Zellbasierte Migration für die avatarbasierte Lastverteilung

Hauptprojekt Ausarbeitung

Sven Allers

HAW Hamburg Master Informatik Wintersemester 2017/18 sven.allers@haw-hamburg.de

Zusammenfassung. Verteilte virtuelle Umgebungen ermöglichen die Interaktion mehrerer Teilnehmer in einer virtuellen Welt. Steigt die Teilnehmerzahl auf ein gewisses Niveau, so ist es unerlässlich, dass eine Verteilung der Last, über mehrere Maschinen, vorgenommen wird. Die avatarbasierte Lastverteilung ist ein Verfahren, dass diese Aufgabe übernehmen soll. Zur dynamischen Migration, während der Laufzeit, wird bisher für jeden Avatar periodisch der passende Spielserver berechnet. Diese Arbeit stellt die zellbasierte Migration vor. Hierbei handelt es sich um ein Verfahren, dass die periodische Neuberechnung verbessern soll, indem vor einer Neuberechnung systematisch nach Kandidaten gesucht wird, bei denen potentiell ein Serverwechsel sinnvoll ist. Zudem wird in dieser Arbeit das Verhalten der zellbasierten Migration untersucht und mit der periodischen Neuberechnung verglichen.

Schlüsselwörter: Lastverteilung, verteilte virtuelle Umgebungen

1 Einführung

In verteilten virtuellen Umgebungen können verschiedene Teilnehmer innerhalb einer dreidimensionalen virtuellen Welt miteinander interagieren. Solche Systeme kommen in den verschiedensten Domänen zum Einsatz. So werden sie im militärischen Bereich als auch für Simulationen und Online-Spiele eingesetzt. Vor allem wenn es sich um sogenannte Massive Multiplayer Online Games (MMOG) handelt, kann die Teilnehmerzahl relativ hoch sein. So wurden bei Guild Wars 2 bereits mehr als 460.000 [15] und Eve-Online über 65.000 Spieler [1] zeitgleich gehostet. Bei solchen Teilnehmerzahlen ist es unerlässlich ein effizientes Lastverteilungsverfahren einzusetzen. Trotz vorhandener Verfahren ist es möglich, dass die Last nicht optimal verteilt ist. Während Spiele wie Guild Wars 2 und World of Warcraft dieses Problem unter anderem mittels Sharding lösen [8,12] und so die maximale Spielerzahl einzelner Shards begrenzt werden kann, ist dies bei Spielen mit einer einzelnen Gesamtumgebung komplexer. So kam es 2013 bei Eve-Online zu erheblichen Problemen bei der Performance [18].

Bei der Lastverteilung innerhalb einer Umgebung kann grundsätzlich zwischen zwei

Arten von Verfahren unterschieden werden. Zum einen partitionierende Verfahren, die die simulierte Umgebung in räumliche Bereiche einteilen, sowie welche die ohne partitionieren der Umgebung auskommen. Die von partitionierenden Verfahren erstellten Partitionen werden genutzt um die Teilnehmer über die Maschinen zu verteilen. So wird jedem Server eine Partition zugewiesen und alle Teilnehmer, die sich innerhalb der Partition eines Servers befinden, werden diesem zugewiesen. Die einfachste Methode hierfür ist die statische Partitionierung, wie sie z.B. in Second Life genutzt wird [16]. Hier wird fest definiert, welcher Bereich von welchem Server verwaltet werden soll. Bei dynamischen Verfahren können die Partitionen nachträglich angepasst werden, damit auf die aktuelle Lastsituation reagiert werden kann. Ein Verfahren auf dem einige dynamische Partitionierungsverfahren basieren sind die sogenannten Mikrozellen. Hierbei wird die simulierte Welt zunächst in viele kleine Zellen unterteilt. Den einzelnen Servern werden dynamisch Zellen zugeordnet, so dass die Menge der zugeordneten Zellen der Partition des Servers entspricht. Sollte sich die Lastsituation verändern, weil sich z.B. innerhalb eines Bereiches mehr Teilnehmer als zuvor befinden, so können einzelne Zellen an andere Server abgetreten werden [6]. Ein Problem bei diesem Verfahren ist, dass die verwalteten Bereiche eines Servers unter Lastsituationen auseinander gerissen werden können und so keine zusammenhängende Regionen mehr verwaltet werden. Dies erhöht jedoch die Interserverkommunikation [5]. Um die Integrität zu garantieren, treten deshalb eine Vielzahl von Verfahren nur Zellen ab, die sich am Rand einer Partition befinden und übergeben diese nur an Server, die an diese Zellen grenzen [7,11]. Andere Verfahren mit dem Ansatz, erlauben es zusätzlich Zellen an nicht benachbarte Server abzutreten, wenn die benachbarten zu stark ausgelastet sind. Um die Integrität der verwalteten Regionen dennoch möglichst hoch zu halten, wird in [5] eine Zusammenführung von Regionen gestartet, wenn die Interserverkommunikation zu hoch ist. Jedoch setzen nicht alle Verfahren auf eine Partitionierung der simulierten Umgebung. Partitonierungsverfahren haben den Nachteil, dass bei einer Repartitionierung auch alle Teilnehmer, die sich in dem umverteilten Gebiet befinden, dem neuen Server zugewiesen werden müssen [10]. Ohne Partitionen können die Avatare unabhängig von ihrer räumlichen Position auf die Server verteilt werden. Hierbei können z.B. Standardverfahren wie Round Robin zum Einsatz kommen [13]. Auch bei der avatarbasierten Lastverteilung handelt es sich um ein partitionsfreies Verfahren. Dadurch ist es möglich, dass Teilnehmer die sich im gleichen Bereich befinden von verschiedenen Servern verwaltet werden. Dennoch wird versucht, Avatare die nah beieinander sind, möglichst auf einem Server zu verwalten um die Interserverkommunikation zu verringern. Dazu wird über die simulierte Umgebung ein Raster mit einer Heat Map aufgezogen, die beschreibt wie viele Avatare ein Server in einer Zelle verwaltet [3].

