies macht paralleles Programmieren im Vergleich zu homogenen Prozessoren schwieriger. Ein herausragender Vorteil der Heterogenität ist jedoch, dass bei vielfältigen Programmen einige der Teilaufgaben von einem komplexen und andere von eher simplen Kernen profitieren. Um dies zu verdeutlichen sind in Abbildung 1 verschiedene Möglichkeiten des Designs eines Manycore-Prozessors dargestellt. [13, 18]

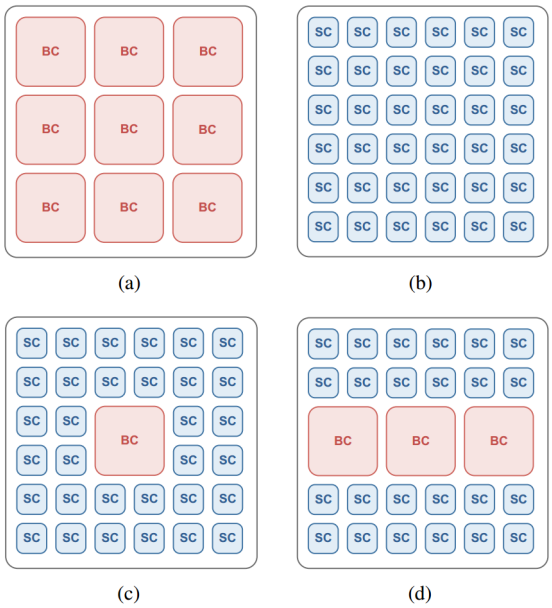

Abbildung 1: Homogenität und Heterogenität bei Manycore-Prozessoren [16]

(a) Ein homogener Manycore, wie er in dieser Abbildung gezeigt wird, besitzt wenige, aber sehr leistungsfähige große Kerne (BC). Ein solcher Prozessor bietet beste Rechenleistung für sequenzielle Aufgaben ohne dabei die Flexibilität zu verlieren zusätzlich parallel zu arbeiten. Dieser Ansatz wird beispielsweise von Firmen wie AMD und Intel verfolgt.[20, 16]

(b) Zu sehen ist ein homogener Prozessor aus vielen simplen Kernen (SC), die im Vergleich wesentlich leistungsschwächer sind als ein BC, dafür aber auch weniger Energie verbrauchen. Ein solcher Prozessor ist auf hochgradig paralleles Rechnen spezialisiert. [20]

(c) Dies entspricht einem heterogenen Mehrkernprozessor. Sollen verschiedene Aufgaben parallel berechnet werden, fehlt es diesem Ansatz an Flexibilität. Dafür bietet ein solcher Manycore durch seinen starken Prozessors hohe Rechen-

leistungen sowohl bei sequenziellen, als auch bei parallelen Aufgaben, wenn bei diesen alle simplen Kerne genutzt werden. [20, 16]

(d) Es können, wie hier zu sehen, auch mehrere große Kerne bei einer Mehrzahl an kleineren zum Einsatz kommen. [20]

2.2 Verbindungsstrukturen

Informationsaustausch zwischen den einzelnen Kernen, dem gemeinsamen Speicher sowie dem I/O Interface eines Mehrkernprozessors ist bei Mehrkernprozessoren von zentraler Wichtigkeit. Frühe CMPs nutzen hierfür einen gemeinsamen Bus [18]. Mit den neuen Herausforderungen der immer komplexer werdenden Chips haben sich bis heute einige unterschiedliche Verbindungsstrukturen entwickelt. [1, 18]
Im Kapitel 3.1.2 wird auf diese genauer eingegangen, da die Wahl der Verbindungsart eine wichtige Rolle beim Design eines Mehrkernprozessors ist. Abbildung 2 zeigt einige der wichtigsten Topologien.

1. **Bus**: Ein Bus bietet einen gemeinsamen Kommunikationsweg für Prozessor, Caches, I/O und Speicher in einem CMP System [10] .

2. **Ring**: Die Kerne sind untereinander in Form eines Rings verbunden. Information wird über benachbarte Knoten weitergegeben. Ringe sind entweder unidirektional oder wie hier, bidirektional [12].

3. **Mesh**: Ein Mesh, oder auch Network-on-Chip (NoC), gehört zu den sog. direkten Netzen. Jeder Kern ist mit einer kleinen Anzahl benachbarter Kerne verbunden. Ein 2D-NoC ist gitterartig aufgebaut [3, 18, 1].

