Grid-based Data Stream Processing in e-Science*

Richard Kuntschke[1] Tobias Scholl[1] Sebastian Huber[1] Alfons Kemper[1] Angelika Reiser[1]
Hans-Martin Adorf[2] Gerard Lemson[3] Wolfgang Voges[3]

[1]Lehrstuhl Informatik III: Datenbanksysteme
Fakultät für Informatik, Technische Universität München
Boltzmannstraße 3, D-85748 Garching, Germany
⟨*firstname.lastname*⟩@in.tum.de

[2]Max-Planck-Institut für Astrophysik
Karl-Schwarzschild-Straße 1, D-85748 Garching, Germany
adorf@mpa-garching.mpg.de

[3]Max-Planck-Institut für extraterrestrische Physik
Giessenbachstraße 1, D-85748 Garching, Germany
{gerard.lemson | whv}@mpe.mpg.de

Abstract

The field of e-science currently faces many challenges. Among the most important ones are the analysis of huge volumes of scientific data and the connection of various sciences and communities, thus enabling scientists to share scientific interests, data, and research results. These issues can be addressed by processing large data volumes on-the-fly in the form of data streams and by combining multiple data sources and making the results available in a network. In this paper, we demonstrate how e-science can benefit from research in computer science in the field of data stream management. In particular, we are concerned with processing multiple data streams in grid-based peer-to-peer (P2P) networks. We introduce spatial matching*, which is a current issue in astrophysics, as a real-life e-science scenario to show how a data stream management system (DSMS) can help in efficiently performing associated tasks. We describe our new way of solving the spatial matching problem and present some evaluation results. In the course of the evaluation, our DSMS* StarGlobe *proves to be a valuable computing platform for astrophysical applications.*

1. Introduction

Information fusion across various data sources is an important task in many e-science applications. In this course, transmitting all the necessary data from the data sources to the data sink for processing (data shipping) is problematic and in many cases will not be feasible any more in the near future due to the large and increasing data volumes. Executing operators that reduce data volume at or near the data sources (query shipping) or distributing query processing operators in a network (in-network query processing) are promising solutions to this problem. In-network query processing, as employed in our *StarGlobe* system, can also be combined with parallel processing and pipelined processing of data streams. This enables further improvements of performance and response times in e-science workflows.

Throughout this paper, we use *spatial matching*, which is a current issue in astrophysics, as an example e-science scenario to describe and evaluate our approach. Spatial matching is an important step in the process of determining *spectral energy distributions (SEDs)* of celestial objects. SED assembly and subsequent classification are rather complex problems [3]. In order to discover and classify new astronomical objects like active galactic nuclei, brown dwarfs, and neutron stars, it is not sufficient to just survey the sky using one specific observation method. Rather, photometric data from various wave bands and catalogs, i. e., data archives, has to be combined to gain ideally seamless SEDs of celestial objects. Due to the distribution of catalogs over various locations, the potentially large data volumes, and the need for often complex data transformations, this scenario poses a real challenge. One of the greatest difficulties is the fact that there is no unique (database) key for the identification of astronomical objects. Instead, the only way of identifying these objects is by using *uncertain* sky positions.

There exist only few tools, such as the GAVO (German Astrophysical Virtual Observatory) *crossmatcher* [4], that are able to perform a cross-matching of catalogs which serves as a basis for subsequent SED assembly and classification. During the process, the GAVO crossmatcher acquires astrometric and photometric data from catalogs covering different wavelength ranges and loads it into a main memory database. Afterwards, the crossmatcher applies various transformations to spatially match astrometric data using statistical methods. Due to the combinatorial explosion during a multi-catalog join, the amount of data can grow so rapidly that it exceeds main memory and cannot be handled anymore.

In this paper, we present a solution to this problem using our *StarGlobe* system. StarGlobe is a grid-based

*This research is supported by the German Federal Ministry of Education and Research within the D-Grid initiative under contract 01AK804F and by Microsoft Research Cambridge under contract 2005-041.

