



# BUNDESPATENTGERICHT

17 W (pat) 5/20

---

(Aktenzeichen)

## BESCHLUSS

In der Beschwerdesache

**betreffend die Patentanmeldung 10 2017 213 160.8**

...

hat der 17. Senat (Technischer Beschwerdesenat) des Bundespatentgerichts am 2. März 2023 unter Mitwirkung des Vorsitzenden Richters Dipl.-Phys. Dr. Morawek, des Richters Dipl. Ing. Baumgardt, des Richters Dipl.-Ing. Hoffmann und der Richterin Akintche

beschlossen:

Auf die Beschwerde der Anmelderin wird der Beschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts vom 20. Februar 2018 aufgehoben und das Patent mit folgenden Unterlagen erteilt:

Patentansprüche 1 bis 14,  
Beschreibung Seiten 1 bis 65 (mit Korrektur eines offensichtlichen Schreibfehlers auf Seite 1, vorletzte Zeile), und  
22 Blatt Zeichnungen mit Figuren 1 bis 13,  
jeweils wie eingereicht am 28. Februar 2023.

## **Gründe**

### **I.**

Die vorliegende Patentanmeldung, welche die Prioritäten dreier Voranmeldungen in den USA vom 25. August 2016, vom 14. September 2016 und vom 1. Februar 2017 in Anspruch nimmt, wurde am 31. Juli 2017 beim Deutschen Patent- und Markenamt in englischer Sprache eingereicht. In der nachgereichten deutschen Übersetzung trägt sie die Bezeichnung:

„Kompilierung für knotenvorrichtung-GPU-basierte Parallelverarbeitung“.

Die Anmeldung war in der Anhörung vom 20. Februar 2018 durch Beschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts mit der Begründung zurückgewiesen worden, dass die Gegenstände der jeweiligen Patentansprüche 1 nach dem (damals geltenden) Hauptantrag sowie den

Hilfsanträgen 2 bis 5 nicht auf einer erfinderischen Tätigkeit des Fachmanns beruhen, soweit sie ein technisches Problem mit technischen Mitteln lösen und damit eine Erfindung auf Gebieten der Technik betreffen; Patentanspruch 1 nach dem Hilfsantrag 1 sei zurückzuweisen gewesen, da er den Gegenstand der Anmeldung unzulässig erweitere.

Gegen diesen Beschluss ist die Beschwerde der Anmelderin gerichtet. Auf einen Zwischenbescheid des Senats vom 5. April 2022 hin hat sie eine überarbeitete Fassung der Patentansprüche eingereicht, um die unter Schutz zu stellende Erfindung klar und deutlich zu umschreiben und um sich gegen den bekannten Stand der Technik abzugrenzen; die Beschreibung und die Zeichnungen wurden daran angepasst. Mit der Eingabe vom 28. Februar 2023 stellt die Anmelderin den Antrag,

den Zurückweisungsbeschluss der Prüfungsstelle aufzuheben und ein Patent zu erteilen auf Basis der folgenden Unterlagen:

Patentansprüche 1 bis 14,  
Beschreibung Seiten 1 bis 65, und  
22 Blatt Zeichnungen mit Figuren 1 bis 13,  
jeweils wie eingereicht am 28. Februar 2023.

Das geltende Patentbegehren, hier bezüglich des Hauptanspruchs mit einer möglichen Gliederung versehen, lautet:

- V0** 1. Computerimplementiertes Verfahren zur wenigstens teilweise parallelen Durchführung von Aufgaben von Analyse-routinen mittels einer Mehrzahl von Knotenvorrichtungen (1500), umfassend:
- V1** Empfangen einer Analyseroutine, die ausführbare Anweisungen für eine oder mehrere Aufgaben in Form von

Taskroutinen zur Ausführung durch wenigstens eine zentrale Verarbeitungseinheit (Central Processing Unit, CPU) einer Knotenvorrichtung umfasst, die mit wenigstens einem Datensatz durchgeführt werden sollen,

- V2** Analysieren eines aktuellen Status von Ressourcen von wenigstens einer Knotenvorrichtung der Mehrzahl von Knotenvorrichtungen, um eine Verfügbarkeit von wenigstens einer Grafikverarbeitungseinheit (Graphics Processing Unit, GPU) der wenigstens einen Knotenvorrichtung zu bestimmen,
  - V2.1** wobei die genannten Ressourcen zumindest eine CPU und zumindest eine GPU umfassen, und
- V3** wenn das Bestimmen ergibt, dass wenigstens eine GPU der wenigstens einen Knotenvorrichtung für die Zuweisung einer ersten Aufgabe der Analyseroutine in Form einer ersten Taskroutine verfügbar ist,
  - V3.1** Analysieren der ersten Taskroutine, ob diese in eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU kompiliert werden kann, so dass die wenigstens eine GPU mehrere Instanzen der ersten Aufgabe der Analyseroutine wenigstens teilweise parallel durchführen kann,
    - V3.1.1** wobei das Analysieren umfasst, ob die erste Taskroutine eine Anweisung umfasst, die eine Kompilierung zur Erzeugung einer GPU-Taskroutine verhindert,

- V3.1.2** und ob Eingaben und Ausgaben der ersten Taskroutine Abhängigkeiten aufweisen, die eine Kompilierung zur Erzeugung einer GPU-Taskroutine verhindern,
- V3.2** wenn das Analysieren ergibt, dass die erste Taskroutine in eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU kompiliert werden kann,
- V3.2.1** Verwenden einer Umwandlungsregel, um wenigstens eine Anweisung der ersten Taskroutine in wenigstens eine entsprechende Anweisung der GPU-Taskroutine umzuwandeln;
- V3.2.2** Kompilieren der wenigstens einen entsprechenden Anweisung der GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU;
- V3.2.3** Zuweisen einer Datensatzpartition des Datensatzes an die wenigstens eine Knotenvorrichtung, um Zugriff auf die Datensatzpartition durch die wenigstens eine GPU zu ermöglichen, und
- V3.2.4** Zuweisen einer Durchführung der GPU-Taskroutine an die wenigstens eine Knotenvorrichtung, um die Durchführung der mehreren Instanzen der ersten Aufgabe mit der Datensatzpartition durch die wenigstens eine GPU mittels der GPU-Taskroutine zu ermöglichen.
2. Computerimplementiertes Verfahren nach Anspruch 1, wobei die genannten Ressourcen Speicherplatz innerhalb wenigstens eines Speichers der wenigstens einen Knotenvorrichtung umfassen.

