



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 10 2008 040 009 A1 2010.03.04**

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2008 040 009.2**

(22) Anmeldetag: **27.08.2008**

(43) Offenlegungstag: **04.03.2010**

(51) Int Cl.⁸: **G06F 15/16 (2006.01)**

G06F 15/177 (2006.01)

G06F 19/00 (2006.01)

(71) Anmelder:

Siemens Aktiengesellschaft, 80333 München, DE

(72) Erfinder:

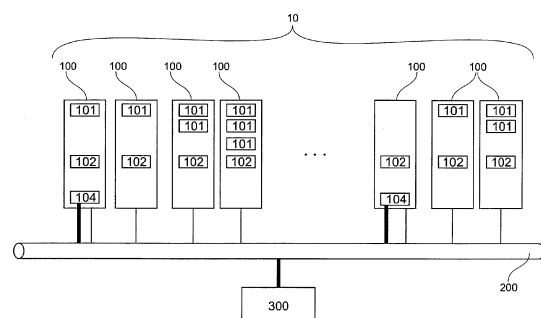
Hofmann, Ralf, 90617 Puschendorf, DE; Schülke, Andreas, 91074 Herzogenaurach, DE; Siwick, Andreas, 91054 Erlangen, DE; von Stockhausen, Hans-Martin, Dr., 91058 Erlangen, DE

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Lastverteiltes Zuweisen von medizinischen Taskflows auf Server einer Serverfarm**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren, ein System und ein Computerprogramm zum lastverteilten Zuweisen von computergestützten medizinischen Taskflows auf zumindest einem Anwendungsserver (100) einer Serverfarm (10). Dazu werden Anforderungsbedingungen und Lastinformationen in einer Konfigurationsphase konfiguriert. In einer Lastverteilungsphase werden dann die Anforderungsbedingungen erfasst. Darüber hinaus wird die Lastinformation über Lastberechnungsagenten (102) erfasst. Daraufhin kann ein Lastverteilungsdienst (104) einen Ziel-Anwendungsserver berechnen, der gemäß den ermittelten Lastinformationen alle erfassten Anforderungsbedingungen erfüllt.



Beschreibung

[0001] Die Erfindung liegt auf dem Gebiet der Medizintechnik und betrifft insbesondere eine Lastverteilung für komplexe medizinische Taskflows innerhalb einer Serverfarm.

[0002] Der klinische Alltag basiert heutzutage in der Regel auf einer Gruppe von Servern, auf die eine Gruppe von Clients zugreift. Das Client-Server-System dient zur Verarbeitung von komplexen medizinischen Taskflows. Dabei handelt es sich beispielsweise um modular aufgebaute und/oder verschachtelte Arbeitsabläufe, die beispielsweise eine Kernspintuntersuchung, eine Laboruntersuchung und ein anschließendes Post-Processing der Daten umfassen. Dabei kann die Untersuchung auf dem Magnetresonanztomographen verschiedene Sub-Tasks umfassen, wie beispielsweise das Erzeugen einer geeigneten Messesequenz für die auszuführende Untersuchung und das Anpassen der jeweiligen Messesequenz auf den MR-Scanner. Um den klinischen Betrieb möglichst effizient ausführen zu können, ist es wichtig die auszuführenden Tasks auf die jeweils dafür geeigneten Server zuzuweisen. Insbesondere müssen die Server über die geeigneten technischen Voraussetzungen verfügen, um den Task überhaupt ausführen zu können.

[0003] Aus der Informatik sind eine Vielzahl von Verfahren auf dem Gebiet der Lastberechnung beziehungsweise des Loadbalancing bekannt. Dabei sind grundsätzlich zwei unterschiedliche Ansätze zu unterscheiden: Zum einen gibt es software-basierte und hardware-basierte Lastverteilungssysteme. Als Beispiel für einen software-basierten Ansatz sei auf das Lastverteilungssystem der Firma Microsoft verwiesen, das sogenannte Microsoft Network Loadbalancing (NLB). Eingehende zu verarbeitende Datenpakete werden dabei an alle grundsätzlich verfügbaren Servermaschinen weitergeleitet. Dann wird geprüft, welche Servermaschine für den jeweiligen Dienst bzw. für das jeweilige Paket am Besten geeignet ist. Diese Servermaschine wird dann identifiziert und zum Verarbeiten des jeweiligen Datenpaketes vorbereitet. Auf allen anderen Servermaschinen wird das empfangene Datenpaket gelöscht. Dieses Vorgehen führt zu einem relativ hohen Datenverkehr.

[0004] Ein Beispiel einer hardware-basierten Lastverteilung ist das System „NitroSwitch“ der Firma NENTEC Netzwerktechnologie GmbH. Im Unterschied zu dem eben vorgestellten System der Firma Microsoft wird hier das zu verarbeitende Datenpaket nur an die Servermaschine weitergeleitet, auf der letztendlich die Verarbeitung stattfinden soll.

[0005] In der Informatik sind darüber hinaus eine Vielzahl von Algorithmen bekannt, nach denen eine Lastverteilung gesteuert werden kann. Hierzu zählen

beispielsweise das sogenannte Weighted-Round-Robin-Verfahren, das Weighted-Least-Connections-Verfahren, das Weighted-Least-Traffic-Verfahren etc. Bei der Lastverteilung nach dem NitroSwitch kann bestimmt werden, welcher der vordefinierten Algorithmen in dem speziellen Fall zur Anwendung kommen soll.

[0006] Neben den beiden vorstehend erwähnten Vorschlägen aus dem Stand der Technik zur Lastverteilung aus der Informatik ist noch eine Vielzahl von weiteren Vorschlägen bekannt. Bei den bekannten Verfahren sind die Kriterien, aufgrund deren eine Lastverteilung erfolgt jedoch meistens hardware-basiert und vorgegeben und passen nicht auf die spezifischen Charakteristika eines medizintechnischen Taskflows.

[0007] Ein weiteres bekanntes System aus dem Stand der Technik ist in der US 7,127,716 der Firma Hewlett-Packard Development offenbart. Dieses Verfahren basiert auf einer latenzzeitbasierten Lastverteilung und setzt ein komplexes Verwaltungssystem für Arbeitsabläufe voraus. Vor der Ausführung eines bestimmten Auftrages wird das Serversystem erst im Hinblick auf den jeweils auszuführenden Auftrag kalibriert. Dabei wird im Vorfeld die anfallende Arbeitslast berechnet und im Hinblick auf die verfügbaren Server so verteilt, dass insgesamt die Latenzzeit minimiert werden kann. Das hier vorgeschlagene Verfahren ist aus mehreren Gründen für die Anwendung in einem klinischen bzw. medizintechnischen Kontext nicht erfolgversprechend, da hier medizin-spezifische und klinische Kriterien berücksichtigt werden müssen, um eine optimale Lastverteilung erreichen zu können.

