

# StreamGlobe: Adaptive Anfragebearbeitung und Optimierung auf Datenströmen

Bernhard Stegmaier                      Richard Kuntschke  
TU München, Fakultät für Informatik, Lehrstuhl Informatik III  
<Vorname.Nachname>@in.tum.de

**Abstract:** Die aktuelle Forschung im Bereich der Datenstromverarbeitungssysteme untermauert die zunehmende Bedeutung von Datenströmen, etwa im Kontext von Sensornetzwerken und in Netzwerken zur Informationsgewinnung. Mit dem Aufkommen verschiedenartiger mobiler Geräte, die in allgegenwärtige (drahtlose) Netzwerke eingebunden werden können, ist die Entwicklung von Datenstrom-Management-Systemen (DMS) zur Informationsgewinnung in derartigen Netzwerken zu einer großen Herausforderung geworden. In dieser Arbeit beschreiben wir die Architektur unseres *StreamGlobe* Projekts, das sich effiziente Anfragebearbeitung auf Datenströmen in verteilten, heterogenen Umgebungen zum Ziel setzt. In diesem System ermöglichen selbstorganisierendes Routing von Datenströmen und ausdrucksstarke Anfragebearbeitung im Netzwerk gezielte Informationsbeschaffung. Wir schließen mit einem Bericht über den aktuellen Status des Projekts.

## 1 Einleitung

In den letzten Jahren haben Peer-to-Peer (P2P) Netzwerke sowohl in den Medien als auch in der Forschung große Aufmerksamkeit erlangt. Ein Beispiel dafür sind Super-Peer-Netzwerke [YGM03], wie sie auch in [LSWN03] zum Einsatz kommen. Aufbauend auf den Techniken des Vorgängerprojekts ObjectGlobe [BKK<sup>+</sup>01], das die verteilte Bearbeitung persistenter Daten im Netzwerk realisiert, verfolgen wir mit *StreamGlobe* das Ziel eines verteilten, selbstorganisierenden Datenstrom-Management-Systems (vgl. [CCD<sup>+</sup>03, YG02, DKR04, KS04, FHK<sup>+</sup>03, ACD<sup>+</sup>03]) für effizienten Datenfluss und Anfragebearbeitung in (P2P) Netzwerken.

Als Beispielszenario für eine mögliche Anwendung von StreamGlobe in einem Sensornetzwerk betrachten wir Abbildung 1. Das dargestellte Netzwerk ist als Super-Peer-Netzwerk organisiert, wobei  $SP_0$  bis  $SP_3$  den sogenannten Super-Peer-Backbone bilden und fünf möglicherweise mobile Peers  $P_0$  bis  $P_4$  mit dem Backbone verbunden sind. Die Peers  $P_0$ ,  $P_2$  und  $P_3$  – im Beispiel ein Mobiltelefon, ein Notebook und ein PDA – registrieren Anfragen im Netzwerk. Im Gegensatz dazu sind die Peers  $P_1$  und  $P_4$  Sensoren, die ihre Sensordaten in Form von XML-Datenströmen an das Netzwerk liefern. Angenommen,  $P_4$  ist ein Datenstrom, der von speziellen Sensoranzügen von Einsatzkräften, z. B. der Feuerwehr, geliefert wird. Dieser liefert kontinuierlich Sensormesswerte, welche die Identität der Einsatzkraft (id), ihre Position in Form von GPS-Koordinaten (x, y) und Informationen über ihre Vitalfunktionen und die Umweltbedingungen enthalten. Wir haben als Messwerte exemplarisch Körpertemperatur (kt), Pulsrate (pr) und Sauerstoffsättigung (ss), sowie Umgebungstemperatur (ut), Kohlendioxidkonzentration (CO2) und Schwefeldioxidkonzentration (SO2) in der Luft gewählt. Aus Gründen der kürzeren Darstellung verwenden wir die folgende vereinfachte DTD, um den Datenstrom zu beschreiben, obwohl StreamGlobe tatsächlich XML Schema verwendet.

```
<!ELEMENT Messung (id, x, y, kt, pr, ss, ut, CO2, SO2)>  
<!ELEMENT id (#PCDATA)> ...
```