3. Computerimplementiertes Verfahren nach Anspruch 1 oder 2, wobei das Analysieren der ersten Taskroutine, ob diese in eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU kompiliert werden kann, umfasst:

Bestimmen, ob die Anweisung der ersten Taskroutine in einem Satz von Anweisungen eingeschlossen ist, die nicht in wenigstens eine Anweisung umgewandelt werden können, die von der wenigstens einen GPU ausgeführt werden kann; und

in Reaktion auf das Bestimmen, dass die Anweisung der ersten Taskroutine nicht im Satz von Anweisungen eingeschlossen ist, Bestimmen, ob die Anweisung der ersten Taskroutine in der ersten Taskroutine in einer Weise verwendet wird, die eine Umwandlung in wenigstens eine Anweisung verhindert, die von der wenigstens einen GPU ausgeführt werden kann.

4. Computerimplementiertes Verfahren nach Anspruch 1 oder 2, wobei das Umwandeln der wenigstens einen Anweisung der ersten Taskroutine in die wenigstens eine entsprechende Anweisung der GPU-Taskroutine umfasst,

die wenigstens eine Anweisung der ersten Taskroutine von einer ersten Programmiersprache in die wenigstens eine entsprechende Anweisung in einer zweiten Programmiersprache gemäß der verwendeten Umwandlungsregel umzuwandeln.

5. Computerimplementiertes Verfahren nach Anspruch 2, wobei:

der wenigstens eine Speicher der wenigstens einen Knotenvorrichtung einen ersten flüchtigen Speicher, der kommunikativ mit der wenigstens einen CPU gekoppelt ist, und einen zweiten flüchtigen Speicher, der kommunikativ mit der wenigstens einen GPU gekoppelt ist, umfasst;

das Zuweisen der Datensatzpartition des Datensatzes an die wenigstens eine Knotenvorrichtung, um Zugriff auf die Datensatzpartition durch die wenigstens eine GPU zu ermöglichen, umfasst, dass veranlasst wird, dass die Datensatzpartition innerhalb des zweiten flüchtigen Speichers gespeichert wird.

6. Computerimplementiertes Verfahren nach Anspruch 5, wobei:

wenn das Bestimmen ergibt, dass keine GPU der wenigstens einen Knotenvorrichtung für die Zuweisung einer ersten Aufgabe der Analyseroutine in Form einer ersten Taskroutine verfügbar ist, das Verfahren umfasst:

Unterlassen des Analysierens der ersten Taskroutine, ob diese in eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU kompiliert werden kann;

Zuweisen der Datensatzpartition des Datensatzes an die wenigstens eine Knotenvorrichtung, um eine Speicherung der Datensatzpartition innerhalb des ersten flüchtigen Speichers zu veranlassen und um Zugriff auf die Datensatzpartition durch die wenigstens eine CPU zu ermöglichen;

Kompilieren der ersten Taskroutine zur Ausführung durch die wenigstens eine CPU; und

Zuweisen einer Durchführung der ersten Aufgabe der Analyseroutine mit der Datensatzpartition an die wenigstens eine Knotenvorrichtung, um die Durchführung der ersten Aufgabe mit der Datensatzpartition durch die wenigstens eine CPU zu ermöglichen.

7. Computerimplementiertes Verfahren nach Anspruch 1, 2 oder 6, wobei:

eine Koordinierungsvorrichtung den Betrieb der Mehrzahl von Knotenvorrichtungen koordiniert;

die Koordinierungsvorrichtung wiederkehrend Aktualisierungen für den aktuellen Status von jeder Knotenvorrichtung der Mehrzahl von Knotenvorrichtungen empfängt; und

das Analysieren eines aktuellen Status von Ressourcen, um eine Verfügbarkeit von wenigstens einer GPU der wenigstens einen Knotenvorrichtung zu bestimmen, umfasst:

das Identifizieren, durch die Koordinierungsvorrichtung, einer Knotenvorrichtung der Mehrzahl von Knotenvorrichtungen, die eine GPU umfasst, die im aktuellen Status als verfügbar angezeigt wird.

8. Computerimplementiertes Verfahren nach Anspruch 7, wobei das Zuweisen einer Datensatzpartition des Datensatzes an die wenigstens eine Knotenvorrichtung umfasst:



Analysieren, durch die Koordinierungsvorrichtung, von Metadaten, die indikativ für strukturelle Merkmale des Datensatzes sind, um eine Beschränkung der Art und Weise zu identifizieren, wie der Datensatz in die Mehrzahl von Datensatzpartitionen aufgeteilt werden kann,

wobei die Beschränkung in einer Angabe einer kleinsten atomaren Einheit von Daten innerhalb des Datensatzes und/oder in einer Spezifikation eines Partitionierungsschemas besteht; und

Ableiten einer Aufteilung des Datensatzes in die Mehrzahl von Datensatzpartitionen wenigstens teilweise basierend auf der Beschränkung.

9. Computerimplementiertes Verfahren nach Anspruch 8, umfassend:

Abrufen der Metadaten aus wenigstens einer Speichervorrichtung, in welcher der Datensatz gespeichert ist; und

Übertragen, an die wenigstens eine Knotenvorrichtung oder die wenigstens eine Speichervorrichtung, einer Anzeige über die Zuweisung der Datensatzpartition, um eine Übertragung der Datensatzpartition von der wenigstens einen Speichervorrichtung an die wenigstens eine Knotenvorrichtung zu veranlassen.

10. Computerimplementiertes Verfahren nach Anspruch 1 oder 2, umfassend:

Analysieren einer zweiten Taskroutine der Analyseroutine, ob diese in eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU kompiliert werden kann, so dass die wenigstens eine GPU mehrere Instanzen einer zweiten Aufgabe der Analyseroutine wenigstens teilweise parallel durchführen kann,

wobei das Analysieren der zweiten Taskroutine umfasst, ob eine Abhängigkeit zwischen Eingaben und Ausgaben der mehreren Instanzen der zweiten Aufgabe vorhanden ist, die eine Kompilierung zur Erzeugung einer GPU-Taskroutine verhindert,

wobei die zweite Taskroutine zur Ausführung durch die wenigstens eine CPU erzeugt wurde, um eine zweite Aufgabe der Analyseroutine durchzuführen; und

wenn das Analysieren ergibt, dass die zweite Taskroutine nicht kompiliert werden kann, um die GPU-Taskroutine zu erzeugen:

Kompilieren der zweiten Taskroutine zur Ausführung durch die wenigstens eine CPU; und

Zuweisen einer Durchführung der zweiten Aufgabe der Analyseroutine mit der Datensatzpartition an die wenigstens eine Knotenvorrichtung, um die Durchführung der zweiten Aufgabe mit den Datensatzpartitionen durch die wenigstens eine CPU zu ermöglichen.