Ein weiterer wichtiger Aspekt ist die sogenannte Area of Interest (AoI). Dieses beschreibt das Gebiet um einen Avatar, in dem sich potentiell Objekte befinden, mit dem dieser interagieren oder die er gar manipulieren könnte [14]. Diese Objekte können allerdings auch von anderen Spielservern verwaltet werden. Sollte dies der Fall sein, so müssen Veränderungen dieser Objekte zwischen den Servern kommuniziert werden [5].

Im nächsten Abschnitt wird zunächst die avatarbasierte Lastverteilung näher erläutert.

Außerdem wird die zellbasierte Migration, als neues Migrationsverfahren, vorgestellt. Im folgenden Abschnitt wird dieses mittels Experimenten, mit der bereits vorliegenden periodischen Neuberechnung verglichen. Anschließend gibt es ein Fazit in dem das Erreichte evaluiert wird. Zum Abschluss gibt es einen Ausblick auf mögliche zukünftige Arbeiten.

2 Avatarbasierte Lastverteilung

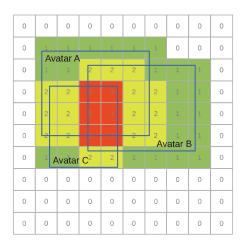


Abb. 1: Heat Map mit den AoIs verschiedener Avatare [3]

Die avatarbasierte Lastverteilung ist ein partitionsfreies Lastverteilungsverfahren. Dennoch ist es das Ziel, Avatare, die räumlich nah beieinander sind, demselben Server zuzuweisen, da so die Interserverkommunikation verringert werden kann. Durch die Partitionslosigkeit, soll effizienter auf kurzfristig auftretende hohe Lastsituationen reagiert werden können. Verwaltet werden die Avatare, die einem Spielserver zugewiesen sind mittels einer Heat Map. Hierbei handelt es sich um ein Raster, das über die simulierte Umgebung aufgespannt wird. Es gibt jeweils eine Heat Map pro Server. Innerhalb der einzelnen Zellen der Heat Map wird abgespeichert, wie viele AoIs von Avataren sich innerhalb einer Zelle befinden, die von dem Server verwaltet werden. Je mehr AoIs sich innerhalb einer Zelle befinden, desto heißer ist die Zelle (Abb. 1). Die Granularität der Heat Map wird mittels der Präzision P beschrieben. Diese bezeichnet die Anzahl der Zellen pro Achse. Eine Präzision von P=10 bedeutet somit in einem dreidimensionalem Raster, dass dieses aus $10 \times 10 \times 10 = 1000$ Zellen besteht. Verwaltet werden die Heat Maps der Spielserver in einer kombinierten Heat Map, die die Heat Maps in einer Datenstruktur gemeinsam ablegt. Mithilfe dieser Heat Map wird ein geeigneter Server gesucht, von dem ein Avatar verwaltet werden soll. Ziel ist es, dass ein Avatar einem Spielserver zugewiesen wird, der möglichst viele AoIs in den Zellen

verwaltet, die sich innerhalb des AoI des Avatars befinden oder alternativ bereits AoIs von Avataren in Zellen verwaltet, die sich möglichst nah an dem Avatar befinden. Um flexibel auf erhöhte Lastbedingungen reagieren zu können, wird zudem die Auslastung der Spielserver betrachtet. Dadurch ist es möglich, dass mehrere Spielserver Avatare innerhalb der selben Zelle verwalten. [3]