4. **Crossbar**: Crossbar ist ein indirektes Netz [3] und besteht aus Adressbussen, die von jedem Kern zu allen Caches geht. Jeweils ein Datenbus führt von jeder Cache-Bank zu den Kernen [10].

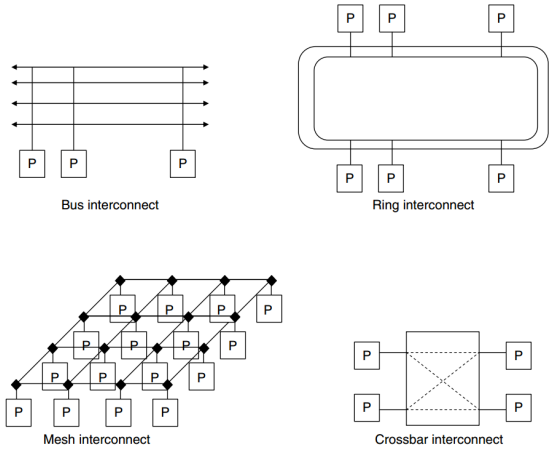

Abbildung 2: Verbindungsstrukturen [18]

2.3 Shared Memory

Mehrkernprozessoren greifen über die Verbindungsstruktur auf den Arbeitsspeicher zu. Die Daten werden dabei in Caches geladen. Ein Cache dient als Puffer zwischen Prozessor und Arbeitsspeicher. Er besitzt meist mehrere Hierarchien und bietet wesentlich höhere Geschwindigkeiten als der RAM, um so die Zugriffszeiten auf die Daten zu verringern. [18]
Jeder Kern besitzt einen privaten L1 Cache, der die höchst Geschwindigkeit bietet. Der langsamere L2 Cache wird sich in den vielen heutigen Architekturen unter allen Kernen über einen gemeinsamen Adressraum geteilt (Shared Memory). Ein solcher stellt eine zusätzliche Herausforderung für Programmierung und Architektur dar und erfordert eine regelnde Instanz, den Speichercontroller [3, 18]. Unabhängig davon, wie der Speichercontroller konkret realisiert wird, muss bei allen CMPs garantiert werden, dass das System cache-kohärent ist. Dies bedeutet, dass wegen der zahlreichen Kopien derselben Daten in den Cache-Hierarchien, ein regelnder Mechanismus garantiert, dass bei jedem Lesezugriff der aktuellste Wert geliefert wird, egal, in welchen Cache dieser zuvor geschrieben wurde. Alle Kopien sollen Cache-Konsistenz aufweisen. Es werden dafür Cache-Kohärenz Protokolle eingesetzt. [18, 12]

Herrscht Cache-Konsistenz, sind Daten in Caches und Hauptspeicher identisch. Dies ist wichtig, wenn Werte der verschiedenen Caches neu beschrieben werden. Auf diese Änderungen reagiert ein cachekohärenter Prozessor, indem alle Kerne über das neue Speicherabbild informiert werden.

3 Herausforderung Skalierung

Dieses Kapitel behandelt Schwierigkeiten, die sich bei der Entwicklung von Manycore-Prozessoren ergeben. Hauptaugenmerk liegt auf der Skalierbarkeit und welche Faktoren diese beeinflussen. Dabei werde anfangs auf die physikalisch limitierenden Faktoren und anschließend auf die Herausforderungen des Chipdesigns eingegangen.

3.1 Physikalische und wirtschaftliche Grenzen

Die Rechenleistung von Mehrkernprozessoren lässt sich durch Erhöhung der Anzahl an Kerne und Transistoren skalieren. Letztere werden dabei immer kleiner. Doch durch diese Maßnahmen treten physikalische Effekte auf, die im folgenden Kapitel behandelt werden [6].