11. Computerimplementiertes Verfahren nach Anspruch 10, umfassend Analysieren der Analyseroutine, um eine

Reihenfolge von Aufgaben der Analyseroutine zu bestimmen, wobei die Reihenfolge der Aufgaben eine relative Reihenfolge der ersten und zweiten Aufgaben umfasst.

12. Computerprogrammprodukt, das in einem nicht-flüchtigen maschinenlesbaren Speichermedium ausgeführt ist, wobei das Computerprogrammprodukt Anweisungen umfasst, die in der Lage sind, einen Prozessor zu veranlassen, Operationen gemäß einem Verfahren nach einem der Ansprüche 1 bis 11 durchzuführen.
13. Vorrichtung, umfassend einen Prozessor und einen Speicher, um Anweisungen zu speichern, die, wenn sie durch den Prozessor ausgeführt werden, den Prozessor veranlassen, Operationen gemäß einem Verfahren nach einem der Ansprüche 1 bis 11 durchzuführen.
14. Vorrichtung nach Anspruch 13, wobei:

das Analysieren des aktuellen Status von Ressourcen der wenigstens einen Knotenvorrichtung, um eine Verfügbarkeit von wenigstens einer GPU der wenigstens einen Knotenvorrichtung zu bestimmen, das Analysieren durch eine CPU dieser Knotenvorrichtung umfasst, ob wenigstens eine GPU dieser Knotenvorrichtung aktuell verfügbar ist.

Eine konkrete **Aufgabe** ist in der Anmeldung nicht angegeben.

Im Zurückweisungsbeschluss (Seite 10 oben / Mitte) war festgestellt worden, die Anmeldung beschäftige sich mit der effektiven Nutzung der verteilten Rechenressourcen eines heterogenen Grids zur Ausführung von Algorithmen zur Analyse

großer Datenmengen. Das objektive technische Problem beim Gegenstand des Anspruchs 1 des Hauptantrags sei es, eine Vorrichtung zu schaffen, die für die Ausführung einer Analyseroutine eine möglichst performante Hard- und Softwarearchitektur zur Verfügung stellt.

In der Beschwerdebegründung (Seite 2 Absatz 3) wurde vorgetragen, die Erfinder hätten eine sehr spezielle Art und Weise ersonnen, wie die Arbeitsverteilung zwischen CPUs und GPUs möglichst effizient zur Parallelverarbeitung strukturiert werden könne.

## II.

Die Beschwerde wurde rechtzeitig eingelegt und ist auch sonst zulässig. Sie hat Erfolg, da das nunmehr geltende Patentbegehren durch den bekannt gewordenen Stand der Technik nicht vorweggenommen oder nahegelegt ist und auch die übrigen Kriterien für eine Patenterteilung erfüllt sind (PatG §§ 1 bis 5, § 34).

1. Die vorliegende Patentanmeldung betrifft die Nutzung verteilter Rechenressourcen eines heterogenen Grids (Netz von Rechenknoten), in welchem sowohl Zentralprozessoren (Central Processing Unit, CPU) als auch Grafik-Prozessoren (Graphics Processing Unit, GPU) zur Durchführung von Berechnungen verfügbar sind.

1.1 In der Beschreibungseinleitung (Abs. [0002] – *diese und die folgenden Absatzangaben beziehen sich auf die deutsche Übersetzung gemäß Offenlegungsschrift*) ist zunächst ausgeführt, dass es für die Analyse von großen Datensätzen üblich geworden sei, Datennetze (Grids) mit zahlreichen Rechenknoten einzusetzen, wobei die einzelnen „Aufgaben“ von Analyseroutinen soweit wie möglich parallel über die Knoten verteilt würden. Für diese Verteilung wird eine Koordination benötigt, wobei es gilt, konkurrierende Ziele wie eine möglichst

stetige Nutzung der einzelnen Ressourcen eines Knotens oder eine möglichst effektive Nutzung von Rechenleistung, Speicherkapazität und Netz-Bandbreite aufeinander abzustimmen.

Gemäß Abs. [0064] ist die Anmeldung auf Techniken gerichtet, welche die effektive Verwendung von Verarbeitungs-, Speicher- und Netzbandbreitenressourcen innerhalb eines Gitters von Knotenvorrichtungen verbessern, um zu ermöglichen, dass einzelne Analyseroutinen in kürzerer Zeit abgeschlossen werden.

Der neue Patentanspruch 1 betrifft ein „Computerimplementiertes Verfahren“, welches ausgeht von vorhandenen „Analyseroutinen“ (1210 – *dieses und die folgenden Bezugszeichen sind zur Erläuterung des Patentanspruchs aus der Beschreibung entnommen*) zur Datenanalyse großer Datensätze, z.B. statistische Analysen, Auswertungen von Hypothesen o.ä. (siehe z.B. Abs. [0181]), wobei die Analyseroutinen in mehrere „Aufgaben“ gegliedert sind, zu welchen „Taskroutinen“ (1211) existieren (vgl. etwa Abs. [0065]: „Die Koordinierungsvorrichtung kann auch eine Analyseroutine empfangen, die ausführbare Anweisungen für mehrere Taskroutinen für mehrere Aufgaben umfasst, die mit wenigstens einem Datensatz durchgeführt werden sollen...“; Abs. [0209]; und Anspruchs-Merkmale **V1**, **V3**). Für diese Analyseroutinen gibt die Anmeldung keine näheren Spezifikationen. Generell wird davon ausgegangen, dass die Analyseroutinen in einer Programmiersprache geschrieben sind, die nicht speziell auf die Besonderheiten der verfügbaren CPUs und GPUs abgestimmt ist (vgl. etwa Abs. [0071]); sie liegen jedoch „in Form von Taskroutinen zur Ausführung durch wenigstens eine zentrale Verarbeitungseinheit (Central Processing Unit, CPU)“ vor (Merkmal **V1**).

Zur Bearbeitung steht ein Netzwerk von Rechenknoten zur Verfügung, wobei die einzelnen Knoten grundsätzlich unterschiedliche „Ressourcen“ wie CPU, GPU und Speicherplatz bereitstellen (vgl. Merkmale **V2**, **V2.1**, Unteranspruch 2). So können die Rechenknoten eine oder mehrere CPUs (1550) und teilweise zusätzlich eine

oder mehrere GPUs (1650) enthalten (vgl. Abs. [0064], [0066] / [0067]; Abs. [0075] mit Bezug auf eine unterschiedliche Speicherausstattung).

