StreamGlobe: Processing and Sharing Data Streams in Grid-Based P2P Infrastructures

Richard Kuntschke Bernhard Stegmaier Alfons Kemper Angelika Reiser

Technische Universität München, Munich, Germany
⟨*firstname.lastname*⟩@in.tum.de

1 Introduction and Motivation

Data stream processing is currently gaining importance due to the developments in novel application areas like e-science, e-health, and e-business (considering RFID, for example). Focusing on e-science, it can be observed that scientific experiments and observations in many fields, e. g., in physics and astronomy, create huge volumes of data which have to be interchanged and processed. With experimental and observational data coming in particular from sensors, online simulations, etc., the data has an inherently streaming nature. Furthermore, continuing advances will result in even higher data volumes, rendering storing all of the delivered data prior to processing increasingly impractical. Hence, in such e-science scenarios, processing and sharing of data streams will play a decisive role. It will enable new possibilities for researchers, since they will be able to subscribe to interesting data streams of other scientists without having to set up their own devices or experiments. This results in much better utilization of expensive equipment such as telescopes, satellites, etc. Further, processing and sharing data streams on-the-fly in the network helps to reduce network traffic and to avoid network congestion. Thus, even huge streams of data can be handled efficiently by removing unnecessary parts early on, e. g., by early filtering and aggregation, and by sharing previously generated data streams and processing results.

To enable these optimizations, we use *Peer-to-Peer (P2P) networking* techniques. P2P has gained a lot of attention in the context of exchanging persistent data—in particular for *file sharing*. In contrast to that, we apply P2P networks for the dissemination of individually subscribed and transformed data streams, allowing for *data stream sharing*. By using the computational capabilities of peers in the P2P network, we can push data stream transforming operators into the network, thus enabling efficient in-network query processing. This yields a reduction of network traffic, load balancing among peers, and improved flexibility in terms of the kinds of peers that can register queries in the network. At the same time, multi-subscription optimization allows for data stream sharing and reuse of computational results among different peers. This provides for both, reduced network traffic and decreased computational load of peers. Ultimately, more subscriptions can be processed. Prototype systems like STREAM [2], TelegraphCQ [4], CACG [11], NiagaraCQ [5], ONYX [6], or sensor networks like Cougar [15] have made advances in processing data streams and locally optimizing the evaluation of multiple queries, but do not yet focus on the optimization of data stream processing in a generic distributed data stream management system (DSMS) by distributing stream processing in the network. We propose *StreamGlobe* [12] as a prototype to meet these challenges. The StreamGlobe implementation adheres to established *Grid Computing* [7] standards (OGSA) in order to fit into existing e-science platforms. To ensure interoperability, StreamGlobe is built on top of standards like XML and XQuery for representing data streams and specifying subscriptions.[1]

As a motivating example, we introduce an astrophysical e-science application. Consider Figures 1 and 2 illustrating a simplified example P2P network. SP_0 to SP_3 are *super-peers*, which are more powerful servers constituting a super-peer backbone network [14]. In contrast, P_0 to P_4 are less powerful *thin-peers* (simply called peers in the following), delivering data streams into the network or registering subscriptions. Here, P_4 is a satellite-bound telescope detecting photons and registering a data stream called `photons` at super-peer SP_3. In our demonstration setup, this data stream consists of real astrophysical data collected during the ROSAT All-Sky Survey (RASS) [13], which we obtained through our cooperation partners at the Max Planck Institute for Extraterrestrial Physics [10]. It contains, among other things, the detection time of a photon, its energy, and its origin in celestial coordinates. Let peers P_0 and P_2 be devices of two astrophysicists used to register subscriptions. The first is interested in a certain area of the sky denoted as the *vela supernova rem-*

Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

[1] The terms *query*, *continuous query*, and *subscription* are treated as synonyms throughout this paper.