P2P data stream management system (DSMS) which is based on *StreamGlobe* [16, 20] and implemented on top of Globus [10]. StarGlobe augments StreamGlobe with functionality specifically tailored to the needs of the astrophysics community. In StarGlobe, data from multiple catalogs is streamed into a grid-based P2P network, i. e., a P2P network with peers implemented as grid services, and processed in a pipelined fashion. The system is able to transform and spatially match large volumes of astrometric data. In addition, the system dramatically reduces the time needed to produce and return the first result data item. Therefore, first results can be seen after a relatively short period of time due to pipelined stream processing. This is much more convenient for astronomers than having to wait for the whole process to complete before receiving any results as in the straightforward approach.

Using DSMSs and exploiting their benefits is an important step in coping with the challenges of the near future. The amount of data in astronomy is growing at an exponential rate. Soon, observational data collected by satellites and telescopes during certain sky observations, e. g., in the LOFAR project [17], will exceed available storage capacities. Also, the traditional way of collecting and storing all observation data and analyzing it afterwards may not be appropriate for future applications. Therefore, alternative ways of processing and analyzing data *on-the-fly* as in StarGlobe are necessary.

To the best of our knowledge, this is the first work to investigate the impact of employing a grid-based distributed data stream management system for supporting and improving an actual (astrophysical) e-science workflow.

2. Problem Definition

A common problem in astrophysics is *SED assembly* and subsequent *classification* [3]. The basic idea is to construct an approximate SED for a celestial object by combining photometry from various catalogs covering different wavelength ranges and to classify the object on the basis of its SED. One problem with this is that measurement results usually are (in some cases also geographically) distributed over many large database tables called *catalogs*. Also, since the measurement results are obtained through various sky observations conducted using different measurement instruments, they contain individual variances due to a variety of reasons, including measurement uncertainties of the instruments used. These uncertainties concern the celestial coordinates, i. e., the positions (astrometry), as well as the photometric data, i. e., the intensities (photometry), of the object under investigation. Therefore, it is very likely that the coordinates of the same object slightly differ from each other in different catalogs. This makes it non-trivial to perform *spatial matching*, i. e., combine ("cross-match") corresponding objects or even whole catalogs, which is a prerequisite for SED assembly.

Solutions such as the GAVO crossmatcher [4] load the results of each queried catalog into a main memory database and then combine[1] and filter the results there (in-memory matching). A major disadvantage of this data shipping approach is the fact that it can easily lead to a memory overflow when matching too many sources, particularly when many catalogs are involved. Besides, scientists initiating a spatial matching for SED assembly have to wait for the crossmatcher to finish completely before receiving any results. Thus, they are unable to gain immediate feedback concerning the correctness of the specified parameters. We address these issues by an alternative approach using and enhancing the stream processing capabilities of a grid-based data stream management system.

3. The SED Scenario

This section gives an explanation and a summary of the mechanism of SED assembly and classification to the extent necessary for understanding this paper.

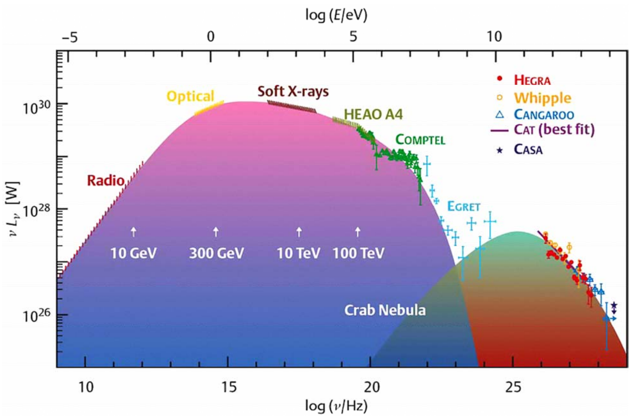
3.1. Overview

As motivated in a recent astrophysical publication [3], SED assembly and classification are essential exploratory techniques to discover new astronomical objects like obscured active galactic nuclei, brown dwarfs, isolated neutron stars, or planetary nebulae. The key principle is the combination of multi-wavelength photometric data from different catalogs to assemble an ideally seamless SED of a celestial object. The workflow of SED classification can be divided into the following five steps: catalog query, spatial (or astrometric) matching, assembly of raw photometry, photometric transformation, and actual SED classification.