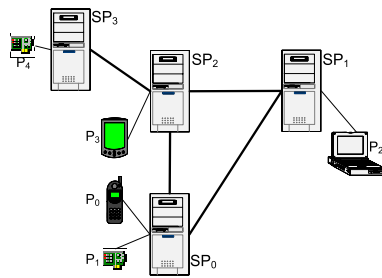


Abbildung 1: Beispielszenario

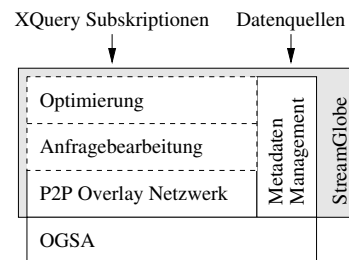


Abbildung 2: Architekturübersicht

Seien nun  $P_0$  bzw.  $P_2$  Geräte, die Notärzte bzw. die Feuerwache verwenden. Erstere sollen eine Benachrichtigung erhalten, wenn die Sauerstoffsättigung einer Einsatzkraft einen kritischen Wert erreicht. Dazu registriert  $P_0$  die folgende Anfrage in XQuery.

```
for $m in stream("Einsatzkräfte")/Messung
where $m/ss < 92 or $m/ss > 98
return <alert> {$m/id} {$m/x} {$m/y} {$m/ss} </alert>
```

Die Feuerwache interessiert sich für die Überwachung der Umweltbedingungen, um z. B. eine Warnung auszugeben, falls die Bedingungen für Einsatzkräfte vor Ort oder Anwohner kritisch werden. Dazu registriert  $P_1$  die folgende Anfrage in XQuery.

```
for $m in stream("Einsatzkräfte")/Messung
return <gas> {$m/id} {$m/x} {$m/y} {$m/CO2} {$m/SO2} </gas>
```

StreamGlobe behandelt dieses Szenario wie folgt. Die Daten von  $P_4$  werden an  $SP_3$  geliefert und dort möglichst früh gefiltert, so dass nur die Elemente id, x, y, ss, CO2 und SO2 im Strom verbleiben, welcher somit die Informationen zur Beantwortung der Anfragen von  $P_0$  und  $P_2$  enthält. Der Strom wird zu  $SP_2$  geleitet, wo er repliziert wird und beide entstehenden Ströme entsprechend gefiltert werden, so dass sie nur noch die von  $P_0$  bzw.  $P_2$  benötigten Daten enthalten. Diese Ströme werden schließlich über  $SP_1$  zu  $P_0$  bzw.  $P_2$  geleitet. Durch dieses Routing der Datenströme wird der Netzwerkverkehr im Vergleich zu herkömmlichen Systemen erheblich verringert, da redundante Übertragungen vermieden werden. Der Ansatz der Verlagerung der Anfragebearbeitung von den Peers ins Netzwerk zur Reduzierung der Netzwerklast ist ein Hauptmerkmal von StreamGlobe und unterscheidet unser System von bisherigen Arbeiten, z. B. im Multicast-Bereich.

Die übrige Arbeit gliedert sich wie folgt: In Kapitel 2 wird die Systemarchitektur von StreamGlobe beschrieben und kurz auf die Optimierung und Anfragebearbeitung eingegangen. Kapitel 3 schließt unsere Ausführungen mit einer Zusammenfassung und einem kurzen Bericht über den Status der Implementierung unseres StreamGlobe Prototyps.

## 2 StreamGlobe Architekturübersicht

Abbildung 2 zeigt die Architektur von StreamGlobe auf den Peers. Gestrichelt gekennzeichnete Komponenten sind in Abhängigkeit der Fähigkeiten der Peers in unterschiedlichem Umfang vorhanden, z. B. besitzen Thin-Peers i. A. nur rudimentäre Möglichkeiten zur Anfragebearbeitung und keine Optimierungskomponente.

### 2.1 Grundlegende Infrastruktur

StreamGlobe setzt auf der *Open Grid Service Architecture (OGSA)* Plattform und deren Referenzimplementierung, dem *Globus Toolkit (GT3)*, auf. Die in Abbildung 2 gezeigten Schichten werden als kooperierende Dienste in dieser Architektur realisiert. OGSA erlaubt die beliebige Kommunikation zwischen Diensten, was in Hinblick auf mobile Peers nicht immer sinnvoll ist – z. B. sollen mobile Sensoren nur über zugeordnete Knoten mit den