3.1.1 Strukturgröße

Das Mooresche Gesetz steht voraussichtlich kurz vor seinem Ende und eine Erhöhung der Anzahl an Transistoren durch immer kleinere Strukturgrößen kann nur noch für einen kurzen Zeitraum fortgeführt werden [2, 11]. Die Verkleinerung der Transistoren wird demnach spätestens bei einer Strukturgröße von etwa 5nm enden. Dies scheint nach dem aktuellen Stand die Grenze für auf Silizium basierende CMOS Transistoren zu sein. Das liegt einerseits daran, dass ab einer Größenordnung von nur wenigen Atomdurchmessern nicht zu verachtende Quanteneffekte auftreten, die unter anderem zu einer Unzuverlässigkeit der Hardware führen [18]. Zum anderen wird der Produktionsprozesses immer aufwändiger und teurer. Transistoren werden heute mittels Lithografie bei UV-Licht der Wellenlänge von 193nm auf die Wafer aufgetragen. Für eine scharfe Abbildung kleinster Strukturen werden jedoch kürzere Wellenlängen erforderlich. Bisher konnte dank trickreicher Beleuchtungstechniken und der stetig abnehmende Chipfläche pro Transistor 193nm UV-Licht nach wie vor verwendet und die Kosten pro Transistor reduziert werden. Doch sobald der Aufwand und damit die Herstellungskosten stärker ansteigen, als sich die Fläche pro Transistor verringert, wird auch der Preis letzterer steigen und das Mooresche Gesetz verliert an seiner Gültigkeit. [11]

3.1.2 Dark Silicion

Des Weiteren ist es technisch unmöglich sehr große Mengen an Transistoren gleichzeitig an- und auszuschalten. Dieses Problem ist bekannt als Dark Silicon. Damit wird der Anteil an Transistoren eines Chips bezeichnet, der nicht die ganze Zeit mit voller Frequenz arbeiten kann [18, 17]. Der begrenzende Faktor ist dabei ein bestimmter TDP. Wären alle Transistoren aktiv, übersteigt die thermische Verlustleistung den Wert unter Umständen sehr stark, was eine aufwändigere Kühlung erfordert. Wirtschaftlich betrachtet ist dies jedoch nicht erwünscht [8]. Der Anteil an Dark Silicon wächst mit jeder Prozessor-Generation stark an. Nach einigen Schätzungen wird davon ausgegangen, dass Dark Silicon bei etwa 8nm Strukturgröße mehr als 50% ausmachen werden [4].
Mit einer Vergrößerung der Chipfläche ließe sich die Wärme besser abführen. Doch dieser Ansatz erweist sich insofern als nicht wirtschaftlich, da dies die Ausbeute an Chips pro Wafer deutlich reduzieren würde. Eine Verkleinerung der Chips mit weniger Transistoren und damit einem geringeren Anteil an Dark Silicon ist auch problematisch, da unter anderem die Energiedichte des Chips exponentiell anwächst was eine die Kühlung erschwert. Zudem bedeuten kleiner Chips nicht zwangsläufig auch geringere Produktionskosten. Es existieren jedoch weitere Ansätze, bei denen u.a. Dark Silicon ausgenutzt wird und eine Verringerung der Taktfrequenz das Problem weiter aufschieben lässt.[18, 17]

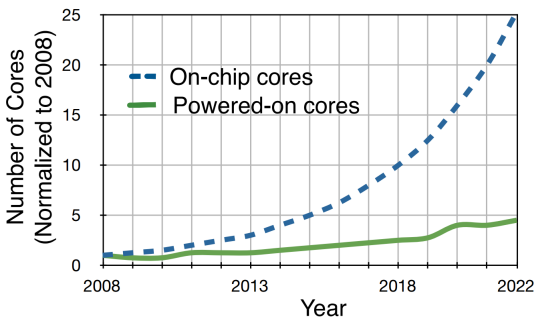

Abbildung 3: Power Wall bei Manycore [8]

Die Problematik lässt sich anlog auf die Anzahl an Kerne eines Manycore-Prozessors übertragen werden. Abbildung 3 veranschaulicht den sog. Manycore Power Wall. Es wird deutlich, dass ein aus wirtschaftlichen Gründen nicht zu überschreitenden TDP die Anzahl an aktiven Kernen in Zukunft immer stärker limitieren wird. [8]

3.2 Herausforderungen im Chipdesign

Neben den Herausforderungen, die phyisikalische Grenzen als Ursache haben, gibt es zahlreiche weitere Schwierigkeiten im Chipdesign selbst.

3.2.1 Wahl der Verbindungsstruktur

Latenz, Speicherbandbreite und Effizienz sind wesentliche Faktoren, die bei der Skalierung von Mehrkernprozessoren beachtet werden müssen.
Latenz entsteht unter anderem in der Verbindungsstruktur. Dort ist sie definiert als die Zeit, die eine Nachricht braucht, um von Sender zu Empfänger zu wandern. [18, 3].
Speicherbandbreite ist ein Faktor, der bei einer steigenden Anzahl an Prozessorkernen mitwachsen muss. Eine große Bandbreite benötigt dabei mehr Fläche auf dem Chip, wodurch im Umkehrschluss weniger Platz für Kerne und Caches bleibt [10].
Der Energieverbrauch resultiert hauptsächlich aus den Kernen. Doch auch die Verbindungsart des Prozessors trägt einen nicht zu verachtenden Anteil bei. In einem 8-Kern Prozessor kann diese beispielsweise den Verbrauch eines einzelnen Kerns ausmachen. [10, 1]
In Kapitel 2.2. wurden bereits gängige Verbindungsarten erläutert. Im Folgenden soll genauer drauf eingegangen werden, wie sich die unterschiedlichen Verbindungsstrukturen auf die Skalierbarkeit auswirken.