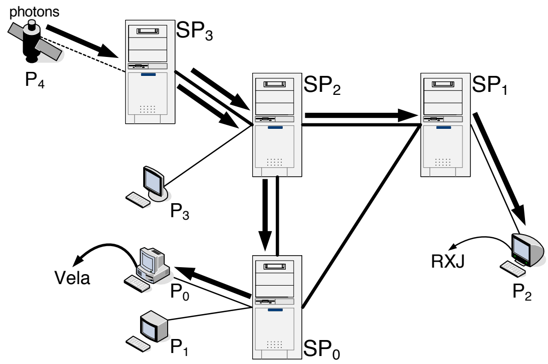

Figure 1: No Optimization

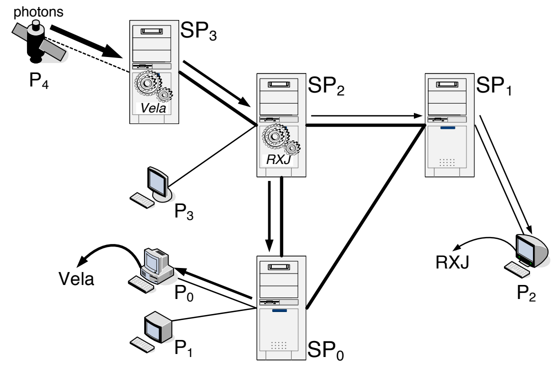

Figure 2: Network-Aware Optimization

nant. Hence, a subscription *Vela* phrased in XQuery that filters out all interesting data of this area from the `photons` data stream is registered. The latter is interested in an area denoted as the *RXJ0852.0-4622 supernova remnant* [3], which is situated within the area of vela but is far more distant and hidden behind vela. Nevertheless, this structure can be visualized by selecting only photons with an energy greater than 1.3 keV. Acquiring this data is accomplished by registering the query *RXJ* at peer P_2. Further, both queries shall select two different (overlapping) sets of measurements from the original `photons` data stream. Unfortunately, we are not able to show the queries and the schema of the data stream due to space limitations.

Systems not performing any in-network query processing in combination with multi-subscription optimization might handle this scenario as depicted in Figure 1. The subscription is instantiated and processed at the peer where it has been registered and the complete data stream is delivered to this peer—even if only small parts of the stream are actually needed. If multiple subscriptions are based on a single data stream, the complete stream is individually transferred to each subscribing peer. Obviously, this may cause network congestion as outlined before. Even if subscriptions would be evaluated directly at the data stream source, there would still be multiple redundant streams in the network, one for each subscription.

StreamGlobe handles this situation differently by combining routing and distributed query processing techniques, as depicted in Figure 2. To achieve this in an efficient way, an optimization component incrementally analyzes newly registered subscriptions with respect to which parts of data streams are needed, which existing data streams may be reused, and how data streams are routed in the network. In our case, it computes the part of the data stream common to both queries *Vela* and *RXJ*. Therewith, a first filter query is computed and instantiated at super-peer SP_3. This filter reduces the size of the original data stream by removing all data not needed by any of the two subscriptions. The precise selection contained in *RXJ* cannot be employed yet, since it would discard data needed for the subscription *Vela*. The now smaller stream is then routed to SP_2, where it is duplicated and routed to P_0 via SP_0 and to P_2 via SP_1. At this point, another filter query at SP_2 is instantiated, which carries out the selection contained in the subscription *RXJ* to further reduce the size of the stream transferred to P_2. Obviously, StreamGlobe is more parsimonious with its resources, since redundant transmissions are avoided and only the relevant parts of data streams are transmitted (depicted by thinner arrows).

Beyond what has been shown in this simple motivating example, StreamGlobe makes the following contributions. We have developed techniques for incrementally optimizing the data flow in the network whenever new data streams or subscriptions are registered. These optimization techniques are able to exploit projections, selections, and window aggregates to compute filter and aggregation queries employing these operations and sharing existing data streams. If no reusable streams are found in the neighborhood, i. e., all data streams are too restricted with respect to their content, we are able to "widen" data streams flowing in the network. That is, existing filters like projections and selections as well as window aggregates are extended such that the new subscription can be satisfied without establishing a new data stream. The optimization is based on a cost model accounting for network traffic and the computational load of peers. The continuously collected and updated statistics needed as input for the cost model are part of each peer's metadata.