Dem Fachmann war bekannt, dass CPUs und GPUs bezüglich der Abarbeitung von Programm-Routinen unterschiedliche Eigenschaften aufweisen. So sind CPUs i.d.R besser geeignet für die serielle Verarbeitung von Taskroutinen, die verschiedene Eingabe-/Ausgabeoperationen und/oder Verzweigungsoperationen enthalten; im Gegensatz dazu sind GPUs i.d.R besser geeignet für die parallele Verarbeitung von Taskroutinen, die einen relativ begrenzten Satz von Anweisungen für mathematische und/oder bitweise Operationen umfassen und insbesondere unabhängig voneinander ausgeführt werden können (vgl. Abs. [0066], [0213]); mit „hochgradig parallel“ („embarrassingly parallel“) sind Aufgaben bezeichnet, bei denen keine Abhängigkeiten zwischen Instanzen der Aufgabe bestehen, so dass eine parallele Verarbeitung einer relativ großen Menge von Instanzen der Aufgabe leicht möglich ist.

Um nun die Analyseroutinen möglichst optimal abarbeiten zu lassen, ist eine Koordinierungsvorrichtung (1300) vorgesehen, welche die Taskroutinen (1211) nach bestimmten Kriterien an die Rechenknoten verteilt (vgl. Abs. [0076]). Gemäß Abs. [0183] kann jeder Rechenknoten (1500) eine oder mehrere CPUs (1550) und zugeordneten Speicher (1560) insbes. für CPU-Taskroutinen (1571) enthalten. Ferner kann „wenigstens eine Teilmenge“ der Rechenknoten zusätzlich einen Grafikcontroller 1600 besitzen, der eine oder mehrere GPUs (1650) und Speicher (1660) insbes. für GPU-Taskroutinen (1671) enthält (siehe Abs. [0184] sowie Fig. 11A / 11B). Die genannten Rechenknoten werden in den Patentansprüchen durchgängig und in der Beschreibung meistens als „Knotenvorrichtungen“ bezeichnet.

**1.2** Der geltende Patentanspruch 1 gibt eine Lehre für die Arbeit einer solchen Koordinierungsvorrichtung, genauer: für die Koordinierung des Betriebs mehrerer Knotenvorrichtungen, um wenigstens teilweise eine parallele Durchführung von

Aufgaben von Analyseroutinen zu verwirklichen (Merkmal **V0**), die mit wenigstens einem Datensatz durchgeführt werden sollen. Dazu umfassen die vorgegebenen Analyseroutinen ausführbare Anweisungen in Form von Taskroutinen, welche zur Ausführung durch wenigstens eine zentrale Verarbeitungseinheit (Central Processing Unit, CPU) einer Knotenvorrichtung vorgesehen sind (Merkmal **V1**).

Ein erster Arbeitsschritt (**V2 / V2.1**) betrifft das Analysieren eines aktuellen Status von Ressourcen (CPU, GPU) von wenigstens einer Knotenvorrichtung, insbesondere ob dort eine Grafikverarbeitungseinheit (Graphics Processing Unit, GPU) verfügbar ist.

Wenn sich für eine der Knotenvorrichtungen ergibt, dass dort wenigstens eine GPU für die Zuweisung einer ersten Aufgabe der Analyseroutine in Form einer ersten Taskroutine verfügbar ist (Merkmal **V3**), werden folgende Schritte unternommen:

Die erste Taskroutine, welche in einer Form zur Ausführung durch eine CPU vorliegt (Merkmal **V1**), wird analysiert, ob sie in eine GPU-Taskroutine (d.h. zur Ausführung durch eine GPU) kompiliert werden kann, so dass dort mehrere Instanzen der ersten Aufgabe der Analyseroutine wenigstens teilweise parallel durchgeführt werden können (Merkmal **V3.1**). Die Analyse umfasst zwei Kriterien:

- enthält die erste (CPU-) Taskroutine eine Anweisung, die eine Kompilierung in eine GPU-Taskroutine verhindert? (Merkmal **V3.1.1** – siehe dazu z.B. Abs. [0069], Abs. [0216] der Offenlegungsschrift)
- weisen Eingaben und Ausgaben der ersten Taskroutine Abhängigkeiten auf, die eine Kompilierung in eine GPU-Taskroutine verhindern? (Merkmal **V3.1.2** – siehe dazu z.B. Abs. [0209] / Abs. [0210])

Wenn diese Analyse ergibt, dass die vorliegende CPU-Taskroutine in eine GPU-Taskroutine kompiliert werden kann (Merkmal **V3.2**), dann wird die GPU mit der

Durchführung der Aufgabe beauftragt (vgl. z.B. Figur 20). Konkret löst ein positives Analyse-Ergebnis folgende Schritte aus (vgl. auch Figur 11A / 11B, Figur 12, Figur 15A / 15B):

Merkmal **V3.2.1 / V3.2.2**: Mittels einer Umwandlungsregel sollen bestimmte Anweisungen der CPU-Taskroutine in entsprechende GPU-Anweisungen umgewandelt und zu einer GPU-Taskroutine kompiliert werden (vgl. z.B. Abs. [0070], Abs. [0212]);

Merkmal **V3.2.3**: Der (in Merkmal **V3** bestimmten) Knotenvorrichtung wird eine Datensatzpartition (1131) des zu verarbeitenden Datensatzes (1130) zugewiesen, um der GPU den Zugriff darauf zu ermöglichen (vgl. z.B. Abs. [0236]);

Merkmal **V3.2.4**: Der Knotenvorrichtung wird die Durchführung der kompilierten GPU-Taskroutine zugewiesen, so dass die GPU diese in mehreren Instanzen parallel abarbeiten kann (vgl. z.B. Abs. [0198] / [0199] und Fig. 13 bis 15, Fig. 19 / 20).

**Im Ergebnis** ermöglicht die beanspruchte Lehre eine automatische Prüfung, inwieweit die vorgegebene Analyseroutine durch Parallelverarbeitung mittels GPUs bearbeitet werden kann, und ggf. eine automatische Kompilierung in GPU-Code. Durch die Kompilierung in einen GPU-Code wird erreicht, dass die Analyseroutine deutlich schneller und effizienter ausgeführt werden kann (vgl. Abs. [0221] / [0222] mit Tabelle der Laufzeiten), ohne dass das Programmier-Personal die Fähigkeiten oder Schulung benötigt, um die Taskroutinen selbst in einer GPU-Programmiersprache zu schreiben (siehe Abs. [0223]). Die Umwandlung in GPU-Code kann eine Übersetzung in eine andere Programmiersprache bedeuten, aber auch „nur“ eine Anpassung des Codes in derselben Programmiersprache speziell für den Ablauf auf der verwendeten GPU (vgl. z.B. Abs. [0218]).