[0008] Aufgabe der folgenden Erfindung ist es daher, einen Weg aufzuzeigen, mit dem ein komplexer medizinischer Taskflow auf mehrere Server einer Serverfarm verbessert aufgeteilt werden kann. Darüber hinaus soll es möglich sein, unterschiedlichen Versionen von Applikationen unter Berücksichtigung von serverspezifischen Merkmalen und taskflow-spezifischen Anforderungen auf dem Gebiet der Medizintechnik in die Lastverteilung einzubinden.

[0009] Diese Aufgabe wird durch die beiliegenden Hauptansprüche gelöst.

[0010] Nachstehend wird die Lösung der Aufgabe in Bezug auf das Verfahren beschrieben. Hierbei erwähnte Merkmale, Vorteile oder alternative Ausführungsformen sind ebenso auch auf die anderen beanspruchten Gegenstände zu übertragen und umgekehrt. Mit anderen Worten können auch die gegenständlichen Ansprüche (die beispielsweise auf ein System oder ein Produkt gerichtet sind) mit den Merkmalen, die in Zusammenhang mit dem Verfahren beschrieben oder beansprucht sind, weitergebildet und umgekehrt sein. Die entsprechenden funktio-

nen Merkmale des Verfahrens werden dabei durch entsprechende gegenständliche Module, insbesondere durch Hardware-Module, des Systems bzw. der Vorrichtung ausgebildet.

[0011] Die Aufgabe wird insbesondere gelöst durch ein Verfahren zum lastverteilten Zuweisen von computer-gestützten medizinischen Taskflows auf zumindest einen Anwendungsserver einer Serverfarm, wobei das Verfahren folgende Schritte umfasst:

- Erfassen eines medizinischen Taskflows mit konfigurierbaren Anforderungsbedingungen
- Erfassen von Lastinformationen von verfügbaren Anwendungsservern der Serverfarm, wobei es – in der Regel im Vorfeld – konfigurierbar ist, welche Lastinformationen erfasst werden sollen, in dem auf ausgewählten oder auf allen Anwendungsservern zumindest eine Programmiererweiterung installiert ist.
- Ermitteln zumindest eines Ziel-Anwendungsservers, wobei zumindest ein solcher Anwendungsserver aus der Serverfarm als Ziel-Anwendungsserver bestimmt wird, der gemäß den erfassten Lastinformationen die erfassten Anforderungsbedingungen erfüllt.

[0012] Im Folgenden werden die im Rahmen dieser Anmeldung verwendeten Begrifflichkeiten näher erläutert.

[0013] Unter dem Begriff „lastverteiltes Zuweisen“ ist ein Zuordnen von Taskflows auf einzelne oder eine Gruppe von Servern einer Serverfarm zu verstehen, wobei die aktuelle Auslastung der Server innerhalb des Netzwerkes berücksichtigt werden soll. Dabei sollen konfigurierbare Kriterien zur Anwendung kommen, die insbesondere auf dem Gebiet der Medizintechnik von Relevanz sind.

[0014] Bei dem Taskflow handelt es sich üblicherweise um einen medizinischen Taskflow, der aus einer Abfolge von mehreren zeitlich versetzt ablaufenden und/oder ineinander verschachtelten Tasks besteht. Die Tasks basieren auf unterschiedlichen Applikationen, die ihrerseits in unterschiedlichen Versionen in das Netzwerk eingebunden sein können.

[0015] Die Server der Serverfarm sind über ein Computernetzwerk miteinander verbunden und können für spezifische medizintechnische Probleme ausgelegt sein. Dabei kann es sich beispielsweise um einen Postprocessing-Server, um spezielle 3D-Volumen-Berechnungskarten oder um MR-Scanner handeln.

[0016] Die Anforderungsbedingungen beziehen sich auf den Taskflow und sind im Hinblick auf die medizinische Fragestellung konfigurierbar. Die Anforderungsbedingungen umfassen in der Regel eine Reihe von Sub-Bedingungen, die über mehrere logi-

sche Operatoren miteinander verknüpfbar sind. Damit können sehr komplexe Merkmalsbedingungen formuliert werden, die im Bezug auf einen Taskflow grundsätzlich erfüllt sein müssen. Vorzugsweise können sowohl die Anforderungsbedingungen als auch die Taskflows gruppiert werden, sodass z. B. eine Anforderungsbedingung für eine bestimmte Gruppe von Taskflows konfigurierbar wird.

[0017] Bei den Lastinformationen handelt es sich um Daten in Bezug auf die jeweilige Last der Anwendungsserver der Serverfarm. Die Lastinformationen können sich auf einen vergangenen Zeitraum, auf den aktuellen Zeitraum und auf einen zukünftigen Zeitraum beziehen. Das Erfassen der Lastinformation wird über sogenannte Lastberechnungsagenten ausgeführt. Die Lastberechnungsagenten sind jeweils auf einem Anwendungsserver installiert und dienen dazu, last-bezogene Informationen in Bezug auf den jeweiligen Server und/oder in Bezug auf das Netzwerk (z. B. Netzwerkverbindungen) zusammen zu tragen und zu erfassen. Gemäß einer bevorzugten Ausführungsform der Erfindung ist es vorgesehen, dass es einstellbar ist, welche Lastinformationen erfasst werden sollen. Damit ist es möglich, das Verfahren zum lastverteiltem Zuweisen ganz detailliert und dezidiert auf die Anwendungssituation hin zuschneiden zu können.

[0018] Ein wesentlicher Vorteil der erfindungsgemäßen Lösung ist in der Konfigurierbarkeit der unterschiedlichen Bedingungen zu sehen. Zum einen ist es möglich, die Anforderungsbedingungen dezidiert zu konfigurieren. Durch eine Vielzahl von logischen Operatoren, die bereitgestellt werden, um bestimmte Sub-Bedingungen miteinander zu verknüpfen, sind sehr komplexe Merkmalsbedingungen als Anforderungsbedingungen formulierbar. Beispielsweise kann formuliert werden, dass eine Hochleistungsgraphikkarte mit mindestens zwei Gigabyte freiem Arbeitsspeicher erforderlich ist, um ein 3D-Rendering für den spezifischen medizinischen Taskflow durchzuführen. Zum anderen sind die Lastinformationen konfigurierbar. Mit anderen Worten ist es möglich, in jedem Anwendungsfall zu bestimmen, welche Lastinformationen erfasst werden sollen und als relevant gelten. In einem Fall kann es beispielsweise relevant sein, wie viel Arbeitsspeicher zur Speicherung von komplexen Bilddatensätzen auf einen Anwendungsserver vorhanden ist, während in einem zweiten Fall dieses Merkmal unerheblich ist und es vielmehr auf die jeweilige Rechenleistung des Anwendungsservers ankommt, um beispielsweise ein kompliziertes Post-Processing durchzuführen.