restlichen Teilnehmern kommunizieren können. Daher wird eine zusätzliche P2P-Schicht etabliert: Das Netzwerk besteht aus einer Menge von *Peers*. Jeder Peer besitzt Nachbar-Peers, mit welchen er kommunizieren kann. Zum Transfer von Daten zwischen zwei beliebigen Peers muss ein *Transferpfad* von Peers im Netzwerk aufgebaut werden. Konkret könnte das Overlay-Netzwerk beispielsweise durch ein Super-Peer-Netzwerk [YGM03] realisiert werden. Zur Integration verschiedenster Peers – von kleinen, mobilen Geräten mit wenig Rechenleistung bis hin zu leistungsfähigen, stationären Servern – zu einem Informationsnetzwerk werden Peers nach ihrer Leistungsfähigkeit klassifiziert. *Thin-Peers* stellen Geräte mit relativ wenig Rechenleistung dar – insbesondere sind sie nicht in der Lage, aufwendige Anfragebearbeitung durchzuführen. Dagegen sind *Super-Peers* stationäre Rechner mit hoher Rechenleistung. Diese Super-Peers bilden einen Backbone und übernehmen im Netzwerk Anfragebearbeitungsaufgaben, welche andere (Thin-)Peers nicht ausführen können. Als Metadaten-Verwaltung (MDV) kommt eine Weiterentwicklung der MDV von ObjectGlobe [KKKK02] zum Einsatz, welche auf den *Service Data* und *Service Discovery* Mechanismen von GT3 basiert. Die MDV verwaltet u. a. die Nachbarschaftsbeziehungen und Fähigkeiten der Peers, vorhandene Subskriptionsregeln, Datenströme, sowie Statistiken über Datenströme, d. h. Größe und Frequenz der enthaltenen Elemente.

## 2.2 Benutzerschnittstelle

Benutzer registrieren *Subskriptionsregeln* zur Informationsgewinnung in XQuery an einem bestimmten Peer, welcher im Normalfall das Arbeitsgerät des Benutzers darstellt. In unserem Kontext sind Subskriptionen echte Anfragen – im Gegensatz zu Ansätzen aus der Literatur, bei welchen lediglich die zu einer Anfrage passenden Dokumente zugestellt werden – und ermöglichen ausdrucksstarke Transformationen der Datenströme, um sie flexibel auf individuelle Bedürfnisse zuschneiden zu können.

Analog registrieren Datenquellen ihre Datenströme an einem bestimmten Peer. Eine Datenquelle kann ihre Daten mit zugehörigem XML Schema als individuellen Datenstrom registrieren, welcher damit im StreamGlobe-Netzwerk unter einer eindeutigen Kennung zur Verfügung steht. Eine andere Möglichkeit ist die Registrierung der Daten als Teil eines *virtuellen Datenstroms*, welcher die Daten aller beteiligten Datenquellen bündelt und unter einer eindeutigen Kennung zur Verfügung stellt. Diese Möglichkeit wird im einleitenden Beispiel benutzt, um die Sensordaten aller Einsatzkräfte zur Auswertung zu vereinigen (mergen). Die Einspeisung in das Netzwerk wird von speziellen Operatoren, den *Wrappern*, übernommen, welche auf den verantwortlichen Peers ausgeführt werden.

## 2.3 Optimierung

In StreamGlobe wird eine verteilte, hierarchisch organisierte Optimierung eingesetzt. Das Netzwerk wird dazu in Segmente partitioniert. In jedem Segment übernimmt ein *Speaker-Peer* die Optimierung und die Koordination mit Nachbarsegmenten. Die Aufgabe der Optimierung in StreamGlobe ist die Bestimmung von Peers, auf welchen (Teil-)Anfragen bzw. Subskriptionen ausgeführt werden, und der dazu nötigen Transferpfade für Datenströme. Dabei werden folgende Ziele verfolgt:

1. Registrierung beliebiger Subskriptionen an den Peers, ungeachtet der Fähigkeiten des jeweiligen Peers.
2. Erzielung eines möglichst guten Datenflusses in Bezug auf das Transfervolumen im Netzwerk, ohne das Netzwerk mit redundanten Übertragungen zu belasten.
3. Gemeinsame Optimierung der Ausführung vieler Subskriptionsanfragen.

Die Ziele (1) und (3) werden durch Verlagerung der Anfragebearbeitung in das Netzwerk erreicht. In jedem Segment analysiert der Speaker-Peer alle Subskriptionsregeln. Gemeinsame Teilanfragen oder Teilanfragen, welche die Fähigkeiten eines bestimmten Peers übersteigen, werden extrahiert und in diesem Segment nur einmalig an Peers mit entsprechenden Kapazitäten ausgewertet. Die ursprünglichen Subskriptionen werden so