2 Application Scenarios in Astrophysics

Even though StreamGlobe is applicable to a variety of application domains, we concentrate on e-science applications in astrophysics. We envision two additional scenarios where StreamGlobe promises benefits for astrophysicists. These are event-triggered observations of rare events (e. g., supernovae) and the classification of luminaries.

The first is a so called "alerter service". Imagine different observatories and robotic telescopes connected to a P2P network. Observational data is streamed into the network and monitored by "alerter subscriptions" registered by facilities operating the robotic telescopes. The subscriptions classify and filter the observational data and trigger (currently available) robotic telescopes to join the observation if potentially interesting events occur.

Automatically classifying spectral energy distributions (SEDs) of observed objects is a pivotal step in various astrophysical key research areas [1]. Roughly, it deals with automatic object classification based on querying multiple

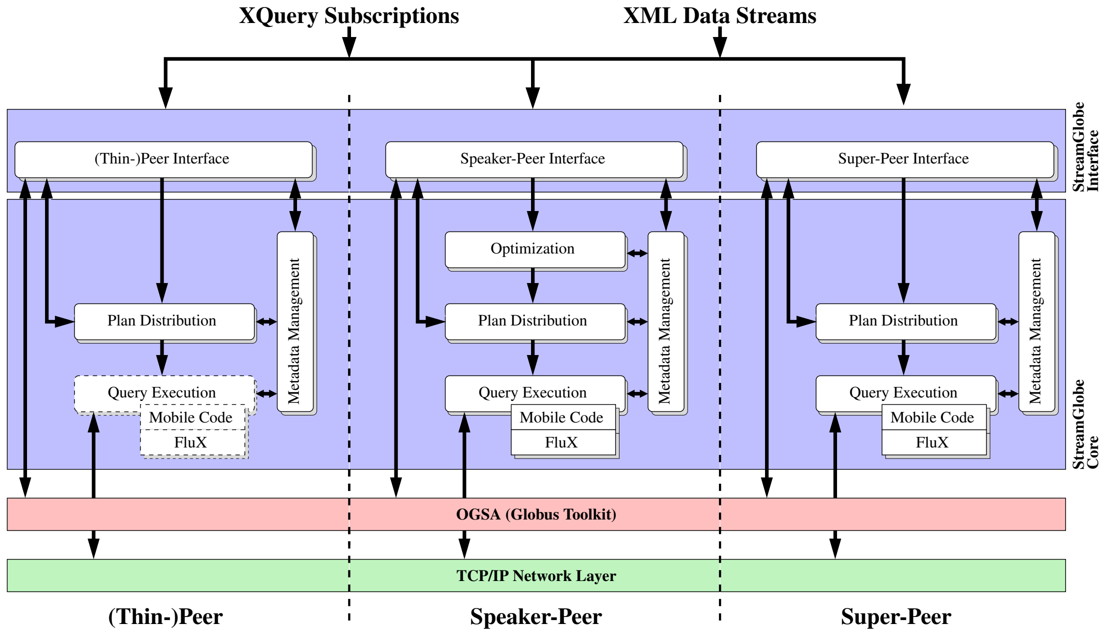

Figure 3: Architecture Overview

distributed catalogs covering different wavelength ranges and computing spatial matches using statistical methods. The catalogs are integrated into StreamGlobe by means of specific wrappers that transform static data into data streams. We push any filters (e. g., filtering of spectra) towards the sites hosting the (wrapped) catalogs to reduce network traffic. The data streams are published in the P2P network and joined with incoming observational data at powerful super-peers calculating the best matching SEDs.