[0019] Gemäß einem Aspekt der vorliegenden Erfindung ist es vorgesehen, dass alle oder einzelne der vorstehend erwähnten Verfahrensschritte automatisch ausgeführt werden. So ist es in dieser Ausführungsform insbesondere vorgesehen, dass das

Ermitteln eines Ziel-Anwendungsservers ohne jegliche Benutzerinteraktion bestimmt wird. Damit wird es möglich, das Lastverteilen wesentlich effizienter und gezielter ausführen zu können und Fehler zu vermeiden, die dadurch entstehen, dass ein nicht geeigneter Anwendungsserver für einen bestimmten Taskflow manuell bestimmt worden ist.

[0020] Das erfindungsgemäße Verfahren kann in zwei Zeitphasen gegliedert sein: Zum einen gibt es eine Konfigurationsphase, in der die Anforderungsbedingungen konfiguriert werden können und/oder in der die Lastinformation konfiguriert werden kann. Zum anderen gibt es eine Lastverteilungsphase, in der tatsächlich zumindest ein Ziel-Anwendungsserver ermittelt wird. Dabei wird auf die in der Konfigurationsphase definierten Bedingungen und Lastinformationen zurückgegriffen. Grundsätzlich sind die beiden Zeitphasen voneinander unabhängig, wobei die Lastverteilungsphase nach der Konfigurationsphase stattfinden muss.

[0021] Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung ist es vorgesehen, dass der Taskflow automatisch an den im Vorfeld ermittelten Ziel-Anwendungsserver zum Zwecke der Ausführung weitergeleitet wird. Damit wird sicher gestellt, dass keine Wartezeit überbrückt werden muss, nachdem der bestgeeignete Ziel-Anwendungsserver ermittelt worden ist und bevor der Task auf dem ermittelten Ziel-Anwendungsserver ausgeführt werden kann. Damit kann also die Effizienz des gesamten medizinischen Taskflows gesteigert werden.

[0022] Wie vorstehend bereits erwähnt, kann die Anforderungsbedingung nach komplexen Kriterien konfiguriert werden. Diese umfasst in der Regel eine Vielzahl von Sub-Bedingungen, die über mehrere logische Operatoren (UND, ODER, Negation etc.) verknüpft werden können. Darüber hinaus sind auch Verschachtelungen bzw. Kaskaden von Anforderungsbedingungen formulierbar. In einer bevorzugten Ausführungsform ist es vorgesehen, dass bestimmte Minimalanforderungen formuliert werden können, die beispielsweise in einer XML-Datei spezifiziert sein können. Darüber hinaus können Präferenzen formuliert werden, indem eine Sub-Bedingung ihrerseits gewichtet wird. Damit kann beispielsweise folgende Anforderungsbedingung formuliert werden: „Volumen-Renderingkarte oder Hochleistungsgrafikkarten, wenn möglich Volumen-Renderingkarte“. Aufgrund der definierbaren Präferenz wird in dem vorstehend erwähnten Beispiel ein Anwendungsserver, der lediglich eine Hochleistungsgrafikkarte besitzt also nur als Ausweichlösung (sozusagen als Fallback) und nur in dem Fall verwendet, wenn kein anderer Anwendungsserver mit einer Volumen-Renderingkarte verfügbar ist. Darüber hinaus können optionale Merkmale als Anforderungsbedingung formuliert werden. Beispielsweise ist es möglich folgende Be-

dingung zu formulieren: „64-Bit-Prozessor wenn möglich: mit zwei Kernen“. In diesem Beispiel wird für einen bestimmten Taskflow ein 64-Bit-Prozessor benötigt. Für eine optimale Parallelisierung der in dem Taskflow enthaltenen Tasks ist jedoch ein Zweikern-Prozessor von Vorteil. Dieses Merkmal kann mit der vorstehend formulierten Anforderungsbedingung spezifiziert werden. Die Flexibilität der Lastverteilung kann damit deutlich gesteigert werden.

[0023] Gemäß einem Aspekt der vorliegenden Erfindung umfasst eine Anforderungsbedingung grundsätzlich neben physikalischen Systemparametern auch logische Merkmale. Eine Anforderungsbedingung kann von daher ausschließlich physikalische Systemparameter betreffen, ausschließlich logische Merkmale betreffen oder eine Kombination von beidem. Als Systemparameter sind hier beispielweise eine CPU-Auslastung, eine Speicherbelegung, eine Netzwerkauslastung etc. zu nennen. Die Systemparameter beziehen sich auf physikalisch messbare Größen. Ein logisches Merkmal ist zum Beispiel „Notfallserver“. Dieses Merkmal richtet eine Suche ausschließlich auf solche Anwendungsserver, die für Notfälle reserviert bleiben sollen.

[0024] Durch das logische Verknüpfen der Vielzahl von Sub-Bedingungen können komplette Anforderungsbedingungen formuliert werden und auch Fallback-Strategien für das lastverteilte Zuweisen von Taskflows, falls der als optimal bestimmte Ziel-Anwendungsserver nicht verfügbar ist. Die Möglichkeiten zur Formulierung von Alternativstrategien sind sehr weitreichend. Damit kann vorteilhafterweise das erfindungsgemäße Verfahren sehr flexibel auf den jeweiligen Anwendungsfall angepasst werden.

[0025] Üblicherweise ist es vorgesehen, dass jeweils ein Lastberechnungsagent auf jeweils einem Anwendungsserver der Serverfarm installiert ist.

[0026] Gemäß einem weiteren Aspekt der Erfindung ist es vorgesehen, dass jeweils ein Lastberechnungsagent die Last des jeweiligen Anwendungsservers berechnet, auf dem er installiert ist. Damit ist eine 1:1-Zuordnung zwischen einem Lastberechnungsagenten und einem Anwendungsserver vorgesehen.

[0027] Grundsätzlich sollte das erfindungsgemäße Verfahren damit enden, dass zumindest ein Ziel-Anwendungsserver ermittelt wird, auf dem der jeweilige Taskflow ausgeführt werden kann. Es ist jedoch auch möglich, dass kein Ziel-Anwendungsserver verfügbar ist. In diesem Fall muss das Verfahren zu einem späteren Zeitpunkt wiederholt werden, bei dem eine andere Lastinformation gegeben ist. Ebenso ist es möglich, dass nicht nur ein Ziel-Anwendungsserver ermittelt wird, sondern eine Gruppe von grundsätzlich möglichen und verfügbaren Ziel-Anwendungsser-

vern. In diesem Fall ist es sehr hilfreich, dass zusätzlich ein Lastergebnis ausgegeben wird. Das Lastergebnis wird üblicherweise in Form einer Liste dargestellt und quantifiziert eine Menge von ermittelten Ziel-Anwendungsservern dahingehend, inwieweit sie die erfassten Anforderungsbedingungen erfüllen. So kann es beispielsweise möglich sein, dass ein erster Ziel-Anwendungsserver und ein zweiter Ziel-Anwendungsserver die ermittelten Anforderungsbedingungen zu 100% erfüllen, während weitere Anwendungsserver die ermittelten Anwendungsbedingungen nur zu 80% erfüllen. Weitere Ziel-Anwendungsserver werden dann nach ihrem Erfüllungsgrad quantifiziert. Damit wird eine weitere Information bereitgestellt, die das Ergebnis eines Matchings zwischen erfassten, konfigurierten Anforderungsbedingungen und erfassten, konfigurierten Lastinformationen kennzeichnet. Dies kann beispielsweise in Form einer Art Hitliste erfolgen.