1. **Bus- und Ring-Topologie:** Busse und Ringe sind für Manycore-Prozessoren schlecht geeignet. Die langen Leitungen verursachen hohen Kapazitäten und erhöhen so die Latenz [18]. Bei einer hohen Anzahl an Kernen gelten sie aufgrund ihrer limitierten Bandbreite ebenfalls als unvorteilhaft, da so der verbleibende Datendurchsatz pro Kern sinkt [3]. Zudem steigt in einem Bus-System durch erhöhten Datenverkehr auch die Verlustleistung an, was die Energieeffizienz mindert [10].

2. **Crossbar:** Bei einer Crossbar-Struktur ist jeder Kern über eine Matrixstruktur mit jedem Speicher verbunden. Dies wird über N^2 Schalter realisiert, wobei N für die Anzahl an Kerne steht. Crossbar bietet so sehr geringe Latenz sowie Skalierbarkeit der Bandbreite mit sich. Ein großer Nachteil ist jedoch, dass sich die Anzahl an Leistungen und somit Fläche sowie Energieverbrauch, quadratisch erhöht. [1, 14]

3. **Mesh:** Während bei Crossbar- und Bus-Systemen das Kommunikationssytsem zenralisiert ist, ist es bei einem Mesh über den Chip verteilt. Jeder Kern inkl. Speicher hat seinen eigenen Router, der wie ein Crossbar-Schalter funktioniert. Meshes haben aufgrund ihrer kürzeren Leitungen weniger Verluste. Skalierbarkeit ist ebenso garantiert, da sich die Bandbreite mit der Kernzahl erhöht. Die Latenz nimmt jedoch mit wachsender Größe des Netzes zu. Man spricht hierbei von sog. Hops [19]. Damit ist die Anzahl der Kerne gemeint, die im Netz übersprungen werden muss, um Information von einem Kern

zum anderen zu übertragen. Bei Manycore-Prozessoren mit beispielsweise 1024 Kernen und einem 32x32 Netz, sind maximal 64 Sprünge nötig. Eine Verringerung der Anzahl an Hops schafft beispielsweise eine Technik, die sich 3D-Stacking nennt, und bei der mehrere 2D-Netze übereinander gestapelt miteinander verbunden werden. [18, 1, 14]

Die Wahl der Verbindungsstruktur hängt somit von mehreren Faktoren ab. Dazu zählen unter anderem die Anzahl an Kernen, benötigte Bandbreite sowie Chipfläche und Leistungsaufnahme. Bei kleinen bis mittelgroßen Mehrkernprozessoren kommen typischerweise Crossbar-Topologien zum Einsatz. Eine größere Anzahl an Kernen ist mit 2D-/3D-Meshes realisierbar. [1]

3.2.2 Energieverbrauch:

Neben der Verbindungsstruktur ist auch der Energieverbrauch der einzelnen Kerne ein limitierender Faktor. Dieser und der TDP legen die maximale Anzahl an Kernen auf dem Chip fest. Eine stetig verbesserte Energieeffizienz ist somit Voraussetzung für die Entwicklung von Manycore-Prozessoren. Die Effizienz ist definiert als Rechenleistung pro Energieverbrauch und gilt es als möglichst hoch zu halten. [18, 10].
Eine Möglichkeit den Energieverbrauch der Kerne zu senken ist es, die Taktfrequenz zu reduzieren. Dies erfordert jedoch im gleichen Zug eine höhere Anzahl an Kernen, um die durch niedrigeren Takt verursachten Einbußen in der Rechenleistung auszugleichen. [18]

3.2.3 Speicher

Memory Wall:
Unter Memory-Wall versteht man die Entwicklung, bei der die Rechenleistung der Kerne in einem weit stärkeren Verhältnis ansteigt, als sich die Zugriffszeiten auf den Speicher und damit die Bereitstellung der Daten verbessern lässt [18, 12]. Die Leistungsfähigkeit von Prozessor als auch Speicher wachsen, wie in Abbildung 4 zu sehen ist, beide an. Der Verlauf der CPU-Leistung steigt dabei jedoch um ein Vielfaches schneller an als der des Speichers. Die Differenz, erhöht sich somit zunehmend mit der Zeit. [7]