Zur Berechnung des Spielservers wird der Gruppierungsfaktor g(a,s) eines Avatars a zum potentiellen Spielserver s ermittelt. Dieser Faktor beschreibt wie stark die räumliche Nähe des neuen Avatars zu den auf dem Server bereits vorhandenen Avataren ist. Dazu wird die Summe der Werte innerhalb der Zellen gebildet, mit denen die AoI des Avatars eine Überschneidung hat. Gibt es keine Überschneidung mit den Zellen eines Spielservers, so ist der Gruppierungsfaktor die negative Manhattan-Distanz des AoI zur nächsten Zelle des Spielservers, in dem sich ein AoI befindet. Unter Einbeziehung der CPU-Auslatung und des freien Speichers eines Spielservers als Metriken für die Auslastung eines Servers ergibt sich folgende Kostenfunktion für die Zuweisung eines Avatars a auf einen Spielserver s:

$$cost(a,s) = w0 * -g(a,s) + w1 * cpu - w2 * memory$$
(1)

Wobei w0, w1 und w2 Gewichtungsfaktoren sind. Zur Ermittlung des geeigneten Spielservers für einen Avatar, wird diese Berechnung für jeden Spielserver durchgeführt und der Avatar dem Spielserver mit den geringsten Kosten zugewiesen. [3]

Zur Ermittlung des Zustandes der Spielserver besteht die Architektur der avatarbasierten Lastverteilung aus zwei Komponenten. Zum einen dem Client Assignment Monitor sowie dem Load Balancer. Der Client Assignment Monitor beobachtet den Spielserver und übermittelt dessen Zustand an den Load Balancer. Darunter die Auslastung des Servers sowie Informationen, die zur Migration einzelner Avatare benötigt werden. Der Load Balancer ist für die Zuweisung der Avatare an die verschiedenen Server zuständig. Dazu wird dieser zunächst kontaktiert, wenn ein neuer Avatar zur simulierten Umgebung hinzustößt. Mithilfe des zuvor beschriebenen Verfahrens ermittelt der Load Balancer einen geeigneten Spielserver. Zudem ist er dafür zuständig festzustellen, wenn die Verteilung der Avatare aus Sicht des Verfahrens nicht mehr gut genug ist, um daraufhin entsprechende Migrationen von Avataren einzuleiten. Aktuell wird dazu für jeden Avatar in regelmäßigen Abständen das zuvor beschriebene Verfahren durchgeführt. Unterscheidet sich der berechnete Server von dem zugewiesenen Server, so wird der Migrationsprozess eingeleitet. [3]

2.1 Zellbasierte Migration

Die zellbasierte Migration ist ein Verfahren um zu ermitteln, ob einzelne Avatare aus Sicht der avatarbasierten Lastverteilung anderen Spielservern zugewiesen werden sollten. Hierbei handelt es sich um den Versuch einer Optimierung der periodischen Neuberechnung für Avatare [3]. Anstatt einer regelmäßigen Neuberechnung für alle Avatare, sucht das zellbasierte Verfahren zuvor nach potentiellen Kandidaten, bei denen ein Wechsel des Spielservers nötig sein könnte. Ziel ist es dadurch die Last des

Load Balancers zu verringern, weil die Neuberechnung für deutlich weniger Avatare stattfindet. Dies soll zudem die Anzahl an Migrationen verringern.

Zur Ermittlung von potentiellen Kandidaten iteriert das Verfahren in periodischen Abständen über alle in der kombinieren Heat Map belegten Felder. Ist ein Feld von mehr als einem Spielserver belegt, so wird eine Neuberechnung für alle Avatare initialisiert dessen AoIs sich in dem Feld befinden. Diese sind potentielle Kandidaten, da sich AoIs, die in der selben Zelle liegen, überschneiden können. Wäre dies der Fall, so könnten die zugehörigen Avatare dieselben Objekte manipulieren. Ein Zusammenführen der Avatare auf den selben Spielserver könnte somit zu einer geringeren Interserverkommunikation führen. Zur Neuberechnung wird nun für jeden Avatar das Verfahren der avatarbasierten Lastverteilung eingesetzt. Wird ein anderer Spielserver ermittelt als der Avatar zugewiesen ist, so wird der Migrationsprozess wie in [3] beschrieben eingeleitet.

Damit bei der zellbasierten Migration bekannt ist, für welche Avatare eine Neuberechnung stattfinden muss, wurde das Protokoll zwischen Monitor und Load Balancer um entsprechende Informationen erweitert. Dazu wird übermittelt welche Zellen durch die AoIs der zugewiesenen Avatare genutzt werden. Bei jeder Übertragung wird hierbei die Differenz zum letzten Stand übermittelt, der übertragen wurde.

3 Experimente

Um die Qualität der zellbasierten Migration zu ermitteln, wurde mittels Experimenten überprüft, wie sich das Verfahren bei bewegenden Avataren verhält. Hierbei wurde auf die Versuchsumgebung aus [2] zurückgegriffen. Im Folgenden wird das zellbasierte Verfahren mit der zuvor eingesetzten periodischen Neuberechnung verglichen. Getestet wurden für die einzelnen Verfahren jeweils zwei Szenarien. In einem gab es eine fixe Anzahl an Avataren. Dieses wurde genutzt um zu analysieren, ob die Verfahren in der Lage sind, die Gruppierung der Avatare, mittels Migration, beizubehalten. Zudem wurde hier überprüft, wie oft Migrationen stattfanden. Im zweiten Szenario wurden nach dem Initialisieren der Umgebung weitere Avatare hinzugefügt. Ziel war es hier, herauszufinden, welche Auswirkungen, die zusätzlichen Avatare, auf die Performance der jeweiligen Verfahren haben.