[0028] Falls eine Gruppe von Ziel-Anwendungsservern ermittelt werden kann, ist es in einer vorteilhaften Weiterbildung der Erfindung vorgesehen, dass die ermittelten Ziel-Anwendungsserver hinsichtlich im Vorfeld konfigurierbarer Kriterien priorisiert werden. Damit ist es möglich, den optimal passenden Ziel-Anwendungsserver aus der Menge der grundsätzlich verfügbaren und passenden Anwendungsserver zu bestimmen.

[0029] Ein weiterer Flexibilitätsgrad kann dadurch erreicht werden, dass die beiden Erfassungsvorgänge grundsätzlich unabhängig voneinander ausgeführt werden können. Insgesamt kann das Erfassen der Anforderungsbedingungen im Hinblick auf den Taskflow unabhängig von dem Erfassen der Lastinformation über die Lastberechnungsagenten erfolgen. Damit können die beiden Erfassungsvorgänge voneinander entkoppelt werden. Dies ist jedoch nur ein optionales Merkmal.

[0030] Gemäß einem Aspekt der vorliegenden Erfindung ist das Verfahren computerimplementiert und kann von Software und/oder Hardwaremodulen gesteuert werden. Insbesondere ist es vorgesehen, dass in der Konfigurationsphase Benutzerinteraktionen notwendig sind, um die jeweiligen Benutzereingaben tätigen zu können. Im Unterschied dazu kann das Verfahren in der Lastverteilungsphase vollautomatisch ohne weitere Benutzerinteraktion ausgeführt werden.

[0031] Eine weitere Aufgabenlösung besteht in einem Computerprogramm, einem Computerprogrammprodukt und in einem System zum lastverteilten Zuweisen von computergestützten medizinischen Taskflows auf zumindest einen Anwendungsserver einer Serverfarm.

[0032] Das System umfasst eine Vielzahl von An-

wendungsservern einer Serverfarm, die über ein Computernetzwerk miteinander in Datenaustausch stehen. Das System umfasst ein Anforderungsmodul zum Erfassen von Anforderungsbedingungen der Taskflows. Das System umfasst darüber hinaus Lastberechnungsagenten, die dazu bestimmt sind, Lastinformationen der jeweiligen Anwendungsserver zu erfassen. Des Weiteren umfasst das System einen Lastverteilungsdienst, der dazu ausgebildet ist, einen Ziel-Anwendungsserver zu ermitteln, wobei der Lastverteilungsdienst zumindest einen Anwendungsserver aus der Serverfarm als Ziel-Anwendungsserver bestimmt, der gemäß den von den jeweiligen Lastberechnungsagenten erfassten Lastinformationen, die von dem Anforderungsmodul erfassten Anforderungsbedingungen erfüllt.

[0033] Die Lastberechnungsagenten dienen zum Berechnen der Last in Bezug auf den ihnen zugeordneten Anwendungsserver (inklusive der Netzwerkverbindungen und der Schnittstellen etc.) und sind nicht nach außen sichtbar, sondern fungieren als „Außenposten“ des Lastverteilungsdienstes. Sie hosten sämtliche Programmiererweiterungen.

[0034] Der Lastverteilungsdienst ist vorzugsweise zentral ausgebildet und dient zum Entgegennehmen des erfassten Taskflows, zum Beauftragen der jeweiligen Lastberechnungsagenten, zum Einholen bzw. Sammeln und gegebenenfalls zum Verarbeiten der berechneten Lastinformationen der Lastberechnungsagenten und zum Ermitteln eines Ziel-Anwendungsservers und gegebenenfalls zum Darstellen eines Ergebnisses. Der Lastverteilungsdienst fungiert sozusagen als Mittler, um den ermittelten Taskflow mit seinen Anforderungsbedingungen entgegenzunehmen und die Lastinformationen von den verfügbaren Anwendungsservern zu erfragen. Der Mittler beauftragt alle beteiligten Lastberechnungsagenten und sammelt deren Ergebnisse ein. Diese Ergebnisse werden dann mit den verfassten Anforderungsbedingungen für den jeweiligen Taskflow abgeglichen bzw. gematched. Der Lastverteilungsdienst ist als Schnittstelle nach außen sichtbar.

[0035] Das Anforderungsmodul ist gemäß einem Aspekt der Erfindung als XML-Datei ausgebildet, die eine entsprechende Konfiguration umfasst. Die Konfiguration kann neu erzeugt oder modifiziert werden, so dass unterschiedliche Anforderungsbedingungen konfiguriert werden können.

[0036] Das System umfasst des Weiteren eine Oberfläche, auf der ein Ergebnis des erfindungsgemäßen Verfahrens darstellbar ist. Diese Oberfläche ist vorzugsweise auf dem Lastverteilungsdienst ausgebildet und kann darüber hinaus auch entweder auf einem Lastberechnungsagenten, auf mehreren ausgewählten oder auf allen Lastberechnungsagenten ausgebildet sein. Sie dient dazu, den ermittelten

Ziel-Anwendungsserver oder die Gruppe der ermittelten Ziel-Anwendungsserver anzugeben und auf einer Oberfläche darzustellen.

[0037] Gemäß einer bevorzugten Ausführungsform sind die jeweiligen Anwendungsserver mit zumindest einer Programmerweiterung ausgebildet. Die Programmerweiterung kann in Form eines Plug-Ins ausgebildet sein, das auf dem Anwendungsserver installiert ist. Je nach Anwendungsfall ist es möglich, auf einem Anwendungsserver keine Programmerweiterung, lediglich eine Programmerweiterung oder mehrere unterschiedliche Programmerweiterungen vorzusehen. Die Programmerweiterungen dienen dazu, die Lastberechnung auf den jeweiligen Anwendungsfall hin konfigurieren bzw. auslegen zu können. Damit wird es möglich, die Lastverteilung situationsspezifisch zu konfigurieren.

[0038] Darüber hinaus ist es möglich, den Lastverteilungsdienst mit einer weiteren Funktionalität auszubilden, sodass der Mittler automatisch den Taskflow an den ermittelten Ziel-Anwendungsserver zum Zwecke der Ausführung zuweist. In diesem Fall erfolgt keine weitere Benutzerinteraktion.