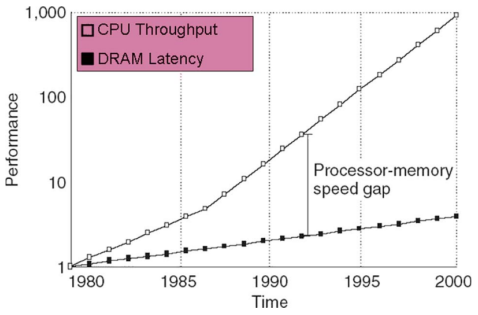

Abbildung 4: Memory Wall [7]

Die Leistungsfähigkeit eines Speichers wird dabei durch seine Zugriffszeit, der Latenz, definiert. Unter dieser versteht man die Zeit, die benötigt wird, um eine Speicheroperation auszuführen. Stetig verbesserte und größere Caches, Pre-Fetch-Mechanismen sowie Scouting- und Helfer-Threads sind gängige Methoden die Leistungsdifferenz klein zu halten. Auch 3D-Stacking gilt als Technologie, mit der in Manycore-Prozessoren dem Problem des Memory-Wall entgegengewirkt werden kann. [18, 1, 7].

3.3 Paralleles Programmieren - Das Gesetz von Amdahl

Im Bereich des parallelen Programmierens mit Manycore-Prozessoren stellt sich die Frage, in wie weit man Programme mit der immer größeren Anzahl an Prozessor-Kernen beschleunigen kann. Bereits im Jahre 1967, noch lange bevor die ersten CMPs auf den Markt kamen, befasste man sich von der theoretischen Seite der Informatik bereits damit. Gene Amdahl, US-amerikanischer Computerarchitekt, beantwortete diese mit einem Modell, das sich Amdahlsches Gesetz nennt und eine theoretische Grenze des Beschleunigens durch Parallelisierung beschreibt. [18, 20]

Die Formel dazu lautet

$$S = \frac{1}{1 - P + \frac{P}{N}} \tag{1}$$

wobei S der Speedup-Faktor ist und P den Anteil an parallel ausführbarem Code-Anteil beschreibt. N steht für die Anzahl der Kerne. [18]

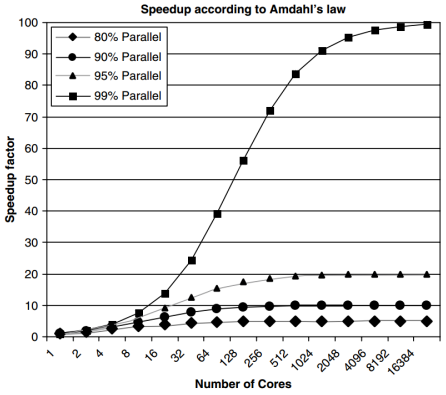

Abbildung 5: Speedup bei homogenen CMPs [18]

Abbildung 5 veranschaulicht Amdahls Gesetz. Es zeigt den sog. Speedup-Faktor im Bezug auf die Anzahl homogener Kerne und sequenziellen Anteil des Codes. Der Speedup-Faktor gibt an, um das Wievielfache sich das Ausführen des Codes, im Vergleich zum Einsatz mit einem einzelnen Kern, beschleunigt. Aus der Grafik wird deutlich, dass gute Leistungsskalierbarkeit nur bei hochgradig parallelisierte Anwendungen erreicht wird. Die Rechenleistung der einzelnen Kerne spielt dabei keine Rolle. [18, 20]

Für heterogene Manycore-Prozessoren ergibt sich die Darstellung in Abbildung 6. Es wird hier ein modellhafter Prozessor angenommen, der eine festgesetzte Anzahl an Ressourcen zur Verfügung stellt, die entweder in 64 schwache Kerne, oder in eine heterogenen Konstellation mit Kernen verschiedener Leistungen zusammengesetzt werden kann. Man erkennt, dass der Speedup durch heterogene Prozessoren größer sein kann, als es Amdahl in seiner Formel vorhersagt. Des weiteren wird ersichtlich, dass es für unterschiedliche Anteile an sequenziellem Code unterschiedlich liegende Maxima des Speedup-Faktors gibt. Diese bedeuten, dass sich für eine maximale Skalierung keine algemeingültige Konfiguratio finden lässt, sondern es auf die jeweilige Anwendung und damit Größe des sequenziellen Anteils ankommt. [18, 20, 16]