3.1 Aufbau der Experminte

Durchgeführt wurden die Experimente mit drei virtuellen Maschinen. Hierbei handelte es sich jeweils um Ubuntu 16.04 Maschinen, die mit einem Intel Xeon Prozessor mit 2,4 GHz und 4 GB RAM-Speicher betrieben wurden. Zur Simulation einer MMOG-Umgebung wurden verschiedene Server gestartet. Dies waren ein QuP-Mock, zum simulieren eines QuP-Servers, ein Load Balancer und drei Spielserver. Zusätzlich wurde ein Masterknoten gestartet, der die Tests kontrolliert. Bei QuP [4] handelt es sich um das Persistenzsystem, welches von der MMOG-Umgebung genutzt wird, um den Zustand der Spielwelt zu verwalten. Zudem können mithilfe des QuP-Clients, der im Spielserver integriert ist, die AoIs der Avatare verwaltet werden. Die einzelnen Server

wurden über die verschiedenen Maschinen verteilt. Auf der ersten Maschine lief der Masterknoten zusammen mit dem QuP-Mock, auf der zweiten Maschine liefen zwei Spielserver und auf der dritten Maschine lief ein Spielserver zusammen mit dem Load Balancer.

In beiden Szenarien hatte die Heat Map eine Präzision von P=10. Da der Test in einer dreidimensionalen Umgebung stattgefunden hat, bedeutet dies, dass die Heat Map aus 1000 Zellen bestand. Die Avatare bewegen sich hierbei in einen normalisierten Raum. Dadurch ist die Position der Avatare auf jeder Achse ein Wert zwischen 0 und 1. Die Tests gingen jeweils über eine Länge von 30 Minuten, wobei sich jeder Avatar in einer Taktrate von 350 ms um eine Strecke von 0,0001 bewegt hat. Jeder Avatar hat somit zu Ende des Tests eine Strecke von etwa 0,5 zurückgelegt. Als Bewegungsmuster kam die in [2] beschriebene Mischung aus Random Way Point (RWP) [9] und Hot Points All (HPA) [17] zum Einsatz.

Im ersten Szenario wurden 2500 Avatare gleichmäßig über die Umgebung verteilt. Im zweiten Szenario wurden insgesamt 4000 Avatare gleichmäßig über die Umgebung verteilt. Wobei hier nur 2500 Avatare bereits zu Testbeginn initialisiert wurden. Die restlichen 1500 Avatare wurden während des Tests in einem Abstand von jeweils einer Sekunde initialisiert. Beide Szenarien wurden jeweils für die periodische Neuberechnung sowie für die zellbasierte Migration durchgeführt. Bei der periodischen Neuberechnung, fand die Neuberechnung für jeden Avatar in einem Abstand von jeweils 15 Sekunden statt. Die zellbasierte Migration durchsuchte die Heat Map auch in einem Rhythmus von 15 Sekunden nach potentiellen Kandidaten zur Neuberechnung.

3.2 Ergebnisse und erste Analyse

Zunächst soll das erste Szenarion betrachtet werden. Bei der Ausgangsposition, zu Minute 0, lag bei beiden Verfahren die selben Bedingungen vor, da bisher nur die initiale Zuweisung stattgefunden hat. In Abb. 2 ist zu sehen, welchen Spielservern die Avatare im Laufe des Experiments, bei der zellbasierten Migration, zugewiesen waren. Es ist zu erkennen, dass die äußeren Bereiche, tendenziell eher von Spielserver 2 verwaltet werden. Die von Spielserver 3 verwalteten Avatare, bewegen sich mit der Zeit in Richtung Zentrum. Ähnlich sieht es mit den verwalteten Avataren von Spielserver 1 aus. Auffallend ist, dass die Streuung der Avatare bei den einzelnen Spielservern mit der Zeit abnimmt und die verwalteten Avatare immer dichter beieinander sind. Insgesamt bleibt die räumliche Gruppierung der Avatare erhalten. Wird das Verhalten mit der periodischen Neuberechnung verglichen (Abb. 3), so ist sichtbar, dass auch dieses Verfahren in der Lage ist die Gruppierung beizubehalten. Ein qualitativer Unterschied lässt sich in diesem ersten Test nicht ausmachen. Der hier durchgeführte Test sowie der Test aus [2] deuten jedoch darauf hin, dass das periodische Verfahren eher dazu neigt einen einzelnen Spielserver mit weniger Avataren auszustatten. So waren auf Spielserver 1 zu Ende des Tests noch 2 Avatare. Auch bei der zellbasierten Migration scheinen einzelne Spielserver mit der Zeit deutlich weniger Avatare zu haben. Gegen Ende des Tests waren hier auf Spielserver 3 noch 99 Avatare vorhanden. Die periodische Neuberechnung scheint jedoch stärker zu dem Phänomen zu neigen. Allerdings müsste