Daraus ergibt sich als **das gelöste Problem**, einen vorgegebenen Programm-Ablauf (für aufwändige Analyseprogramme) bei einer Verteilung auf mehrere unter-



schiedliche Rechenknoten automatisch für die einzelnen Rechenknoten in einen Programm-Code zu übersetzen, welcher jeweils die vorhandenen Ressourcen des Rechenknotens bestmöglich ausnutzt.

Als **Fachmann**, der mit einer solchen Aufgabe betraut wird, sieht der Senat – in weitgehender Übereinstimmung mit den Überlegungen der Prüfungsstelle, siehe Zurückweisungsbeschluss Seite 10 oben – einen Informatiker mit Hochschulabschluss an, der über eine mehrjährige Berufserfahrung auf dem Gebiet der Programmierung für verteilte Rechen-Netze, insbesondere unter Nutzung von Grafik-Prozessoren zur Parallelverarbeitung, verfügt.

**2.** Das geltende Patentbegehren ist zulässig. Die Patentansprüche und die überarbeitete Beschreibung bleiben innerhalb des Rahmens der ursprünglichen Offenbarung. Auch andere Mängel liegen nicht vor.

**2.1** Der Senat hat die von der Anmelderin zusammen mit der Beschwerdebegründung als (damaligen) Hauptantrag eingereichte Anspruchsfassung (Ansprüche 1 bis 30), die mit der Übersetzung der Ansprüche gemäß Offenlegungsschrift übereinstimmt, im Zwischenbescheid vom 5. April 2022 als mangelbehaftet bezeichnet, weil die unter Schutz zu stellende Erfindung diesen Patentansprüchen nicht klar und deutlich entnehmbar war. Er hat festgestellt, dass die Lehre der Anmeldung und insbesondere der damals geltenden Patentansprüche an vielen Stellen ungenau, mehrdeutig oder teilweise auch gar nicht im Detail nachvollziehbar ist. Im Zwischenbescheid wurden zahlreiche Fragen dazu gestellt, wobei auf konkrete Unklarheiten und Widersprüche hingewiesen wurde.

Denn im Patent-Erteilungsverfahren ist für Patentansprüche zu sorgen, die die unter Schutz gestellte Erfindung klar und deutlich umschreiben (BGH GRUR 1988, 757 – *Düngerstreuer*, Abschnitt V Absatz 3) und geeignet sind, den Anmeldegegenstand eindeutig zu kennzeichnen und vom Stand der Technik abzugrenzen (BGH GRUR 1979, 461 – *Farbbildröhre*, II. 2d). Der Schutzbereich muss, ggf.

unter Auslegung mittels der Beschreibung, so klar und eindeutig definiert sein, dass er „für Außenstehende hinreichend sicher vorhersehbar ist“ (BGH GRUR 1989, 903 – *Batteriekastenschnur*, III.2.). Die Beseitigung vermeidbarer Unklarheiten hat im Prüfungsverfahren zu erfolgen (BGH GRUR 2013, 1210 – *Dipeptidyl-Peptidase-Inhibitoren*, III. 1a)).

Die Anmelderin hat daraufhin die Patentansprüche überarbeitet und eine neue Anspruchsfassung vorgelegt.

**2.2** Den mit der Anmeldung ursprünglich formulierten Patentansprüchen kommt im Rahmen des Erteilungsverfahrens keine eine weitergehende Offenbarung in der Beschreibung einschränkende Bedeutung zu (BGH GRUR 2005, 1023 – *Einkaufswagen II*). Vielmehr ist es (lediglich) erforderlich, dass der Fachmann die im Anspruch bezeichnete technische Lehre den Ursprungsunterlagen unmittelbar und eindeutig als mögliche Ausführungsform der Erfindung entnehmen kann; dabei ist eine unangemessene Beschränkung des Anmelders bei der Ausschöpfung des Offenbarungsgehalts der Anmeldung zu vermeiden (BGH GRUR 2015, 976 – *Einspritzventil*, Rn 45)

Die vorliegende Patentanmeldung erfolgte in englischer Sprache. Als Ursprungsunterlagen sind die am Anmeldetag eingereichten englischsprachigen Unterlagen anzusehen. Eine nachträgliche Korrektur von Übersetzungsfehlern ist im Patenterteilungsverfahren jederzeit zulässig (vgl. Schulte, PatG, 11. Auflage (2022), § 35a Rdnr. 21).

Zum Beleg der ursprünglichen Offenbarung wird nachfolgend auf die am Anmeldetag eingereichten englischsprachigen Patentansprüche (Claims 1-30 vom 31. Juli 2017) und die englischsprachige Beschreibung (Seiten 1 bis 99 vom 31. Juli 2017) Bezug genommen.

### 2.3 Zum Hauptanspruch

Der geltende Hauptanspruch geht zurück auf den ursprünglichen Claim 21. Die Unterschiede der einzelnen Merkmale des neuen Hauptanspruchs zum jeweils entsprechenden Merkmal des ursprünglichen Claims 21 sind im Folgenden markiert.

**Merkmal V0:** Computerimplementiertes Verfahren zur wenigstens teilweise parallelen Durchführung von Aufgaben von Analyseroutinen mittels einer Mehrzahl von Knotenvorrichtungen (1500), umfassend:

- Bezüglich der eingefügten Zweckangabe siehe z.B. Claim 21 „operation of the plurality of node devices is coordinated to perform tasks of analysis routines at least partially in parallel“.

**Merkmal V1:** Empfangen einer Analyseroutine, die ausführbare Anweisungen für eine oder mehrere Aufgaben in Form von Taskroutinen zur Ausführung durch wenigstens eine zentrale Verarbeitungseinheit (Central Processing Unit, CPU) einer Knotenvorrichtung umfasst, die mit wenigstens einem Datensatz durchgeführt werden sollen,

- neues Merkmal, basierend z.B. auf Beschreibung Abs. [0065] „The coordinating device may also receive an analysis routine that includes executable instructions for multiple task routines for multiple tasks to be performed with at least one data set“, i.V.m. Claim 21 „the analysis routine is generated for execution by at least one central processing unit (CPU) of the at least one node“ / „the first task routine is generated for execution by the at least one CPU to perform the first task of the analysis routine“.