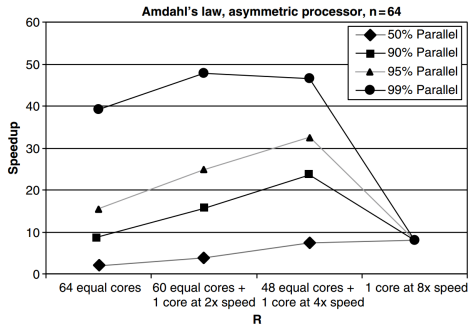

Abbildung 6: Speedup bei heterogenen CMPs [18]

4 Manycore Architekturen

Im Folgenden werden exemplarisch zwei Manycore Architekturen vorgestellt und auf die in den vorangegangenen Kapiteln erläuterten Charakteristiken untersucht. Es ist zu erwähnen, dass es neben diesen noch zahlreiche weitere Architekturen, wie etwa GPUs oder Neuromorphische Prozessoren, existieren. Diese werden in dieser Arbeit jedoch nicht näher erläutert, sondern in anderen Seminararbeiten ausführlich behandelt.

4.1 Tilera TILE

TILE ist eine Architektur für einen homogenen Manycore-Prozessor und wurde von der Firma Tilera, welche 2014 von EZChip übernommen wurde, entwickelt. Als Inspiration diente dabei der, vom MIT entworfene, Raw Forschungs-Prozessor. Besonders nennenswert sind bei dieser Architektur die Verbindungsstruktur sowie die Cache-Organisation. [19]

4.1.1 iMesh

Die Verbindungsstruktur nennt sich *iMesh* und besteht aus fünf separaten 2D-NoCs. Jedes der Netze ist mit kachelartig angeordneten, identischen Tiles verbunden. Diese entsprechen jeweils einer einzelnen, leistungsfähigen Recheneinheit des Prozessors. TILE64 besteht beispielsweise aus einem Mesh mit 8 x 8 Tiles. Jedes davon besitzt neben einem Kern noch assoziativen L1 und L2 Cache sowie einen Crossbar Schalter, der die Verbindung zu den einzelnen Verbindungsstrukturen darstellt. Man sagt deshalb, dass iMesh ein sog. Switched-Network ist. Die Aufgabenfelder der Netze unterscheiden sich jedoch. Zu diesen gehören Prozesskommunikation, I/O Kommunikation, Speicherkommunikation, Cache-Kohärenz sowie ein statischer Kommunikationskanal. Abbildung 7 zeigt schematisch die TILE Architektur. [18, 19]
Der Vorteil, der sich aus einer solchen Anordnung ergibt, ist eine sehr hohe Bandbreite von 1,28 TBit/s, die einem einzelnen Tile bei der Kommunikation zur Verfügung steht. Zudem kann der Benutzer direkt auf die Netze zugreifen, was als *TileDirect* bezeichnet wird. Das heißt, dass Daten ins Mesh geladen werden, ohne den Umweg des DDR Speichers gehen zu müssen. So wird die zusätzliche Bandbreite genutzt, damit Programme gleichzeitig Daten auf die unterschiedlichen Netze senden und auch zwischen diesen kommunizieren können. Der Datenverkehr verläuft so getrennt und Interferenzen werden vermindert. [18, 19].

4.1.2 Cache-Organisation

Die TILE Architektur besitzt neben den L1 und L2 Caches noch einen gemeinsamen, virtuellen L3 Cache. Dieser besteht aus den, über den Chip verteilten L2 Caches und ist als eine Kopie dieser in einem sog. *HomeTile* komplett abgespeichert. Mittels eines der fünf Netze kann so von jedem Tile aus auf den gemeinsamen Speicher zugegriffen werden und Cache-Kohärenz garantiert werden. Zudem bewirkt die relative Nähe zum HomeTile sowie die hohe Bandbreite eine gute Geschwindigkeit des Caches. Ein Nachteil ist jedoch, dass der Zugriff auf verschiedene Bereiche des Caches abhängig von der Distanz der Tiles ist. [18, 19]