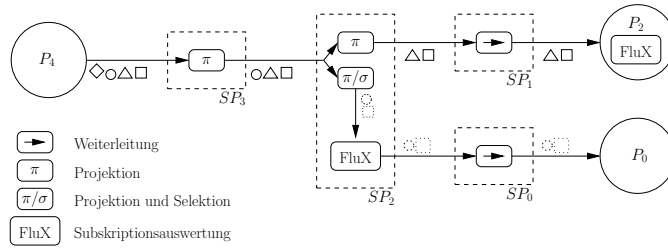


Abbildung 3: Auswertungsplan des Beispielszenarios

modifiziert, dass sie anstatt der individuellen Auswertung dieser Teilanfragen die von den extrahierten Teilanfragen neu generierten, spezialisierten Datenströme benutzen. Diese Optimierung kann, z. B. bei Aggregation von Datenströmen, weiter zur Verringerung des Netzwerkverkehrs beitragen. Die Aggregation wird dazu einmalig nahe der Datenquelle ausgeführt und nur das (viel kleinere) Ergebnis an die entsprechenden Peers übermittelt. Das zweite Ziel wird durch *Filterung* und *Clustering von Datenströmen* erreicht und wurde bereits am einleitenden Beispiel verdeutlicht. Zur strukturellen Filterung werden aus einer Subskription die minimal notwendigen Teile der Schemata der beteiligten Datenströme bestimmt. Die beteiligten Datenströme werden auf diese minimalen Schemata projiziert. Weiterhin findet eine inhaltliche Filterung statt, indem Prädikate aus der Subskription extrahiert werden. Damit werden Datenobjekte, welche diese Prädikate nicht erfüllen und somit nichts zur Subskriptionsregel beitragen, frühzeitig aus den Datenströmen entfernt. Diese zweiteilige Filterung wird von *Filter-Anfragen* auf den Transferpfaden der Datenströme so nahe wie möglich an den jeweiligen Datenquellen ausgeführt. Dadurch lässt sich das Transfervolumen der Datenströme erheblich verringern. Das nachfolgend erläuterte Clustern von Datenströmen verbessert die Situation weiter. In bisherigen Systemen werden Datenströme individuell (und damit auch redundant) von Datenquellen zu den Konsumenten übertragen. StreamGlobe analysiert den Datenfluss im Netzwerk und verhindert die mehrfache Übertragung, indem für mehrere Ausprägungen des gleichen Datenstroms nur ein Datenstrom (Cluster) mit dem gemeinsamen Schema aller Ausprägungen übermittelt, kurz vor den Konsumenten entsprechend aufgespaltet und wiederum gefiltert wird. Abbildung 3 zeigt beispielhaft die globale Auswertungsstrategie anhand des Beispielszenarios. Die Symbole an den Datenströmen symbolisieren dabei Gruppen von Elementen, z. B. steht das Rechteck für die Elemente  $id$ ,  $x$  und  $y$ . Bei strukturellen Filtern fallen entsprechende Elemente weg, das Ergebnis von inhaltlichen Filtern (Selektionen) und anderen Subskriptionsauswertungen ist gestrichelt gekennzeichnet. In weiteren Ausbaustufen soll die Optimierungskomponente, aufbauend auf Arbeiten in ObjectGlobe [BKK03], um Quality-of-Service Management, z. B. zur Einbeziehung von Maximal-Bandbreiten für Verbindungen, und Lastbalanzierungskonzepte erweitert werden.

Die Optimierung ist ein kontinuierlicher Prozess, welcher alle Änderungen, z. B. An- und Abmeldungen von Subskriptionen und Datenquellen, berücksichtigt. Damit wird ein sich selbst optimierendes und ständig rekonfigurierendes Netzwerk erzielt, welches sich jederzeit ohne Fremdeingriff in einem möglichst guten Zustand befindet.

## 2.4 Anfragebearbeitung

StreamGlobe zielt auf die Bearbeitung von Datenströmen ab und beinhaltet deshalb *push-basierte* Auswertungsstrategien – im Gegensatz zu traditionellen DBMS, wo Daten von untergeordneten Operatoren angefordert werden. Zur Auswertung der (Teil-)Subskriptionen setzen wir neue Optimierungstechniken (FluX [KSS04]) zur Auswertung von XQuery-Anfragen auf Datenströmen ein. Sie minimieren den Speicherbedarf bei der Anfragebearbeitung und erlauben so die skalierbare Auswertung von Subskriptionsregeln. Bestimmte

Subskriptionen können per se nicht skalierbar auf Datenströmen ausgeführt werden, z. B. wenn Joins enthalten sind. In solchen Fällen wird unendliches Zwischenspeichern verhindert, indem die Benutzer zur Spezifikation von Fensterbedingungen gezwungen werden. Damit wird die Ausführbarkeit auf unendlichen Datenströmen sichergestellt. Die (z. B. zeitliche) Konsistenz von individuellen Datenfenstern obliegt den ausführenden Operatoren und ist Gegenstand aktueller Forschung.

