

Vorlesung *Verteilte und Web-Informationssysteme* im Wintersemester 2011/2012

Hausaufgabe Blatt 3
Gruppe 14
Ralf Vogler
Hoffmann Max-Emanuel
Sebastian Wiesner
08.11.2011

Aufgabe 1)

Der `Receive-Operator` erzeugt einen Hintergrund-Thread `Receiver.ReceiverThread`, welcher die gesamte Netzwerkkommunikation durchführt. Die Kommunikation zwischen Thread und Operator geschieht über einen Warteschlange mit 300 freien Plätzen. Der Thread fügt empfangene Objekte (sowohl die Attribut-Namen der Relation als auch die Tupel selbst) in diese Warteschlange ein, der Operator entfernt sie und gibt sie über die `DBIterator`-Schnittstelle zurück.

Der Thread hält die Verbindung permanent offen und fordert mit einem Request immer 100 Tupel aus der entfernten Relation an. Dies geschieht immer, sobald 100 Plätze im Puffer frei sind. Der Puffer ist also immer mindestens zu zwei Dritteln gefüllt, so dass im Regelfall immer Tupel für den `Receive-Operator` zur Verfügung stehen.

Dazu sendet er einem speziellen `MultipleNext-Request`, dem die Anzahl der erwarteten Tupel folgt. Der `Send-Operator` auf Server-Seite antwortet auf diesem Request mit einer entsprechenden Anzahl Tupel aus der darunter liegenden Relation. Enthält die Relation keine weiteren Tupel, so verschickt der `Send-Operator` `null`-Werte.

Beim Schließen der Relation unterbricht der `Receive-Operator` den Thread, welcher daraufhin die entfernte Relation und anschließend die Netzwerkverbindung schließt.

Die Proxy-Klassen verbinden lediglich die Send- und Receive-Operatoren und führen ihrerseits keine Netzwerk-Operationen durch.

Quellcode: Siehe "VWIS Gruppe 14_ Blatt 3 (Vogler, Hoffmann, Wiesner)-project.zip"

Aufgabe 2)

Welche Information wird in den inneren Knoten abgelegt, um den Mobilfunkbenutzer zu lokalisieren?

Ein Knoten wird durch eine Tabelle repräsentiert, welche einer Handynummer entweder einen weiteren Knoten in einer tieferen Ebene zuordnet (Pointer) oder die direkten Kundeninformationen (-> Blatt). Falls der Knoten also für diesen Nutzer verantwortlich ist, befindet sich dieser im Senderaum dieser Zelle und enthält die spezifischen Kundeninformationen. Wenn die Zelle nicht für den Nutzer verantwortlich ist, dann hat sie einen Zeiger auf eine tiefer gelegene Zelle/Knoten der für diese Nummer zuständig ist.

Wo wird die Suche initiiert? Wie sieht die Suche nach der aktuellen Zone eines Benutzers in dem Baum aus?

Die Suche beginnt an dem hierarchisch niedrigsten Knoten, welcher sich für die zu suchende Handynummer verantwortlich fühlt.

Wenn dieser Knoten einen Eintrag über den Kunden hat, dann ist zu unterscheiden ob es nur ein Verweis ist oder aber die spezifischen Kundeninformationen direkt vorliegen. Im letzteren Fall können die Kundeninformationen direkt weiter geleitet werden. Beinhaltet der Knoten jedoch keine Information über die Nummer, dann wird der nächst höhere Knoten in der Hierarchie befragt und dieser wird wieder geprüft. Diese Prozedur wiederholt sich so lange bis der Knoten gefunden wird der die Kundeninformationen enthält (fertig) oder ein Knoten der einen Verweis auf diesen hat. Im letzteren Fall müssen natürlich die Pointer wieder den Baum abwärts verfolgt werden um die Basisstation zu ermitteln.

Wie wird der Baum bei einem Zonenwechsel eines Mobilfunkbenutzers aktualisiert?