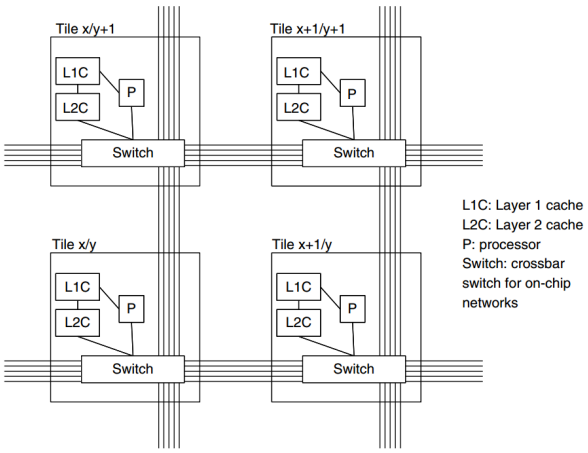

Abbildung 7: TILE Architektur [18]

4.2 Intel Xeon Phi - Knights Landing

Mit Knights Landing (KNL) brachte Intel 2016 seine zweite Generation an Xeon Phi Prozessoren auf den Markt. Beim Xeon Phi KNL handelt es sich um einen eigenständigen Manycore-Prozessor, der auf HPC und hochgradiges paralleles Rechnen wie Deep Learning spezialisiert ist. Dafür bieten er diverse Innovationen in seiner Architektur. [15]

4.3 Tiles

Der Prozessor besitzt 36 aktive Tiles. Jede davon besteht aus zwei Rechenkernen sowie zwei Vektorprozessoren (VPU) und einem 1MB L2 Cache. Bei den Kernen handelt es sich um stark an die Bedürfnisse des HPC angepasste Intel Atom Kerne. Sie bieten unter anderem 4 Threads pro Kern sowie höhere Cache-Bandbreite und Größe an. VPUs sind spezialisiert auf die parallele Berechnung mehrerer Daten, die eine gleichartige Bearbeitungsweise erfordern. [15]

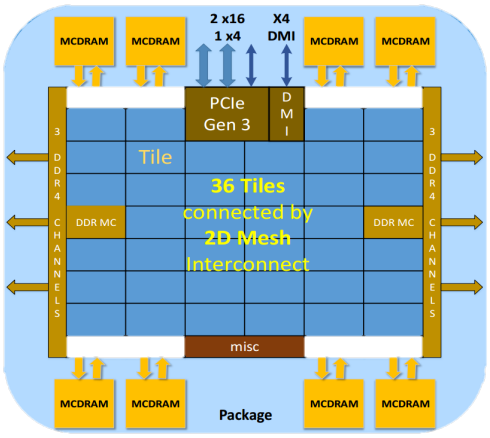

Abbildung 8: Intel Xeon Phi KNL [15]

4.4 Verbindungsstruktur

Von weiterer Bedeutung für diese Arbeit ist die Verbindungsstruktur. Beim Xeon Phi KNL sind die einzelnen Tiles über ein cache-kohärentes 2D-Mesh miteinander verbunden. Dieses basiert auf einer Ring-Architektur, ist aber keine in dem Sinn. Dadurch werden höhere Bandbreite und geringere Latenz verglichen mit einer reinen Ring-Verbindung, wie sie beim Xeon Phi der ersten Generation eingesetzt wurde, erzielt. Das Mesh kann vier parallele Netzwerke versorgen, bei dem jedes unterschiedliche Arten von Paketen braucht. Dadurch reduziert sich der Datenverkehr und eine sehr hohe gesamte Bandbreite von bis zu über 700 GB/s wird erreicht. Die Datenübertragung folgt dabei einer sog. XY-Routing Regel. Dabei werden Daten zuerst durch die vertikale und anschließend die horizontale Leitung des Meshes verschickt, wodurch Deadlocks reduziert werden. [15, 3]

4.5 Speicher

Der Hauptspeicher des Xeon Phi KNL ist ebenso eine Besonderheit. Es werden hier zwei Arten von Speicher verwendet. Zum einen den sogenannten Multi-Channel DRAM (MCDRAM), der 16GB bemisst und eine hohe Speicherbandbreite zur Verfügung stellt. Zum anderen kommt ein DDR4 Speicher zum Einsatz, der mit bis zu 384 GB für eine hohe Speicherkapazität sorgt. [15]