3 StreamGlobe Architecture

In this section, we will give a brief overview of the StreamGlobe system architecture, which is depicted in Figure 3.

Up to now, we have classified peers as super-peers and thin-peers. To be able to efficiently manage and optimize large networks, we further divide the whole network into subnets. In each subnet, a *speaker-peer*, which is elected among all the super-peers of the subnet, is responsible for managing and optimizing its subnet. Additionally, speaker-peers of different subnets establish a backbone for efficient metadata exchange.

Basically, all components of StreamGlobe which are represented by rounded rectangles in Figure 3 are implemented as collaborating Grid services on top of the Globus Toolkit [8]. According to the classification of peers, the provided services are shown individually for each type. The Grid services are divided into two groups as follows. *Interface services* represent the different peers in the network. These services constitute the interfaces between peers and users as well as between peers and other parts of the StreamGlobe system. Users register subscriptions and data streams at these interfaces. Non-XML data streams are fed into the system by means of wrappers which are executed at the corresponding peer to convert the given data format into XML. Furthermore, these interface services receive control messages from other peers and forward them to the appropriate *core service*. The set of core services of a peer comprises all the functionality of that peer. Every peer runs exactly one instance of an interface service and a set of core services according to its capabilities.

Thin-peers are peers with varying functionality. They publish data streams in the network and/or receive the results of their subscriptions, but usually do not carry out complex query processing. Since even such a peer has to provide metadata, e. g., statistics of a data stream needed for optimization, it mainly runs the metadata management service. Beyond that, only the plan distribution service and a minimum query execution service are provided on every thin-peer. The plan distribution service is needed for all peers, since this component is responsible for correctly setting up data communication with other peers and instantiating the queries in the query execution service. The query execution service of a thin-peer is at least able to display results of subscriptions, to publish—and possibly wrap—data streams, and to maintain statistics, if needed. Some thin-peers, depending on their individual capabilities, might be able to perform additional query processing.

The next type of peers are super-peers. In addition to the basic components of thin-peers, super-peers provide extensive query processing capabilities by enabling fully-fledged query processing for data streams. For carrying out query processing tasks, the query execution service employs our extensible XQuery engine *FluX* [9] for data streams, which is installed on every super-peer. FluX is an event-based query engine that achieves buffer minimization through optimizations based on schema information of data streams. It is therefore applicable for efficient stream processing. Furthermore, user-defined stream operators which are implemented as mobile code can be loaded and executed to provide additional functionality. This yields a great amount of flexibility in specifying subscriptions.

Finally, speaker-peers are basically super-peers with the additional role of optimizing and managing their subnets. Hence, speaker-peers provide an optimization service for carrying out these additional tasks.

In Figure 3, the communication paths are depicted by arrows. Different StreamGlobe components/services on a single peer communicate directly via mechanisms provided by the Globus Toolkit. Inter-peer communication takes place between the interface services of the two communicating peers using the RPC mechanisms of Globus. As the OGSA framework does not yet provide any suitable means for data stream transfer, we realized our own protocols based on TCP/IP networking techniques for direct data exchange between query execution services.

4 Demonstration Outline

We will demonstrate our StreamGlobe system for data stream management in P2P networks. The demonstration will be based on astrophysical application scenarios similar to those outlined in Sections 1 and 2 which we have developed in cooperation with our partners at the Max Planck Institute for Extraterrestrial Physics. We employ real astrophysical data and operations provided by our partners. In detail, the demonstration will consist of the following parts.

- We show how to publish data and how to retrieve information by interactively registering peers, data streams, and subscriptions.
- We present the optimization of subscriptions and the routing of data streams by sharing existing data streams already flowing through the network.
- We demonstrate the prevention of overload situations at both network connections and peers.
- We present throughput experiments indicating the increased number of subscriptions that can be satisfied using our optimizations compared to the non-optimized case.
- We show the effectiveness of StreamGlobe in various network topologies and scenarios.