**Merkmal V2:** Analysieren eines aktuellen Status von Ressourcen von wenigstens einer Knotenvorrichtung ~~einer~~ der Mehrzahl von Knotenvorrichtungen, um eine Verfügbarkeit von wenigstens einer Grafikverarbeitungseinheit (Graphics

Processing Unit, GPU) der wenigstens einen Knotenvorrichtung zu bestimmen, ~~die zugewiesen werden soll, um eine erste Aufgabe einer Analyseroutine durchzuführen, wobei~~  
~~der Betrieb der Mehrzahl von Knotenvorrichtungen koordiniert ist, um Aufgaben von Analyseroutinen wenigstens teilweise parallel durchzuführen;~~  
~~die Analyseroutine zur Ausführung durch wenigstens eine zentrale Verarbeitungseinheit (Central Processing Unit, CPU) des wenigstens einen Knotens erzeugt wird; und~~

- redaktionelle Anpassung / Streichungen (die hier gestrichene Lehre findet sich nun z.T. im neuen Merkmal **V1**)

**Merkmal V2.1:** ~~wobei die genannten Ressourcen der wenigstens einen Knotenvorrichtung aus einer Gruppe bestehend aus der wenigstens~~ zumindest ~~einen CPU der wenigstens einen GPU und Speicherplatz innerhalb wenigstens eines Speichers der wenigstens einen Knotenvorrichtung ausgewählt werden~~ und zumindest eine GPU umfassen, und

- Vereinfachung / Reduzierung des ursprünglichen „selected from a group consisting of“ auf das hier Notwendige (zum „Speicherplatz“ siehe neuen Unteranspruch 2)

**Merkmal V3:** ~~in Reaktion auf eine Bestimmung, dass die wenigstens eine GPU verfügbar ist, um zugewiesen zu werden, die erste Aufgabe der Analyseroutine durchzuführen;~~ wenn das Bestimmen ergibt, dass wenigstens eine GPU der wenigstens einen Knotenvorrichtung für die Zuweisung einer ersten Aufgabe der Analyseroutine in Form einer ersten Taskroutine verfügbar ist,

- klarstellende Umformulierung des ursprünglichen Merkmals, siehe Claim 21: „in response to a determination that the at least one GPU is available to be assigned to perform the first task of the analysis routine“; vgl. auch Beschreibung Abs. [00260] „At 2120, the processor of the coordinating

device may check whether there are any GPUs indicated in the node statuses as being sufficiently available within any of the node devices such that a task could be assigned to those node devices to be performed by such available GPUs.“

**Merkmal V3.1:** ~~Analysieren einer der ersten Taskroutine der Analyseroutine, um zu bestimmen, ob diese in erste Taskroutine kompiliert werden kann, um eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU zu erzeugen, um zu veranlassen, dass die wenigstens eine GPU mehrere Instanzen der ersten Aufgabe der Analyseroutine wenigstens teilweise parallel ohne eine Abhängigkeit zwischen Eingaben und Ausgaben der mehreren Instanzen der ersten Aufgabe durchführt, wobei: kompiliert werden kann, so dass die wenigstens eine GPU mehrere Instanzen der ersten Aufgabe der Analyseroutine wenigstens teilweise parallel durchführen kann,~~  
~~die erste Taskroutine zur Ausführung durch die wenigstens eine GPU erzeugt wird, um die erste Aufgabe der Analyseroutine durchzuführen; und~~

- klarstellende Umformulierung des ursprünglichen Merkmals, siehe Claim 21: „analyzing a first task routine of the analysis routine to determine whether the first task routine is able to be compiled to generate a GPU task routine for execution by the at least one GPU to cause the at least one GPU to perform multiple instances of the first task of the analysis routine at least partially in parallel“.

**Merkmal V3.1.1:** ~~wobei die Bestimmung, ob die erste Taskroutine kompiliert werden kann, um die GPU-Taskroutine zu erzeugen, eine Bestimmung, das Analysieren umfasst, ob die erste Taskroutine eine Anweisung umfasst, die die eine Kompilierung zur Erzeugung einer GPU-Taskroutine verhindert, um die GPU-Taskroutine zu erzeugen,~~

- klarstellende Vereinfachung des ursprünglichen Merkmals, siehe Claim 21: „the determination of whether the first task routine is able to be compiled to generate the GPU task routine comprises a determination of whether the first task routine includes an instruction that prevents the compilation to generate the GPU task routine“.

**Merkmal V3.1.2:** ~~und eine Bestimmung, ob Eingaben und Ausgaben der ersten Taskroutine definiert sind, um die Abhängigkeit nicht zu erfordern, umfasst; und Abhängigkeiten aufweisen, die eine Kompilierung zur Erzeugung einer GPU-Taskroutine verhindern,~~

- klarstellende Vereinfachung des ursprünglichen Merkmals, siehe Claim 21: „and a determination of whether inputs and outputs of the first task routine are defined to not require the dependency“ i.V.m. dem ursprünglichen Wortlaut des Merkmals **V3.1.1**.

**Merkmal V3.2:** ~~in Reaktion auf eine Bestimmung, dass die erste Taskroutine kompiliert werden kann, um die GPU-Taskroutine zu erzeugen; wenn das Analysieren ergibt, dass die erste Taskroutine in eine GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU kompiliert werden kann,~~

- klarstellende Umformulierung des ursprünglichen Merkmals, siehe Claim 21: „in response to a determination that the first task routine is able to be compiled to generate the GPU task routine“.

**Merkmale V3.2.1 bis V3.2.4:**

~~**V3.2.3** Zuweisen einer Datensatzpartition einer Mehrzahl von Datensatzpartitionen eines Datensatzes an die wenigstens eine Knotenvorrichtung, um Zugriff auf die Datensatzpartition durch die wenigstens eine GPU zu ermöglichen~~  
[Merkmal wurde um zwei Absätze nach hinten verschoben]

- V3.2.1** Verwenden einer Umwandlungsregel, um wenigstens eine Anweisung der ersten Taskroutine in wenigstens eine entsprechende Anweisung der GPU-Taskroutine umzuwandeln;
- V3.2.2** Kompilieren der wenigstens einen entsprechenden Anweisung der GPU-Taskroutine zur Ausführung durch die wenigstens eine GPU; ~~und~~
- V3.2.3** Zuweisen einer Datensatzpartition ~~einer Mehrzahl von Datensatzpartitionen eines~~ des Datensatzes an die wenigstens eine Knotenvorrichtung, um Zugriff auf die Datensatzpartition durch die wenigstens eine GPU zu ermöglichen, und
- V3.2.4** Zuweisen einer Durchführung ~~der ersten Aufgabe der Analyseroutine mit der Datensatzpartition~~ der GPU-Taskroutine an die wenigstens eine Knotenvorrichtung, um die Durchführung der mehreren Instanzen der ersten Aufgabe mit der Datensatzpartition durch die wenigstens eine GPU mittels der GPU-Taskroutine zu ermöglichen.