[0039] In einer bevorzugten Ausführungsform der vorliegenden Erfindung umfasst das System also zwei lastberechnungs-bezogenen Instanzen: Zum einen zumindest einen Lastverteilungsdienst und zum anderen eine Vielzahl von Lastberechnungsagenten. Vorzugsweise ist auf jedem Anwendungsserver ein Lastberechnungsagent installiert. Gemäß einer ersten Variante der Erfindung ist es vorgesehen, dass nur ein Lastverteilungsdienst ausgebildet ist. Der kann auf einem Anwendungsserver installiert sein. Um die Ausfallsicherheit des gesamten Systems zu erhöhen, kann es in einer zweiten Variante der Erfindung auch vorgesehen sein, mehrere Lastverteilungsdienste vorzusehen. Sie können auf unterschiedlichen Anwendungsservern installiert sein und sind über ein Netzwerk miteinander verbunden. Ebenso kann der Lastverteilungsdienst auf einem Taskflowverwaltungssystem oder auf einer über das Netzwerk angeschlossenen Instanz ausgebildet sein.

[0040] Der Lastverteilungsdienst arbeitet nach einem abhängigen Modus. Bei dem abhängigen Modus ist es vorgesehen, dass der Lastverteilungsdienst die Auslastung des jeweiligen Anwendungsservers bzw. die Lastinformationen in Abhängigkeit von den Anforderungsbedingungen berechnet, die er von dem Taskflowverwaltungssystem erhalten hat. Sobald eine Anforderungsbedingung an einen Taskflow geknüpft ist, wird diese auch unbedingt berücksichtigt. Im anderen Fall, also wenn dem Taskflow keine Anforderungsbedingung zugeordnet ist, greifen vorkonfigurierbare Regeln des Lastverteilungsdienstes.

[0041] Mit anderen Worten ist es also vorgesehen, dass der Lastverteilungsdienst auf vorkonfigurierbare Regeln zugreift, die es ihm ermöglichen, die jeweiligen Anfragen an die Lastberechnungsagenten spezifisch zu gestalten. Mit anderen Worten kann es aufgrund der Anforderungsbedingungen sinnvoll sein, die Lastberechnungsagenten unterschiedlich anzusprechen. Abhängig von den erfassten Anforderungsbedingungen kann es beispielsweise Sinn machen, einen bestimmten Lastberechnungsagenten eines bestimmten Anwendungsservers gar nicht erst anzusprechen, weil auf ihm die angeforderte Konfiguration nicht bereit steht (beispielsweise zu wenig CPU-Ressourcen), während andere Anwendungsserver jedoch grundsätzlich die erfassten Anforderungsbedingungen erfüllen können und somit über deren Lastberechnungsagenten angesprochen werden sollen.

[0042] Ein Lastberechnungsagent kann mit Gewichtungen belegt werden, wobei die Gewichtungen jeweils auf die Relevanz des jeweiligen Lastberechnungsagenten hinweisen soll.

[0043] Nach dem Erfassen der Lastinformationen über die Vielzahl der Lastberechnungsagenten können diese je nach Anwendungsfall hin in einer Weiterbildung der Erfindung mit Gewichtungsfaktoren belegt sein und werden anschließend vom Lastverteilungsdienst erfasst. Der Lastverteilungsdienst kann dann den Ziel-Anwendungsserver ermitteln. Dies erfolgt durch ein Abgleichen der erfassten Lastinformationen mit den erfassten Anforderungsbedingungen. Bei einem erfolgreichen Matching (also die geforderten Anforderungsbedingungen könne von einem bestimmten Anwendungsserver gemäß Lastberechnungsagent erfüllt werden) wird der Taskflow auf dem jeweiligen Ziel-Anwendungsserver zum Zwecke der Ausführung weitergeleitet.

[0044] Die vorstehend beschriebenen, erfindungsgemäßen Ausführungsformen des Verfahrens können auch als Computerprogrammprodukt ausgebildet sein, wobei der Computer zur Durchführung des oben beschriebenen, erfindungsgemäßen Verfahrens veranlasst wird, wenn das Programm auf dem Computer bzw. einem Prozessor des Computers ausgeführt wird.

[0045] Eine alternative Aufgabenlösung sieht ein Speichermedium vor, das zur Speicherung des vorstehend beschriebenen, computerimplementierten Verfahrens bestimmt ist und von einem Computer lesbar ist.

[0046] Darüber hinaus ist es möglich, dass einzelne Komponenten des vorstehend beschriebenen Verfahrens oder Systems (die Lastberechnungsagenten, den Lastverteilungsdienst, das Anforderungsmodul) in einer verkaufsfähigen Einheit und die restlichen

Komponenten in einer anderen verkaufsfähigen Einheit – sozusagen als verteiltes System – ausgeführt werden können.

[0047] In der folgenden detaillierten Figurenbeschreibung werden nicht einschränkend zu verstehende Ausführungsbeispiele mit deren Merkmalen und weiteren Vorteilen anhand der Zeichnung besprochen. In dieser zeigen:

[0048] [Fig. 1](#) eine übersichtsartige Darstellung eines erfindungsgemäßen Systems gemäß einer bevorzugten Ausführungsform, und

[0049] [Fig. 2](#) ein Ablaufdiagramm gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung.

[0050] Im Zusammenhang mit [Fig. 1](#) soll der grundlegende Aufbau eines erfindungsgemäßen Systems zum lastverteilten Zuweisen von computergestützten medizinischen Taskflows auf zumindest einen Anwendungsserver **100** einer Serverfarm **10** erläutert werden.

[0051] Im klinischen Alltag umfassen medizinische Taskflows in der Regel eine Vielzahl von komplexen Tasks, die in einer bestimmten zeitlichen Abfolge ausgeführt werden müssen. Dabei können die Tasks wiederum Sub-Tasks umfassen und sie können ineinander verschachtelt oder verschränkt sein. Beispielsweise kann der Task „Untersuchungsplanung für den Patienten X“ den Task „MR-Aufnahme des Schädels“ umfassen. Dieser Task kann beispielsweise die Sub-Tasks umfassen „Generiere Messsequenz für den MR-Scanner“, „Erfasse Bilddaten“, „Postprocessing der erfassten Bilddaten“ etc.

[0052] Jeder Taskflow hat spezifischen Anforderungsbedingungen. So setzt beispielsweise der Taskflow „Generiere MR Gehirnschicht“ andere technische Voraussetzungen voraus, als der Taskflow „Untersuchung des Blutes auf bestimmte Laborwerte“. Im ersten Fall müssen andere technische Parameter erfüllt sein, um den Taskflow überhaupt zur Ausführung bringen zu können. Von daher ist es in der Praxis sehr wichtig, die Anforderungsbedingungen auch für medizintechnische Taskflows definieren und formulieren zu können. Erfindungsgemäß können deshalb die Anforderungsbedingungen im Vorfeld in einer Konfigurationsphase konfiguriert werden. Dies erfolgt üblicherweise über eine Benutzerschnittstelle, etwa in Form einer Maske auf dem Bildschirm.

[0053] Wie in [Fig. 1](#) dargestellt, erfolgt das Verwalten der klinischen Taskflows in einem Taskflowverwaltungssystem **300**. Das Taskflowverwaltungssystem **300** ist über ein Netzwerk **200** an die jeweiligen Anwendungsserver **100** angeschlossen und ebenfalls Bestandteil der Serverfarm **10**.