In the first step, astronomers usually create an "input list" of astronomical objects they are interested in. The celestial coordinates of the objects could be derived, e. g., from a catalog which the astronomer wants to use as a starting point. Besides the coordinates, the list also contains an identifier for each object. The input list is then used to query various catalogs covering as many wavelength ranges as needed. For each object in the input list, a (simple) cone search is conducted on each catalog. The result of the search together with the *id* of the corresponding object from the input list is returned in a so-called *primary match result (PMR)*, together with the associated photometric data. The PMRs are the basis for the astrometric matching [2] performed in the second step which is explained in detail in Section 3.2.

When the astrometric matching is complete, photometric data has to be assembled and transformed into an SED in the third and fourth step. Photometric transformation (the actual SED assembly) is a non-trivial process which is not addressed in this paper. Figure 1 illustrates what the results of a photometric transformation could look like by the example of the emissions from the Crab Nebula, shown in Figure 2, in the constellation Taurus. Alternatively to the

[1] A left outer join ⟕ is computed over the input list of objects, which in theory could be a complete catalog by itself, and all primary match results.



Source: [22]

Figure 1. Spectral energy distribution of the emissions from the Crab Nebula

Source: [22]

Figure 2. The Crab Nebula in the constellation Taurus in optical light

photometric transformation, so-called *ratios* of photometric values can be calculated to filter characteristic objects (e. g., "radio-loud" objects: $F_{radio}/F_{optical} > ratio_{threshold}$).

After photometric transformation, the resulting SED can be fed into a statistical classifier for supervised classification. This final step performs photometric matching with so-called *template SEDs*, which may be synthesized based on a library of template spectra [3]. The results of this classification can then be reviewed and evaluated by astronomers.

The emphasis of our work lies on the spatial matching in the second step.

3.2. Spatial (Astrometric) Matching

After having completed the first step of the SED assembly and classification workflow, i. e., all PMRs of each queried catalog have been collected, the secondary match can be performed. This match is "fuzzy" since a statistical method of result filtering is applied (*fuzzy match*). The fuzzy match itself consists of a deterministic and a statistical part. The deterministic part is basically a left outer equijoin over the *id* attribute (indicated by $⟕_{id}$) of all PMRs $P_1, \ldots, P_n$ including the input list I on the very left side:

$$(((I ⟕_{id} P_1) ⟕_{id} P_2) ⟕_{id} \ldots) ⟕_{id} P_n$$

Here, *id* is an attribute introduced in I to identify objects in all PMRs related to one *id* in I.

The "fuzzy" part takes the results of the join and calculates a statistical metric, hypothetically assuming that all objects within one join result, also denoted as *counterparts*, belong to the same physical source. This metric is subsequently referred to as the *reduced* χ^2*-metric* [2, 4]. Figure 3 illustrates two join results or "spiders", each consisting of three counterparts from three different catalogs that obviously fit together forming two *match candidates*. Match candidates with a χ^2 value above a certain user-defined threshold are discriminated because they are statistically unreasonable match combinations. In Figure 4, one counterpart is associated with both match candidates. The spider in the lower left corner seems to be quite reasonable. But the spider in the upper right corner connecting the stray counterpart with a dashed line obviously does not make sense and can be dropped because its reduced χ^2 is too high. One can view the $\chi^2_{reduced}$-metric as a measure of *compactness* of a spider or match candidate.