- kleinere Korrekturen zur Klarstellung; die Verschiebung des Merkmals **V3.2.3** nach hinten bedeutet keine Änderung der Reihenfolge, sondern macht lediglich den Zusammenhang mit der Durchführung der GPU-Taskroutine (Merkmal **V3.2.4**) deutlicher.

Zusammenfassend behebt die geltende Fassung des Anspruchs 1 Unstimmigkeiten und Ungenauigkeiten der ursprünglichen Fassung, stellt letztlich aber nur eine etwas andere Übersetzung der englischen Anspruchsfassung dar, ohne den Schutzbereich inhaltlich zu verändern. Die ursprüngliche Offenbarung lässt sich, wie ausgeführt, nachvollziehbar belegen.

## 2.4 Zu den Unteransprüchen 2 bis 11

Der **Anspruch 2** ist neu, er entspricht der dritten Alternative des ursprünglichen Claims 21: „wherein ... the resources of the at least one node device are selected from a group consisting of the at least one CPU, the at least one GPU, and storage space within at least one storage of the at least one node device“. In den folgenden Unteransprüchen wurden die Rückbeziehungen entsprechend angepasst.

Die **Ansprüche 3 bis 5** entsprechen den ursprünglichen Claims 22, 23 und den ersten zwei Absätzen des Claim 24, wobei lediglich Klarstellungen in der Übersetzung vorgenommen wurden und der Bezug auf Teilmerkmale des Anspruchs 1 genauer an diesen angepasst wurde.

Der neue **Anspruch 6** greift den zweiten Teil des ursprünglichen Claims 24 auf, welcher sich mit der zum Merkmal **V3** des Anspruchs 1 entgegengesetzten Alternative befasst, dass nämlich keine GPU für die Zuweisung einer ersten Aufgabe verfügbar ist. Die dafür beanspruchten Maßnahmen, unter Klarstellung der dafür vorgesehenen Bedingung, entsprechen dem ursprünglichen Claim 24, lediglich mit kleineren Anpassungen.

Die **Ansprüche 7 bis 11** entsprechen den ursprünglichen Claims 25 bis 27 und 29, 30. Im Anspruch 8 wurde der Ausdruck „in a manner in which“ des Claims 26 genauer übersetzt („um eine Beschränkung der Art und Weise zu identifizieren, wie...“ / zur Offenbarung siehe Beschreibung Absatz [00208]: „indications of various restrictions on the manner in which the data set 1130 may be divided“), und im Anspruch 10 ebenso der Ausdruck „wherein the second task routine is generated for execution by the at least one CPU“ des Claims 29 („wobei die zweite Task-routine zur Ausführung durch die wenigstens eine CPU erzeugt wurde“ / zur Offenbarung vgl. Beschreibung Absatz [00212] „compiling instructions of task routines 1211 originally generated for execution by the CPUs 1550 into instructions generated for execution by the GPUs 1650“, und den geltenden Anspruch 1 Merkmal **V1** „in Form von Taskroutinen zur Ausführung durch wenigstens eine zentrale



Verarbeitungseinheit (Central Processing Unit, CPU“). Darüber hinaus wurden (lediglich) kleinere sprachliche Anpassungen vorgenommen.

Damit ist auch für diese Patentansprüche die ursprüngliche Offenbarung gegeben.

## **2.5** Zu den Nebenansprüchen 12 und 13 sowie Unteranspruch 14

Der formal nebengeordnete **Anspruch 12** ist auf ein „Computerprogrammprodukt“ gerichtet, welches Anweisungen umfasst, die in der Lage sind, einen Prozessor zu veranlassen, Operationen gemäß einem Verfahren nach einem der Ansprüche 1 bis 11 durchzuführen. Er kann sich grundsätzlich auf den ursprünglichen Claim 11 stützen, ist aber nunmehr als abhängiger Anspruch formuliert.

Ebenso ist der formal nebengeordnete **Anspruch 13** auf eine „Vorrichtung“ gerichtet, umfassend einen Prozessor und einen Speicher, um Anweisungen zu speichern, die, wenn sie durch den Prozessor ausgeführt werden, den Prozessor veranlassen, Operationen gemäß einem Verfahren nach einem der Ansprüche 1 bis 11 durchzuführen. Dies entspricht dem Oberbegriff des ursprünglichen Claims 1, wobei die dortigen Verfahrensmerkmale sich jetzt durch den Rückbezug auf die Verfahrensansprüche ergeben.

Der neue **Anspruch 14** bezieht sich zurück auf den Vorrichtungsanspruch 13. Er basiert auf dem ursprünglichen Claim 8, insbesondere dessen letztem Teilmerkmal „to analyze ...“, wurde aber sprachlich angepasst und präzisiert.

Daher bestehen auch hier an der ursprünglichen Offenbarung keine Zweifel.

## **2.6** Die Beschreibung wurde unter Berücksichtigung des entgegengehaltenen Standes der Technik in zulässiger Weise daran angepasst. Insbesondere wurden die ursprünglichen Figuren 1 bis 10 und die zugehörige Beschreibung gestrichen, weil sie sich auf andere Aspekte der Datenverarbeitung mit einem Netz von Rechenknoten beziehen, jedoch nichts zu der anspruchsgemäßen Lehre

beitragen. Ferner wurden Ungenauigkeiten in der Übersetzung und offensichtliche Fehler wie z.B. falsche Bezugszeichen korrigiert. Ein offensichtlicher Schreibfehler auf Seite 1 der geltenden Beschreibung (vorletzte Zeile: „behandle twerden“ » „behandelt werden“) wurde in Abstimmung mit der Anmelderin nachträglich noch korrigiert.

3. Der Gegenstand des geltenden Patentanspruchs 1 ist durch den bekannt gewordenen Stand der Technik weder vorweggenommen noch nahegelegt.