[0054] Erfindungsgemäß sind die Server **100** (in diesem Dokument auch Anwendungsserver genannt) mit weiteren Modulen ausgestattet, sodass einen erweiterte Funktionalität auf den Servern **100** ausgeführt werden kann.

[0055] In einer bevorzugten Ausführungsform ist jeder Server **100** mit einem Lastberechnungsagenten **102** ausgestattet, der dazu bestimmt ist, die Last des jeweiligen Servers **100** zu ermitteln bzw. zu errechnen. Grundsätzlich können auch mehrere Lastberechnungsagenten auf einem Server **100** ausgebildet sein. Dies macht beispielsweise dann Sinn, wenn die Lastberechnung eine eher komplexe Aufgabe darstellt und dafür mehr Rechenleistung notwendig ist, die dann auf mehrere Lastberechnungsagenten verteilt werden kann. Alternativ ist es auch möglich, einen Lastberechnungsagenten **102** für eine Gruppe von Servern **100** vorzusehen, wobei der jeweilige Lastberechnungsagent **102** die Lasten der ihm zugeordneten Server **100** berechnen soll.

[0056] Gemäß einem weiteren Aspekt der vorliegenden Erfindung kann (aber muss nicht) ein Server **100** zumindest eine Programmerweiterung **101** umfassen. Es kann also – wie auch in [Fig. 1](#) dargestellt – der Fall sein, dass einzelne Server **100** keine Programmerweiterung **101**, lediglich eine Programmerweiterung **101** oder auch mehrere unterschiedliche Programmerweiterungen **101** umfassen.

[0057] Des Weiteren ist es vorgesehen, dass auf zumindest einem Server **100** der Serverfarm **10** ein Lastverteilungsdienst **104** installiert ist (in [Fig. 1](#) auf zwei Anwendungsservern **100**). Er dient dazu, den entsprechenden Taskflow von den Taskflowverwaltungssystem **300** entgegenzunehmen und mit oder ohne weitere Verarbeitung an die jeweiligen Lastberechnungsagenten **102** der Server **100** der Serverfarm **10** weiterzuleiten, damit die Lastberechnungsagenten **102** dann ihre jeweilige Last berechnen und an ihn (an dem Lastverteilungsdienst **104**) zurücksenden können. Entsprechend empfängt der Lastverteilungsdienst **104** die Ergebnisse der Lastberechnungsagenten **102** und verarbeitet sie weiter, um den Ziel-Anwendungsserver zu bestimmen. Je nach Ergebnis des Verfahrens können, ein oder mehrere Ziel-Anwendungsserver ermittelt werden. Ein Ziel-Anwendungsserver kennzeichnet sich dadurch, dass er die erfassten Anforderungsbedingungen des Taskflows gemäß den aktuellen ermittelten Lastinformationen des ermittelten Servers **100** erfüllen kann.

[0058] Im Zusammenhang mit [Fig. 2](#) soll ein grundsätzlicher Ablauf gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung vorgestellt werden. Wie in [Fig. 2](#) angedeutet, kann das Verfahren grundsätzlich in zwei aufeinanderfolgende Zeitphasen eingeteilt werden. In einer ersten Konfigurationsphase werden alle zu konfigurierenden Parame-

ter konfiguriert. In einer zweiten Lastverteilungsphase wird dann die tatsächliche Last ermittelt, die der jeweilige Taskflow benötigt. Die zweite Lastverteilungsphase weist zwei weitere Sub-Phasen auf, die unabhängig voneinander zur Ausführung gebracht werden können: Eine zum Erfassen von Parametern dienende Erfassungsphase S10, S15, S20 und eine der eigentliche Lastberechnung dienende Lastberechnungsphase S25.

[0059] In einem ersten Schritt S1 werden die Anforderungsbedingungen konfiguriert. Hier kann festgelegt werden, welche Anforderungsbedingungen überhaupt zu erfassen sind.

[0060] In einem zweiten Schritt S5 werden die zu erfassenden Lastinformationen konfiguriert.

[0061] In einem Schritt S10 wird der aktuelle Taskflow erfasst, indem der Taskflow auf dem Taskflowverwaltungssystem **300** gestartet werden soll.

[0062] In einem Schritt S15 werden die Anforderungsbedingungen des Taskflows erfasst werden. Vorzugsweise werden die Anforderungsbedingungen zunächst vom Taskflowverwaltungssystem **300** an den Lastverteilungsdienst **104** gesendet.

[0063] Daraufhin können in einem Schritt S20 die Lastinformationen ermittelt werden. Dies erfolgt über die jeweiligen Lastberechnungsagenten **102**. Dazu reicht jeder der Lastberechnungsagenten **102** die jeweils erfassten Anforderungsbedingungen an alle Lastberechnungsagenten **102** weiter. Jeder der Lastberechnungsagenten **102** seinerseits leitet diese weiter an alle gehosteten Programmiererweiterungen **101**, die dann jeweils ein (Teil-)Ergebnis berechnen. Die Teilergebnisse der Programmiererweiterungen **101** werden anschließend vom zugehörigen Lastberechnungsagenten **102** konsolidiert und an den Lastverteilungsdienst **104** gesendet.

[0064] In einem letzten Schritt S25 wird dann der zumindest eine Ziel-Anwendungsserver aufgrund der erfassten Anforderungsbedingungen und aufgrund der erfassten Lastinformationen berechnet. Der Lastverteilungsdienst **104** konsolidiert dazu ebenfalls die Ergebnisse der Lastberechnungsagenten **102** und ermittelt den optimalen Anwendungsserver **100**. Danach gibt der Lastverteilungsdienst **104** das konkrete Ergebnis an das Taskflowverwaltungssystem **300** zurück, damit dieses den Taskflow auf dem ermittelten Ziel-Anwendungsserver starten kann.

[0065] Für den Fachmann liegt es auf der Hand, dass es sinnvoll ist, das Ergebnis des Lastverteilungsverfahrens anzuzeigen und bereitzustellen. Üblicherweise erfolgt dies beispielsweise auf dem Lastverteilungsdienst **104**, der eine entsprechende Be-

nutzerschnittstelle aufweist. Ebenso und alternativ ist es möglich, das Ergebnis des Verfahrens auch an das Taskflow-Managementsystem **300** weiterzuleiten.

[0066] Die Anforderungsbedingungen basieren sowohl auf physikalischen Systemparametern, als auch auf logischen Merkmalen und können in anderen Weiterbildungen noch weitere Parameter des Systems betreffen. Sie können durch logische Operatoren auf vielfältige Weise als boole'sche Konfiguration formuliert werden.