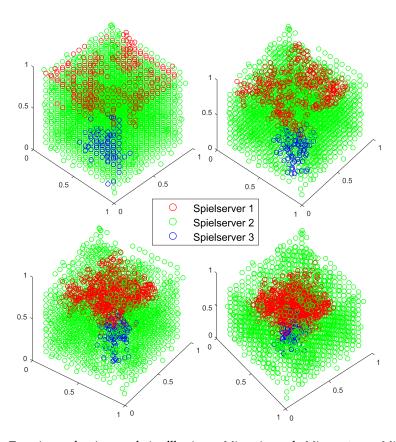


Abb. 2: Zuweisung der Avatare bei zellbasierter Migration: o.l.: Minute 0, o.r.: Minute 9, u.l.: Minute 18, u.r.: Minute 27

dies noch genauer analysiert werden, um die Hypothese zu bestätigen.

In Tabelle 1 ist zu sehen wie stark die Avatare, im ersten Szenario, bei der zellbasierten Migration gewechselt sind. Im Durchschnitt haben alle Avatare während des Tests 0,3192 mal den Server gewechselt. Werden nur die Avatare betrachtet, die während des Tests mindestens einmal den Server gewechselt haben, so ist das unter denen im Schnitt 1,0528 mal. Zudem hat der Avatar mit mit den meisten Wechseln 3 mal den Server gewechselt. Verglichen mit der periodischen Neuberechnung (Tabelle 2) ist dies deutlich geringer. Dort wechseln die Avatare im Durchschnitt 0,8092 mal den Server und somit ca. 2,5 mal häufiger als bei der zellbasierten Migration. Werden nur die Avatare betrachtet, die den Server während des Tests gewechselt haben, so haben diese 1,7746 mal den Server gewechselt und somit ca. 1,7 mal häufiger als bei der zellbasierten Migration. Noch stärker fällt dies aus, wenn das Maximum betrachtet wird. So hat der Avatar mit den meisten Wechseln 8 mal den Spielserver gewechselt. Dies ist etwa das 2,67-Fache von der zellbasierten Migration. Auch die Anzahl der zugewiesenen Server pro Avatar ist bei der zellbasierten Migration geringer. Hier wurde jeder Avatar

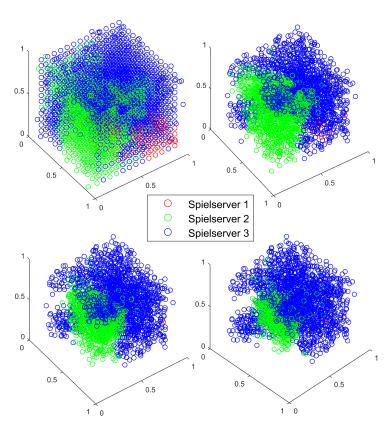


Abb. 3: Zuweisung der Avatare bei periodischer Neuberechnung: o.l.: Minute 0, o.r.: Minute 9, u.l.: Minute 18, u.r.: Minute 27

im Schnitt 1,3064 Servern zugewiesen. Werden nur die Avatare betrachtet, die den Server gewechselt haben, so waren diese im Durchschnitt 2,0106 verschiedenen Servern zugewiesen. Bei der periodischen Neuberechnung war dies mit 1,4924 knapp 1,14 mal mehr als bei der zellbasierten Neuberechnung. Werden nur die Avatare betrachtet, die Server gewechselt haben, so wurden diese im Durchschnitt 2,0798 Servern zugewiesen, was ca. 1,03 mal mehr als bei der zellbasierten Migration ist. Die minimal sowie die maximal zugewiesenen Server sind mit einem bzw. drei Spielservern bei beiden Verfahren identisch. Auch wenn tiefer gehende Tests noch benötigt werden, um die Zahlen zu bestätigen, so ist ein Trend erkennbar. Die zellbasierte Migration scheint zu weniger Migrationen zu neigen. Das Ziel, die Anzahl der Migrationen zu verringern, scheint sich somit in diesem Test zu bestätigen.