3.1 Im Prüfungsverfahren wurden entgegengehalten:

**D1** Suchard, M.A. et al.: Understanding GPU Programming for Statistical Computation: studies in Massively Parallel Massive Mixtures. PMID: PMC2945379 in <<https://www.ncbi.nlm.nih.gov/pmc/>> am 25.9.2010

**D2** Hong, C. et al.: MapCG: Writing Parallel Program Portable between CPU and GPU. In: Proceedings of the PACT'10, September 11 – 15, Vienna, Austria, 2010, pp. 217 - 226

Die Druckschrift **D1** befasst sich mit der Frage, wie Graphische Prozessoren (GPUs) für die Berechnung von Statistik-Programmen (welche als die „Analyse-routinen“ der Anmeldung verstanden werden können) zur Parallelverarbeitung eingesetzt werden können. Die Prüfungsstelle verweist insbesondere auf die Abschnitte 3.1.2 und 3.1.3. Dort wird die Benutzung von Rechner-Clustern, GPUs und Mehrkern-CPU's (auch in einem „embarrassingly parallel“-Modus) beschrieben. Ausgehend von den besonderen Eigenschaften von GPUs (siehe z.B. Abschnitt 3.2.4) wird eine spezielle Kodierung und Kompilierung für diese speziellen Prozessoren empfohlen (insbesondere mittels des CUDA-System Development Kit von NVIDIA, siehe Abschnitt 3.2). Eine beispielhafte Vorgehensweise (Seite 8 Mitte „... the following basic steps:“) schließt das Zuweisen von Speicher, auf den eine CPU und eine GPU gemeinsam zugreifen können, mit ein. Jedoch findet sich

nirgendwo eine Anregung, „gleichzeitig“ mit unterschiedlichen Rechenknoten mit oder ohne GPU zu arbeiten und zunächst zu prüfen, ob eine GPU verfügbar ist, um dann ggf. den Programmcode der Analyseroutine automatisch in einen GPU-Code umzusetzen (vgl. insbes. Anspruchsmerkmale **V2** bis **V3.2.2**).

Die Druckschrift **D2** greift den offensichtlichen Nachteil auf, dass Prozessor-spezifischer Programmcode für spezielle GPUs wartungsintensiv ist und für jeden Prozessor-Typ individuell geschrieben und gepflegt werden muss. Als Verbesserung wird ein „Framework“ beschrieben (Seite 218 linke Spalte unten: „MapCG, a framework which offers source code level portability“), das dem Programmierer eine C-ähnliche Hochsprache anbietet, die dann mit Hilfe von runtime-Vorgaben automatisch für unterschiedliche Prozessoren übersetzt werden kann. Im Abschnitt 5.2 auf Seite 225 wird ausgeführt, dass drei Übersetzungsarten für das Hochsprache-Programm angeboten werden: CPU+GPU co-processing, CPU-only und GPU-only. Dabei habe es sich aber gezeigt, dass der CPU+GPU co-processing Modus aus verschiedenen angegebenen Gründen nachteilig sei. Zwar verweist die Prüfungsstelle auf Seite 217 linke Spalte Absatz 2: „We believe it is desired to have a programming model which provides source code portability between CPUs and GPUs, and different GPUs“. Irgendeine Anregung, die Verfügbarkeit von GPUs zu prüfen (Anspruchsmerkmal **V2**) und daraufhin ggf. die Übersetzung des Programm-Codes automatisch an eine gerade verfügbare GPU anzupassen, wenn der Programm-Code dies zulässt (Anspruchsmerkmale **V3** bis **V3.2**), gibt diese Druckschrift aber nicht.

**3.2** Mit dem Zwischenbescheid vom 5. April 2022 hat der Senat nachträglich noch die folgende Druckschrift ins Verfahren eingeführt, die aus dem parallelen britischen Prüfungsverfahren bekannt geworden ist:

**D3** US 2011 / 252 411 A1

Die Druckschrift **D3** geht aus von einem Netz von Rechenknoten, die unterschiedliche Ressourcen zur Bearbeitung von Programmen bereitstellen (Abs. [0018] / Figur 1: client 110, server 120 mit GPU 130 und/oder CPU 140, verbunden über Netzwerk 150). Gemäß Abs. [0020] empfängt (oder erzeugt) der Client 110 irgendeinen „technischen“ Computerprogramm-Code (TCE-Code 530, siehe Abs. [0047] – der Ausdruck **TCE** steht für „Technical Computing Environment“, vgl. insbesondere Abs. [0016], Abs. [0033] bis [0035], Abs. [0040]). Insoweit entspricht die Ausgangssituation derjenigen der Anmeldung.

Der TCE-Code 530 kann Abschnitte enthalten, die effizienter von einer CPU ausgeführt werden, und Abschnitte, die effizienter von einer GPU ausgeführt werden (siehe Abs. [0047] Satz 2). Ein „code type determiner 500“ ist vorgesehen, um eine entsprechende Entscheidung zu treffen und den TCE-Code 530 in entsprechende Abschnitte aufzuteilen, die dann entweder einem GPU-Compiler 510 oder einem CPU-Compiler 520 zur Übersetzung zugewiesen werden; der Code wird dann entweder einer GPU 130 oder einer CPU 140 zur Bearbeitung zugewiesen (Figur 5, Abs. [0049] bis [0054] – vgl. auch Figur 8 und zugeh. Beschreibung).

Im Unterschied zur Anmeldung wird nach der Lehre der Druckschrift **D3** nicht überprüft, ob eine Kompilierung in GPU-Code „möglich“ ist (vgl. insbes. Anspruchsmerkmale **V3.1** bis **V3.2**), sondern es steht gleich die Frage nach der Effizienz im Vordergrund; dafür sind andere Kriterien vorgesehen als in der Anmeldung (vgl. Abs. [0047] / [0048]: input size/type information 540; size threshold, degree of parallelism). Ferner ist in **D3** keine Analyse der Verfügbarkeit beschrieben (Anspruchsmerkmal **V2**).

**3.3** Demnach nimmt keine der in Betracht gezogenen Druckschriften die Lehre des geltenden Patentanspruchs 1 vorweg, und es sind auch keine Anregungen erkennbar, die in naheliegender Weise zu seiner Lehre führen würden.

4. Der geltende Patentanspruch 1 ist sonach gewährbar. Seine Unteransprüche 2 bis 11, die formal nebengeordneten, jedoch auf ein Verfahren nach einem der Ansprüche 1 bis 11 zurückbezogenen Patentansprüche 12 und 13 sowie der auf Anspruch 13 bezogene Unteranspruch 14 sind in Verbindung mit Anspruch 1 ebenfalls gewährbar.

Das Patent war daher so wie nunmehr beantragt zu erteilen.

Dr. Morawek

Baumgardt

Hoffmann

Akintche