[0067] Ebenso können die Lastinformationen, die es zu erfassen gilt, konfiguriert werden. Vorzugsweise geschieht dies über eine entsprechend konfigurierte Maske. Dazu wird auf einem Server **100** die Programmiererweiterung **101** bereitgestellt. Dies erfolgt vorzugsweise in Form eines Plug-In's. Das Plug-In dient dazu, Berechnungsanfragen vom Lastberechnungsagenten **102** entgegenzunehmen. Vorzugsweise wird für diesen Zweck eine Methode „DetectFreeCapacity()“ aufgerufen. Eine Berechnungsanfrage enthält eine komplexe Merkmalsbedingung (Server Query) mit in der Regel einer Vielzahl von Anforderungsbedingungen. Die Programmiererweiterung **101** bewertet nun abhängig von der Konfiguration – alle, einige oder keine der Sub-Bedingungen der Anforderungsbedingung. Die Bewertung erfolgt auf einer normierten Skala von 0–100. Dabei bedeutet der Wert 0, dass das entsprechende Merkmal der Anforderungsbedingung vollständig ausgelastet ist und für die aktuelle Lastanfrage nicht mehr zur Verfügung steht, während der Wert 100 die vollständige Verfügbarkeit signalisieren soll. Am Beispiel der Volumenrendringkarte mit 8 GB Speicher, bedeutet also ein Wert von 25, dass von den verfügbaren 8 GB noch 2 frei sind. Nachdem alle für die Programmiererweiterung **101** relevanten Anforderungsbedingungen mit ihren Sub-Bedingungen entsprechend bewertet worden sind, gibt die Programmiererweiterung **101** eine bewertete komplexe Antwort an den Lastberechnungsagenten **102** zurück.

[0068] In bevorzugten Weiterbildungen der Erfindung ist es möglich, diese Antwort noch weiter zu verarbeiten. Insbesondere ist es möglich, die Antwort im Hinblick auf die einzelnen Anforderungsbedingungen und/oder auf die einzelnen Sub-Bedingungen einer Anforderungsbedingung auszuwerten. Mit der Methode „GetUtilisationInfo()“ wird der aktuelle Belastungszustand aller Sub-Bedingungen auf der normierten Skala von 0–100 zurückgegeben, während die Methode „GetStateInfo()“ eine beliebige XML-basierte Zeichenkette enthält, die den inneren Zustand der Programmiererweiterung **101** beschreibt. Die Zeichenkette kann ein sogenanntes XML-Stylesheet enthalten, das zur geeigneten Visualisierung der enthaltenen Informationen dient.

[0069] Darüber hinaus sind noch weitere Methoden vorgesehen, um den inneren Zustand der Programmiererweiterungen **101** zu verwalten. Dazu zählen folgenden Methoden: „HandleRequestAssigned()“, „HandleRequestNotAssigned()“, „HandleServiceStarted()“ und „HandleServiceFinished()“.

[0070] Nachdem ein Lastberechnungsagent **102** eine Methode „DetectFreeCapacity()“ an einer Programmiererweiterung **101** aufgerufen hat, erfolgen zwei weitere Protokollschritte. In einem ersten Schritt wird die Methode „HandleRequestAssigned()“ oder „HandleRequestNotAssigned()“ aufgerufen. Dies erfolgt in Abhängigkeit davon, ob der Server **100**, auf dem die jeweilige Programmiererweiterung **101** läuft, durch den Lastverteilungsdienst **104** als am Besten geeigneter Server und damit als Zielservers für den zu startenden Taskflow identifiziert worden ist oder nicht.

[0071] Demnach haben Programmiererweiterungen **101** auf einem Server **100** die Möglichkeit, bestimmte Einstellungen und Konfigurationen im Vorfeld zu treffen. So ist es beispielweise möglich, dass die Programmiererweiterung **101** reservierte Kontingente für bestimmte Anforderungsbedingungen als belegt markiert oder entsprechend wieder frei gibt. Als zweiter Schritt wird an allen Programmiererweiterungen **101**, die auf den ausgewählten Ziel-Anwendungsservern laufen, eine Methode „HandleServiceStarted()“ aufgerufen. Analog dazu erfolgt der Aufruf der Methode der „HandleServiceFinished()“ nachdem der Taskflow beendet worden ist. Auch diese Methoden dienen der Akquirierung bzw. der Freigabe von Kontingenten von bestimmten geforderten Anforderungsbedingungen.

[0072] Eine Klasse „Clustermanagement“ (die dem Lastverteilungsdienst **104** zugeordnet ist) bietet die Methode „GetServer()“ um den Ziel-Anwendungsserver für eine spezifische Anforderungsbedingung (ServerQuery) zu ermitteln.

[0073] Eine Klasse „ApplicationServer“ (die dem Lastberechnungsagenten **102** zugeordnet ist) bietet die Methode „DetectFreeCapacity()“ an, um die Belastung des dedizierten Servers **100** im Hinblick auf die Anforderungsbedingungen zu berechnen.

[0074] Ein besonderer Vorteil der erfindungsgemäßen Lösung ist darin zu sehen, dass der erfindungsgemäße Dienst zum lastverteilten Zuweisen von Taskflows auf eine Serverfarm **10** das bestehende Taskflow-Managementsystem **300** integriert werden kann.

[0075] Darüber hinaus ist es möglich, eine Lastverteilung nicht nur nach den im Stand der Technik bekannten Algorithmen (RoundRobin/LeastConnection) durchzuführen, sondern es können auch spezi-

fische medizinische und taskflow-spezifische Kriterien konfiguriert werden. Dadurch kann zum Beispiel der Zielservers **100** ermittelt werden, der für den jeweiligen Anwendungsfall beispielsweise mit einer Volumenrenderingkarte oder mit einer Hochleistungsgrafikkarte ausgestattet ist. Es können auch heterogene Serverfarmen **100** unterstützt werden.

[0076] Durch die erfindungsgemäßen Konfigurationsoptionen der Anforderungsbedingungen ist es möglich, nicht nur die Bedingungen bzw. Sub-Bedingungen an sich zu definieren, sondern zusätzlich auch noch festzulegen, wie häufig die jeweilige Bedingung bzw. Sub-Bedingung erfüllt sein muss. Somit kann auch die Quantität einer Anforderungsbedingung bzw. einer Sub-Bedingung erfasst werden. Auch ist es möglich, die jeweiligen Sub-Bedingungen einer Anforderungsbedingung über das Zuweisen von Gewichtungen untereinander geeignet zu priorisieren.

[0077] Für das Installieren der Programmiererweiterung **101** auf den Servern **100** steht vorzugsweise eine Programmiererweiterungsschnittstelle auf den jeweiligen Lastberechnungsagenten **102** zur Verfügung. Die Programmiererweiterungsschnittstelle ist insbesondere so ausgebildet, dass Programmcode eingebunden werden kann, der beispielsweise auf .NET basiert.