### 3 Zusammenfassung und Projektstatus

In dieser Arbeit wurde die Architektur von StreamGlobe vorgestellt. StreamGlobe zielt auf die effiziente Informationsgewinnung in heterogenen Netzwerken ab. Im Gegensatz zu bestehenden P2P-Systemen lokalisiert StreamGlobe nicht nur Daten und führt dann die Anfragebearbeitung durch, sondern erzielt durch intelligente Ausnutzung von Anfragebearbeitungskapazitäten im Netzwerk einen möglichst optimalen Datenfluss. Dies wird mittels struktureller und inhaltlicher Filterung, sowie durch Clustering erreicht.

Zu diesem Zeitpunkt ist die grundlegende Infrastruktur für den Aufbau des P2P-Netzwerks implementiert. Des Weiteren existiert eine eigenständige Implementierung für FluX, welche zur Zeit in StreamGlobe integriert wird. Die Optimierungstechniken, wie sie in Abschnitt 2.3 vorgestellt wurden, befinden sich aktuell in der Entwicklung.

### Literatur

- [ACD<sup>+</sup>03] Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Ponceva, M., und Schmidt, R.: P-Grid: a self-organizing structured P2P system. *ACM SIGMOD Record*. 32(3):29–33. September 2003.
- [BKK<sup>+</sup>01] Braumandl, R., Keidl, M., Kemper, A., Kossmann, D., Kreutz, A., Seltzsa, S., und Stocker, K.: ObjectGlobe: Ubiquitous query processing on the Internet. *The VLDB Journal*. 10(1):48–71. August 2001.
- [BKK03] Braumandl, R., Kemper, A., und Kossmann, D.: Quality of Service in an Information Economy. *ACM Transactions on Internet Technology*. 3(4):291–333. November 2003.
- [CCD<sup>+</sup>03] Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, M. J., Hellerstein, J. M., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F., und Shah, M. A.: TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In: *Proc. of the Conf. on Innovative Data Systems Research*. Asilomar, CA, USA. January 2003.
- [DKR04] Domagalski, R. und König-Ries, B.: Möglichkeiten der Anfragebearbeitung in mobilen Ad-hoc-Netzwerken. In: *Grundlagen und Anwendungen mobiler Informationstechnologie, Workshop des GI-Arbeitskreises Mobile Datenbanken und Informationssysteme*. Heidelberg, Germany. March 2004.
- [FHK<sup>+</sup>03] Florescu, D., Hillery, C., Kossmann, D., Lucas, P., Riccardi, F., Westmann, T., Carey, M. J., Sundararajan, A., und Agrawal, G.: The BEA/XQRL Streaming XQuery Processor. In: *Proc. of the Intl. Conf. on Very Large Data Bases*. S. 997–1008. Berlin, Germany. September 2003.
- [KKKK02] Keidl, M., Kreutz, A., Kemper, A., und Kossmann, D.: A Publish & Subscribe Architecture for Distributed Metadata Management. In: *Proc. of the IEEE Intl. Conf. on Data Engineering*. S. 309–320. February 2002.
- [KS04] Krämer, J. und Seeger, B.: PIPES - A Public Infrastructure for Processing and Exploring Streams. In: *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*. Paris, France. June 2004.
- [KSS04] Koch, C., Scherzinger, S., Schweikardt, N., und Stegmaier, B.: Schema-based Scheduling of Event Processors and Buffer Minimization on Structured Data Streams. In: *Proc. of the Intl. Conf. on Very Large Data Bases*. Toronto, Canada. August 2004. Accepted for publication.
- [LSWN03] Löser, A., Siberski, W., Wolpers, M., und Nejd, W.: Information Integration in Schema-Based Peer-To-Peer Networks. In: *Proc. of the Intl. Conf. on Advanced Information Systems Engineering*. S. 258–272. Klagenfurt/Velden, Austria. June 2003.
- [YG02] Yao, Y. und Gehrke, J.: The Cougar Approach to In-Network Query Processing in Sensor Networks. *ACM SIGMOD Record*. 31(3):9–18. September 2002.
- [YGM03] Yang, B. und Garcia-Molina, H.: Designing a Super-Peer Network. In: *Proc. of the IEEE Intl. Conf. on Data Engineering*. S. 49–60. Bangalore, India. March 2003.