Verändert sich die verantwortliche Zelle eines Benutzers, dann müssen die Daten entlang des Suchpfades aktualisiert werden. Somit wird wie bei der gewöhnlichen Suche der Pfad zum ursprünglich verantwortlichen Knoten beschriftet und auf jedem Knoten der passiert wird, dieser Eintrag gelöscht falls er einen Eintrag über genau diesen Knoten besitzt. So lange man sich in der Hierarchie nach oben bewegt wird der Eintrag aktualisiert bzw. neu eingefügt falls er nicht vorhanden war. Am Ende werden die spezifischen Kundeninformationen mit in den nun zuständigen Knoten kopiert.

Aufgabe 3)

Content-Addressable Networks (CANs)

CAN kann man als verteilte Infrastruktur beschreiben, welche die Probleme bekannter verteilter Systeme wie Napster und Gnutella adressiert. Dabei werden wie bei Hashtabellen CAN Schlüssel auf Werte abgebildet.

Ein CAN besteht aus mehreren Knoten, die jeweils ein Teil der gesamten Hashtabelle enthalten und einen Koordinatenraum bilden. Informationen über Nachbarknoten sind in jedem Knoten abgelegt. Bei einer Operation auf einem Schlüssel wird die Anfrage von Knoten zu Knoten weitergeleitet, bis der Endknoten erreicht wird, in dessen Zone das Schlüssel-Wert-Paar abgespeichert ist.

Damit das verteilte System CAN wachsen kann, müssen neue Knoten der Struktur hinzugefügt werden. Dazu wird der Koordinatenraum eines existierenden Knoten in zwei gleiche Hälften aufgeteilt. Die eine Hälfte behält der alte Knoten, die andere Hälfte wird dem neuen Knoten zugewiesen. Das Hinzufügen eines Knotens zum CAN findet in 3 Schritten statt:

- Zuerst muss der neue CAN-Knoten die IP-Adresse eines beliebig teilnehmenden Knotens herausfinden. Dies geschieht durch eine Anfrage an den zum CAN gehörenden DNS Domain-Namen, der die IP-Adresse dieses Knotens zurückliefert.
- Der neue Knoten wählt nun zufällig einen Punkt P des Koordinatenraumes aus. Danach sendet der Knoten eine Join-Anfrage für Punkt P an das CAN. Die Nachricht wird von Knoten zu Knoten weitergeleitet, bis der Knoten mit dem Punkt P in seiner Zone erreicht wird. Die Zone wird geteilt, der neue Knoten erhält die eine Hälfte der Zone mit Schlüssel-Wert-Paaren, der alte Knoten als sein Trennungspartner die andere Hälfte.
- Damit nach dem Einfügen ein korrektes Routing sichergestellt werden kann, müssen die Nachbarn des aufgeteilten Raums für den neuen und den alten Knoten benachrichtigt und entsprechend auf den neuesten Stand gebracht werden. Um das System auf den neuesten Stand zu halten, sendet jeder Knoten periodisch eine Update-Nachricht mit dem Koordinatenbereich seiner eigenen Zone an alle Nachbarn.

Wird ein Knoten entfernt, muss die zugeordnete Zone von den übrigbleibenden Knoten übernommen werden. Falls die Zone mit genau einer Nachbarzone verschmolzen werden kann, ist die Aufgabe damit erledigt. Ansonsten wird die Verwaltung der Zone temporär an den Nachbarn mit der kleinsten eigenen Zone übergeben.

Unterschiede zwischen CAN und DNS:

Beim DNS ist die Struktur hierarchisch aufgebaut, im Gegensatz zum CAN bei dem jeder Knoten gleichwertig ist.

Ein CAN Netzwerk kann beliebige Informationen auf beliebige Schlüssel mappen, was beim DNS-System nicht der Fall ist, denn dort sind die Formate für Domain –Namen und IP-Adressen fest vorgegeben.

CAN regelt die Größe der Datenhaltungspunkte automatisch, was bei DNS nicht der Fall ist.