[0078] Wie vorstehend bereits erläutert zielt eine weiterbildende Ausführungsform der Erfindung darauf ab, das Ergebnis des Lastverteilungsverfahrens zu quantifizieren. Dabei wird eine Information bereitgestellt, welcher Teil des Taskflows auf welchem Zielservers **100** zur Ausführung gebracht werden soll. Ein Ergebnis kann beispielsweise lauten, dass der Taskflow XY zu 10% auf Servermaschine A kommt, zu 20% auf Servermaschine B und zu 70% auf Servermaschine C auszuführen ist.

[0079] Abschließend sei darauf hingewiesen, dass die Beschreibung der Erfindung und die Ausführungsbeispiele grundsätzlich nicht einschränkend in Hinblick auf eine bestimmte physikalische Realisierung der Erfindung zu verstehen sind. Für einen einschlägigen Fachmann ist es insbesondere offensichtlich, dass die Erfindung teilweise oder vollständig in Soft- und/oder Hardware und/oder auf mehrere physikalische Produkte – dabei insbesondere auch Computerprogrammprodukte – verteilt realisiert werden kann.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Patentliteratur

- US 7127716 [\[0007\]](#)

Patentansprüche

1. Verfahren zum lastverteiltem Zuweisen von computergestützten medizinischen Taskflows auf zumindest einen Anwendungsserver (**100**) einer Serverfarm (**10**), mit folgenden Verfahrensschritten:

- Erfassen eines Taskflows mit konfigurierbaren Anforderungsbedingungen;
- Erfassen von Lastinformationen von verfügbaren Anwendungsservern (**100**) der Serverfarm (**10**), wobei es konfigurierbar ist, welche Lastinformationen erfasst werden sollen, indem auf ausgewählten oder auf allen Anwendungsservern (**100**) zumindest eine Programmiererweiterung (**101**) installiert ist;
- Ermitteln zumindest eines Ziel-Anwendungservers, wobei zumindest ein solcher Anwendungsserver (**100**) aus der Serverfarm (**10**) als Ziel-Anwendungsserver bestimmt wird, der gemäß den erfassten Lastinformationen die erfassten Anforderungsbedingungen erfüllt.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das Verfahren, alle oder ausgewählte Verfahrensschritte des Verfahrens, insbesondere das Ermitteln des Ziel-Anwendungservers, automatisch ausführt.

3. Verfahren nach zumindest einem der vorstehenden Ansprüche dadurch gekennzeichnet, dass das Verfahren zusätzlich folgenden Verfahrensschritt umfasst:

- Automatisches Weiterleiten des Taskflows an den Ziel-Anwendungsserver zum Zwecke der Ausführung.

4. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die Anforderungsbedingung mehrere Sub-Bedingungen umfassen kann, wobei die Sub-Bedingungen mit logischen Operatoren verknüpfbar sind.

5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass das Verfahren computer-implementiert ist und insbesondere software-basiert gesteuert wird.

6. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die Lastinformationen über jeweils einen Lastberechnungsagenten (**102**) erfasst werden, der jeweils auf einem der Anwendungsserver (**100**) der Serverfarm (**10**) installiert ist.

7. Verfahren nach zumindest einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass jeweils ein Lastberechnungsagent (**102**) als Lastinformation eine Last des jeweiligen Servers (**100**) berechnet, auf dem er installiert ist.

8. Verfahren nach zumindest einem der vorste-

henden Ansprüche, dadurch gekennzeichnet, dass das Verfahren zusätzlich folgenden Verfahrensschritt umfasst:

- Ausgeben eines Lastergebnisses, das eine Menge von ermittelten Ziel-Anwendungsservern dahingehend quantifiziert, zu welchem Prozentsatz sie die erfasste Anforderungsbedingung erfüllen.

9. Verfahren nach zumindest einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die Anforderungsbedingung physikalische Systemparameter und/oder logische Merkmale umfasst.

10. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass ein Ziel-Anwendungsserver oder eine Gruppe von Ziel-Anwendungsservern ermittelt wird, wobei die Gruppe von ermittelten Ziel-Anwendungsservern hinsichtlich konfigurierbarer Kriterien priorisiert wird.

11. Verfahren nach zumindest einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass bei dem Ermitteln des zumindest einen Ziel-Anwendungservers die aktuelle Lastverteilung der Serverfarm (**10**) nach konfigurierbaren Lastverteilungskriterien berücksichtigt wird.

12. Verfahren nach zumindest einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass das Erfassen des Taskflows mit konfigurierbaren Anforderungsbedingungen und das Erfassen der Lastinformationen in einer Konfigurationsphase ausgeführt werden, die einer Lastverteilungsphase zeitlich vorangestellt ist, in der das Ermitteln des Ziel-Anwendungservers ausgeführt wird.

13. Verfahren nach zumindest einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass das Erfassen der Lastinformationen unabhängig von dem Erfassen des Taskflows ist.

14. Verfahren nach zumindest einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass zumindest ein Lastverteilungsdienst (**104**) auf zumindest einem Anwendungsserver (**100**) der Serverfarm (**10**) installiert ist, der mit einem Taskflowverwaltungssystem (**300**) kommuniziert und von ihm den Taskflow entgegennimmt und der mit den Lastberechnungsagenten (**102**) der Anwendungsserver (**100**) kommuniziert und sie beauftragt, die Last gemäß den aktuell konfigurierten Kriterien zu berechnen und der deren Ergebnis einsammelt und gegebenenfalls weiterverarbeitet.

15. Computerprogrammprodukt mit Computerprogrammcode zur Durchführung aller Verfahrensschritte des Verfahrens nach einem der Patentansprüche 1 bis 14, wenn der Computerprogrammcode auf dem Computer ausgeführt wird.

16. Computerprogrammprodukt nach Anspruch 15, wobei der Computerprogrammcode auf einem von einem Computer lesbaren Medium gespeichert ist.

17. System zum lastverteiltem Zuweisen von computergestützten medizinischen Taskflows auf zumindest einen Anwendungsserver einer Serverfarm, umfassend:

- zumindest ein Taskflowverwaltungssystem (**300**), das die Taskflows verwaltet;
- eine Vielzahl von Lastberechnungsagenten (**102**), wobei auf jedem Anwendungsserver (**100**) zumindest ein Lastberechnungsagent (**102**) installiert ist, der zum Berechnen einer Last des jeweiligen Anwendungsservers (**100**) bestimmt ist;
- zumindest einen zentralen Lastverteilungsdienst (**104**), der auf zumindest einem Anwendungsserver (**100**) installiert ist und in Datenaustausch mit dem Taskflowverwaltungssystem (**300**) und mit den Lastberechnungsagenten (**102**) steht, wobei der Lastverteilungsdienst (**104**) zumindest ein Anforderungsmodul umfasst, das zum Erfassen von Anforderungsbedingungen für den zuzuweisenden Taskflow bestimmt ist und wobei der Lastverteilungsdienst (**104**) die Anforderungsbedingungen des Anforderungsmoduls zum Erfassen der Last an die jeweiligen Lastberechnungsagenten (**102**) weitergibt.

Es folgen 2 Blatt Zeichnungen

Anhängende Zeichnungen