Im zweiten Szenario wurde Untersucht, wie sich eine höhere Anzahl an Avatare auf die Belastung des Load Balancers auswirkt. In Abb. 4a ist zu sehen, wie sich, bei Einsatz der zellbasierten Migration, eine höhere Anzahl an Avataren auf die initiale Be-

Tabelle 1: Wechsel der Avatare zwischen den Servern bei der zellbasierten Migration

	Durchschnitt	Durchschnitt	Min	Max
	alle	Avatare die gewechselt haben		
Wechsel	0,3192	1,0528	0	3
Anzahl Zugewiesene Server	1,3064	2,0106	1	3

Tabelle 2: Wechsel der Avatare zwischen den Servern bei der periodischen Neuberechnung

	Durchschnitt alle	Durchschnitt Avatare die gewechselt haben	Min	Max	
Wechsel	0,8092	1,7746	0	8	
Anzahl Zugewiesene Server	1,4924	2,0798	1	3	

rechnungszeit zur Zuweisung eines Avatars auf die Server, auswirkt. Es ist zu erkennen, dass bei steigender Anzahl der Avatare die Berechnungszeit konstant bei um die 10 ms bleibt. Wird jedoch die Berechnungszeit während des laufenden Tests, mit der vor Beginn des Tests verglichen, so ist zu sehen, dass mit dem Beginn des Tests und somit dem Einsetzen der zellbasierten Migration die Berechnungszeit steigt. Erkennbar ist dies in Abb. 5a. Hier ist die Berechnungszeit pro Avatar dargestellt, wobei diese wiederum in die Avatare vor Testbeginn und zu Testlaufzeit aufgeteilt sind. Bis Avatar 2500 handelt es sich um die initiale Zuweisung zu Beginn des Tests. Alle folgenden Berechnungen wurden während des laufenden Tests durchgeführt. Hier ist ein deutlicher Anstieg zu erkennen. So ist die durchschnittliche Berechnungszeit während der Initialisierung bei ca. 0,9 ms, beim laufenden Test liegt sie jedoch bei etwa 10,85 ms und somit etwa 12 mal so hoch. Wird dies mit der periodischen Neuberechnung verglichen, so ist zu sehen, dass auch hier bei einer steigenden Anzahl der Avatare die Berechnungszeit konstant bleibt (Abb. 4b). Jedoch ist die Berechnungszeit geringer und liegt bei etwa

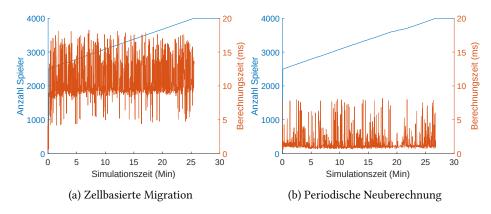


Abb. 4: Berechnungszeit bei steigender Anzahl an Avataren (95er Perzentile)

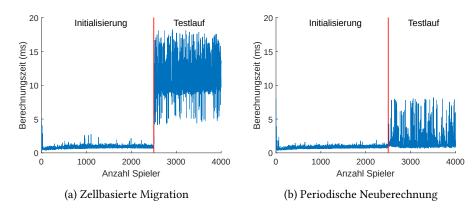


Abb. 5: Berechnungszeiten pro Avatar (95er Perzentile)

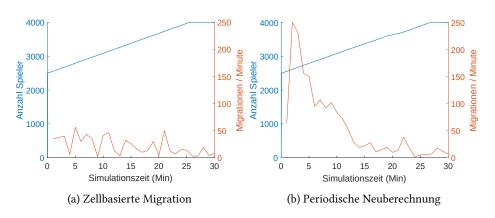


Abb. 6: Migrationen pro Minute bei steigender Anzahl an Avataren

0,5 ms. Wird die Berechnungszeit während der Initialisierung mit der zur Testlaufzeit verglichen, so ist zu erkennen, dass diese im Gegensatz zur zellbasierten Migration nicht so gravierend ansteigt (Abb. 5b). Hier liegt ähnlich, wie bei der zellbasierten Migration während der initialen Zuweisung, die Berechnungszeit bei Durchschnittlich ca. 0,83 ms. Die Berechnungszeit zur Laufzeit des Tests liegt jedoch bei ca. 1,54 ms. Sie entspricht damit zwar dem etwa 1,86-fachen der initialen Zuweisung, die Berechnungszeit für die zellbasierte Migration zur Laufzeit dauert jedoch im Schnitt ca. 7 mal länger, als die der periodischen Neuberechnung. Auffällig ist bei der periodischen Neuberechnung, dass Ausreißer bei denen die Berechnung länger dauert zur Laufzeit stärker sind als während der Initialisierung. Somit ist bei beiden Verfahren eine Zusatzbelastung durch die Verfahren sichtbar. Bei der zellbasierten Migration fällt diese jedoch deutlich kräftiger aus. Trotz der geringeren Anzahl an Neuberechnungen, konnte somit eine Entlastung des Load Balancers nicht erreicht werden. Die Belastung ist stattdessen sogar gestiegen. Dies könnte an der aufwändigeren Suche der potentiellen Kandidaten liegen. Hier sind jedoch genauere Analysen nötig.