We visualize the underlying network by means of a graphical user interface. This GUI shows the relevant information about the network, e. g., network topology, registered data streams and subscriptions, data flow, utilization of network connections, computational load of peers, etc.

5 Conclusions

In this paper, we have briefly sketched our key ideas for StreamGlobe. The StreamGlobe system has been designed to meet the challenges that arise in processing data streams in an information retrieval network based on P2P networking and Grid technologies. In the demonstration, we will present a running prototype system and confirm the efficiency and scalability of our approach on behalf of some exemplary e-science application scenarios.

In future work, we will continue to extend the functionality of StreamGlobe. Periodic or event-based reoptimization using static multi-query optimization on the set of registered subscriptions can help to retain an optimized data flow in the network over long periods of time. Furthermore, support for subscriptions with multiple input data streams and joins is planned.

References

[1] H.-M. Adorf, F. Kerber, G. Lemson, R. Mignani, A. Micol, T. Rauch, and W. Voges. Assembly and Classification of Spectral Energy Distributions - A new VO Web Service. In *Conf. on Astronomical Data Analysis Software and Systems*, Pasadena, CA, USA, Nov. 2004.

[2] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, R. Motwani, I. Nishizawa, U. Srivastava, D. Thomas, R. Varma, and J. Widom. STREAM: The Stanford Stream Data Manager. *IEEE Data Engineering Bulletin*, 26(1):19–26, Mar. 2003.

[3] B. Aschenbach. Discovery of a young nearby supernova remnant. *Nature*, 396(6707):141–142, Nov. 1998.

[4] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. A. Shah. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *Proc. of the Conf. on Innovative Data Systems Research*, Asilomar, CA, USA, Jan. 2003.

[5] J. Chen, D. J. DeWitt, F. Tian, and Y. Wang. NiagaraCQ: A Scalable Continuous Query System for Internet Databases. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, pages 379–390, Dallas, TX, USA, May 2000.

[6] Y. Diao, S. Rizvi, and M. J. Franklin. Towards an Internet-Scale XML Dissemination Service. In *Proc. of the Intl. Conf. on Very Large Data Bases*, pages 612–623, Toronto, Canada, Aug. 2004.

[7] I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 2nd edition, 2004.

[8] The Globus Alliance. http://www.globus.org, 2005.

[9] C. Koch, S. Scherzinger, N. Schweikardt, and B. Stegmaier. Schema-based Scheduling of Event Processors and Buffer Minimization on Structured Data Streams. In *Proc. of the Intl. Conf. on Very Large Data Bases*, pages 228–239, Toronto, Canada, Aug. 2004.

[10] R. Kuntschke, B. Stegmaier, F. Häuslschmid, A. Reiser, A. Kemper, H.-M. Adorf, H. Enke, G. Lemson, and W. Voges. Datenstrom-Management für e-Science mit StreamGlobe. *Datenbank-Spektrum*, 4(11):14–22, Nov. 2004.

[11] S. Madden, M. A. Shah, J. M. Hellerstein, and V. Raman. Continuously Adaptive Continuous Queries over Streams. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, pages 49–60, Madison, WI, USA, June 2002.

[12] B. Stegmaier, R. Kuntschke, and A. Kemper. StreamGlobe: Adaptive Query Processing and Optimization in Streaming P2P Environments. In *Proc. of the Intl. Workshop on Data Management for Sensor Networks*, pages 88–97, Toronto, Canada, Aug. 2004.

[13] W. Voges et al. The ROSAT All-Sky Survey Bright Source Catalogue. *Astronomy and Astrophysics*, 349(2):389–405, July 1999.

[14] B. Yang and H. Garcia-Molina. Designing a Super-Peer Network. In *Proc. of the IEEE Intl. Conf. on Data Engineering*, pages 49–60, Bangalore, India, Mar. 2003.

[15] Y. Yao and J. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. *ACM SIGMOD Record*, 31(3):9–18, Sept. 2002.