5 Schluss und Ausblick

In dieser Arbeit wurden Grundlagen von Mehrkernprozessoren vorgestellt und Schwierigkeiten genannt, die sich bei der Entwicklung von Manycore-Prozessoren ergeben. Es wurde ersichtlich, dass neben physikalisch limitierenden Faktoren, wie der Strukturgröße und Dark Silicon, auch Energieeffizienz und Speicherbandbreite immer mehr zum Flaschenhals für eine weitere Skalierung von CMPs geworden sind. Folglich ist das Design von Verbindungsstrukturen, Caches und Kernen entscheidend. [10, 18] Der Intel Xeon Phi KNL und Tileras TILE Prozessor dienten anschließend als Beispiele für Manycore Architekturen und wurden auf die, für diese Arbeit, wesentlichen Eigenschaften behandelt.
Abschließend ist zu sagen, dass auch in der Zukunft, und nicht nur im Bereich des HPC, die Entwicklung neuer Prozessor-Architekturen wohl immer mehr in Richtung Manycore gehen wird. Denn nur so ist es möglich bei steigender Transistor-Anzahl und unter wirtschaftlichen Aspekten die Rechenleistung von Prozessoren weiter anzuheben. [5, 21, 10]

Literatur

[1] Abderazek Ben Abdallah. *Multicore Systems On-Chip: Practical Software/Hardware Design*. Atlantis Publishing Corporation, 2013.

[2] A. A. Chien and V. Karamcheti. Moore's law: The first ending and a new beginning. *Computer*, 46(12):48–53, Dec 2013.

[3] Jose Duato, Sudhakar Yalamanchili, and Ni Lionel. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[4] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark si-

licon and the end of multicore scaling. *IEEE Micro*, 32(3):122–134, May 2012.

[5] D. Geer. Chip makers turn to multicore processors. *Computer*, 38(5):11–13, May 2005.

[6] Mark D. Hill and Michael R. Marty. Amdahl's law in the multicore era. *Computer*, 41(7):33–38, July 2008.

[7] P. Jacob, A. Zia, O. Erdogan, P. M. Belemjian, J. W. Kim, M. Chu, R. P. Kraft, J. F. McDonald, and K. Bernstein. Mitigating memory wall effects in high-clock-rate and multicore cmos 3-d processor memory stacks. *Proceedings of the IEEE*, 97(1):108–122, Jan 2009.

[8] U. R. Karpuzcu, B. Greskamp, and J. Torrellas. The bubblewrap many-core: Popping cores for sequential acceleration. In *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 447–458, Dec 2009.

[9] N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan. Leakage current: Moore's law meets static power. *Computer*, 36(12):68–75, Dec 2003.

[10] R. Kumar, V. Zyuban, and D. M. Tullsen. Interconnections in multi-core architectures: understanding mechanisms, overheads and scaling. In *32nd International Symposium on Computer Architecture (ISCA'05)*, pages 408–419, June 2005.

[11] C. Mack. The multiple lives of moore's law. *IEEE Spectrum*, 52(4):31–31, April 2015.

[12] Michael R. Marty. *Cache Coherence Techniques for Multicore Processors*. PhD thesis, Madison, WI, USA, 2008. Adviser-Hill, Mark D.

[13] J. Parkhurst, J. Darringer, and B. Grundmann. From single core to multi-core: Preparing for a new exponential. In *2006 IEEE/ACM International Conference on Computer Aided Design*, pages 67–72, Nov 2006.

[14] Korey LaMar Sewell. *Scaling High-Performance Interconnect Architectures to Many-Core Systems*. PhD thesis, The University of Michigan, 2012.

[15] A. Sodani, R. Gramunt, J. Corbal, H. S. Kim, K. Vinod, S. Chinthamani, S. Hutsell, R. Agarwal, and Y. C. Liu. Knights landing: Second-generation intel xeon phi product. *IEEE Micro*, 36(2):34–46, Mar 2016.

[16] W. J. Song, S. Mukhopadhyay, and S. Yalamanchili. Amdahl's law for lifetime reliability scaling in heterogeneous multicore processors. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 594–605, March 2016.

[17] M. B. Taylor. A landscape of the new dark silicon design regime. *IEEE Micro*, 33(5):8–19, Sept 2013.

[18] Andrs Vajda. *Programming Many-Core Chips*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[19] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C. C. Miao, J. F. Brown III, and A. Agarwal. On-chip interconnection architecture of the tile processor. *IEEE Micro*, 27(5):15–31, Sept 2007.

[20] D. H. Woo and H. H. S. Lee. Extending amdahl's law for energy-efficient computing in the many-core era. *Computer*, 41(12):24–31, Dec 2008.

[21] X. Zhang and A. Louri. A multilayer nanophotonic interconnection network for on-chip many-core communications. In *Design Automation Conference*, pages 156–161, June 2010.