Abb. 6a stellt die Anzahl der Migrationen mit einer höheren Anzahl an Avataren, für die zellbasierte Migration, dar. Es ist zu erkennen, dass die Anzahl an Migrationen, mit einer höheren Anzahl an Avataren leicht fällt. Die Anzahl der Migrationen bewegt sich von ca. 40 Migrationen pro Minute in dem Zeitraum zwischen 5 und 10 Minuten auf ca. 20 zwischen den Minuten 20 und 25. Der Abfall an Migrationen ist mit mehr werdenden Avataren jedoch nicht signifikant und es sind auch starke Schwankungen vorhanden, so dass es bei Minute 21 genau 50 Migrationen gab. Mit um die 10 Migrationen pro Minute ist die Anzahl ab Minute 25 deutlich am geringsten und auch die Schwankungen sind hier geringer. Dies kann damit zusammenhängen, dass ab diesem Zeitpunkt keine neuen Avatare den Servern mehr zugewiesen werden. Die periodische Neuberechnung zeigt ein ähnliches Verhalten auf, wie sie es bei vorherigen Tests [2] bereits getan hat (Abb. 6b). Sie neigt dazu zu Beginn des Test sehr viele Migrationen durchzuführen, die im Laufe des Tests stark abfallen. So gab es während Minute 2 ganze 250 Migrationen. Zu Minute 13 waren es noch 29 Migrationen und blieb ab dem Zeitpunkt durchgehend bei unter 50 Migrationen pro Minute. Somit ist festzustellen, dass sich die Anzahl der Migrationen, nachdem sie anfänglich deutlich höher war, mit der Zeit auf das Niveau der zellbasierten Migration einpendelt.

4 Fazit

In dieser Arbeit wurde die zellbasierte Migration vorgestellt. Ziel war es den bisherigen Berechnungsprozess zur Migration zu verbessern und so eine Entlastung des Load Balancers zu erreichen. Mittels verschiedener Experimente wurde das Verfahren überprüft und mit der bisherigen periodischen Neuberechnung verglichen. Hierbei hat sich herausgestellt, dass bei der zellbasierten Migration weniger Migrationen bei den Avataren auftreten. Damit konnte, nach diesen ersten Tests, scheinbar eines der Ziele des neuen Verfahrens erreicht werden. Jedoch hat sich auch gezeigt, dass sich das Niveau der Migrationen bei der periodischen Neuberechnung mit der Zeit auf das Niveau der zellbasierten Migration einpendelt. Eine längerfristige Analyse ist deswegen zwingend nötig. Es konnte zudem gezeigt werden, dass die zellbasierte Migration den Load Balancer stärker belastet als die periodische Neuberechnung. Das Ziel, die Belastung zu verringern, konnte somit nicht erreicht werden. Eine erste Hypothese ist, dass die Ursache in dem Suchprozess nach potentiellen Kandidaten zum Serverwechsel liegt. Dies bedarf allerdings einer gründlicheren Analyse. Trotz der stärkeren Belastung konnte während des laufenden Tests eine konstante Berechnungszeit bei einer steigenden Anzahl an Avataren erreicht werden.

5 Ausblick

In dieser Arbeit wurde mit der zellbasierten Migration ein erster Ansatz vorgestellt, der die periodische Neuberechnung verbessern sollte. Es hat sich allerdings gezeigt, dass die Belastung des Load Balancers durch die zellbasierte Migration nicht verringert werden konnte und sogar gestiegen ist. In zukünftigen Arbeiten soll dies deshalb weiter untersucht werden. Eine mögliche Lösung für dieses Problem könnte ein weiteres Optimieren des Ermitteln der potentiellen Kandidaten sein, indem schon bei Erhalt der

Zustände der Spielserver nachvollzogen wird, in welchen Zellen mehrere Spielserver die AoIs von Avataren verwalten. Somit könnte auf eine Suche über alle Zellen verzichtet werden. Des weiteren wurden noch nicht alle Aspekte in dem vorgestellten Verfahren berücksichtigt. So ist es weiterhin möglich, dass ein Avatar in jeder Neuberechnungsphase einem neuen Spielserver zugewiesen wird und somit dauerhaft zwischen zwei oder mehr Spielservern wechselt. In einer weiteren Optimierung wäre es das Ziel, die Wahrscheinlichkeit hierfür zu verringern. Zudem hat das aktuelle Verfahren den Nachteil, dass Zellen in denen die Avatare nur von einem Spielserver verwaltet werden, nicht betrachtet werden. Wenn sich die Avatare innerhalb der Zelle jedoch relativ weit weg von den restlichen Avataren auf dem Spielserver sind, könnte es Sinn machen die Avatare dennoch einem neuen Spielserver zuzuweisen. In zukünftigen Arbeiten könnte ein Verfahren entwickelt werden, um solch isolierte Zellen zu ermitteln und somit zu stark verteilte Avatare auf einem Spielserver zu vermeiden. Die nächste Arbeit soll sich zunächst mit einer tieferen Analyse des Verfahrens auseinandersetzen und einen Vergleich mit anderen bereits vorliegenden Verfahren tätigen.

Literatur

- 1. EVE-Offline :: EVE-Online Status monitor. http://eve-offline.net/?server=tranquility, Abruf: 2017-08-07
- Allers, S.: Testumgebung für die Avatarbasierte Lastverteilung in verteilten virtuellen Systemen. (2017), S. 1–18
- 3. Behnke, L.; Allers, S.; Wang, Q.; Grecos, C.; Luck, K. von: Avatar Density Based Client Assignment. In: *Entertainment Computing ICEC 2016* Bd. 1, 2016. ISBN 9783319461007, S. 137–148
- 4. ВЕНИКЕ, L. ; GRECOS, C. ; LUCK, K. von: QuP : Graceful Degradation in State Propagation for DVEs. (2014)
- CHEN, J.; WU, B.; DELAP, M.; KNUTSSON, B.; LU, H.; AMZA, C.: Locality aware dynamic load management for massively multiplayer games. In: Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming - PPoPP '05 (2005), 289. http: //dx.doi.org/10.1145/1065944.1065982. - DOI 10.1145/1065944.1065982. ISBN 1595930809
- DE VLEESCHAUWER, B.; VAN DEN BOSSCHE, B.; VERDICKT, T.; DE TURCK, F.; DHOEDT, B.;
 DEMEESTER, P.: Dynamic microcell assignment for massively multiplayer online gaming.
 In: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games NetGames '05 (2005), 1-7. http://dx.doi.org/10.1145/1103599.1103611. DOI 10.1145/1103599.1103611. ISBN 1595931562
- Deng, Y.; Lau, R. W. H.: Dynamic load balancing in distributed virtual environments using heat diffusion. In: ACM Transactions on Multimedia Computing, Communications, and Applications 10 (2014), feb, Nr. 2, 1–19. http://dx.doi.org/10.1145/2499906. – DOI 10.1145/2499906. – ISSN 15516857
- 8. Johanson, C.; Loretan, S.: Introducing the Megaserver System | Guild-Wars2.com. https://www.guildwars2.com/en/news/introducing-the-megaserver-system/. Version: 2014, Abruf: 2017-08-07
- JOHNSON, D. B.; MALTZ, D. A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: *Mobile Computing* 353 (1996), 153–181. http://dx.doi.org/10.1007/b102605. – DOI 10.1007/b102605. ISBN 978-0-7923-9697-0

- KAZEM, I.; AHMED, D. T.; SHIRMOHAMMADI, S.: A visibility-driven approach to managing interest in distributed simulations with dynamic load balancing. In: Proceedings IEEE International Symposium on Distributed Simulation and Real-Time Applications, DS-RT (2007), S. 31–38. http://dx.doi.org/10.1109/DS-RT.2007.11. DOI 10.1109/DS-RT.2007.11. ISBN 0769530117
- LAU, R. W.: Hybrid load balancing for online games. In: Proceedings of the international conference on Multimedia MM '10 (2010), 1231. http://dx.doi.org/10.1145/1873951.1874194. DOI 10.1145/1873951.1874194. ISBN 9781605589336
- LEE, Y. T.; CHEN, K. T.: Is server consolidation beneficial to MMORPG? A case study of world of warcraft. In: Proceedings 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010 (2010), S. 435-442. http://dx.doi.org/10.1109/CLOUD.2010.57. DOI 10.1109/CLOUD.2010.57. ISBN 9780769541303
- Lu, F.; Parkin, S.; Morgan, G.: Load balancing for massively multiplayer online games.
 In: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games NetGames '06 (2006), 1. http://dx.doi.org/10.1145/1230040.1230064. DOI 10.1145/1230040.1230064. ISBN 1595935894
- Lui, J. C. S.; Chan, M. F.: An efficient partitioning algorithm for distributed virtual environment systems. In: *IEEE Transactions on Parallel and Distributed Systems* 13 (2002), Nr. 3, 193–211. http://dx.doi.org/10.1109/71.993202. DOI 10.1109/71.993202. ISSN 10459219
- 15. O'BRIEN, M.: Guild Wars 2: The First Year | GuildWars2.com. https://www.guildwars2.com/en/news/guild-wars-2-the-first-year/. Version: 2013, Abruf: 2017-08-07
- 16. ROSEDALE, P.; ONDREJKA, C.: Enabling Player-Created Online Worlds with Grid Computing and Streaming. http://www.gamasutra.com/resource_guide/20030916/ rosedale_pfv.htm. Version: 2003, Abruf: 2015-08-10
- RUEDA, S.; MORILLO, P.; ORDUÑA, J. M.; VALENCIA, U. D.: A Peer-To-Peer Platform for Simulating Distributed Virtual Environments. (2007). ISBN 9781424418909
- 18. X, C. P.: Building a Balanced Universe EVE Community. http://community.eveonline.com/news/dev-blogs/building-a-balanced-universe/. Version: 2013, Abruf: 2017-01-15