The $\chi^2_{reduced}$ value is calculated for each match candidate as follows:

1. Calculate the weighted mean (Euclidean) coordinates of all n counterparts within one match candidate (assuming that the two-dimensional spherical coordinates of each object have already been transformed into three-dimensional Euclidean coordinates lying on the unit sphere):

$$\vec{m} = \frac{1}{w} \cdot \sum_{i=1}^{n} (w_i \vec{v}_i) , \qquad (1)$$

where $w_i = 1/\sigma_i^2$ is the statistical weight factor for each counterpart $i \in [1..n]$, $w = \sum_{i=1}^{n} w_i$ is the sum of all weight factors, σ_i is the astrometric uncertainty associated with each counterpart i, and $\vec{v}_i = (x_i, y_i, z_i)^T$ are the Euclidean coordinates of a counterpart i.

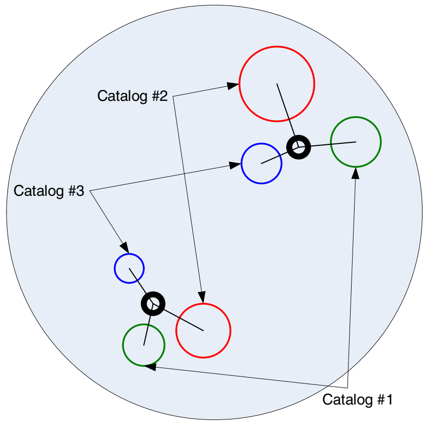
2. Normalize the Euclidean coordinates of the weighted vector $\vec{m}$ to the unit sphere:

$$\vec{\overline{m}} = \frac{\vec{m}}{\sqrt{x_m^2 + y_m^2 + z_m^2}} , \qquad (2)$$

with $\vec{m} = (x_m, y_m, z_m)^T$.

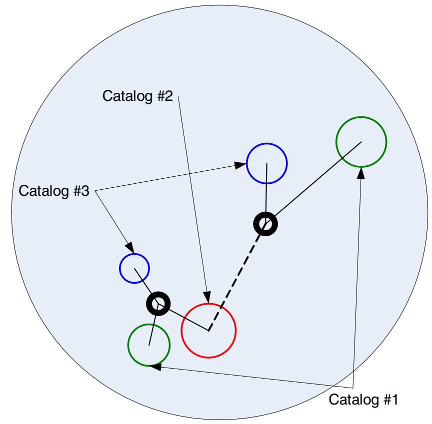
3. With $\vec{\overline{m}}$ as "optimum center" (see small thick black circles in Figures 3 and 4 which represent the formal astrometric uncertainty) now calculate arc distances between each counterpart i and the optimum center $\vec{\overline{m}}$:

$$\varphi_i = arccos(\vec{v}_i \cdot \vec{\overline{m}}) = arccos(x_i x_{\overline{m}} + y_i y_{\overline{m}} + z_i z_{\overline{m}}) , \quad (3)$$



Source: [2]

Figure 3. Two "spiders"



Source: [2]

Figure 4. Two "spiders" with stray counterpart

with $\vec{\overline{m}} = (x_{\overline{m}}, y_{\overline{m}}, z_{\overline{m}})^T$.

4. The Mahalanobis distance r_i between each counterpart i and the center $\vec{\overline{m}}$ then is:

$$r_i = \frac{\varphi_i}{\sigma_i} . \tag{4}$$

5. χ^2 is calculated from:

$$\chi^2 = \sum_{i=1}^{n} r_i^2 . \tag{5}$$

6. And finally the reduced χ^2 value is calculated by dividing χ^2 by the *degrees of freedom* (per spider):

$$\chi^2_{reduced} = \frac{\chi^2}{2n-2} . \tag{6}$$

This sketch of the calculation steps should be sufficient for the understanding of the distributed spatial matching scenario presented in this work. More detailed information, e. g., more information on the degrees of freedom, can be found in the description of the GAVO crossmatcher [2].

4. Astrometric Matching in StarGlobe

In this section, a spatial matching scenario which has been implemented and executed on our grid-based distributed data stream management platform StarGlobe and which is the basis for SED assembly and classification is presented together with the corresponding results.

4.1. Preliminaries

Spatial matching is an important step in the process of SED assembly and classification. PMRs from various catalogs are cross-combined, which can lead to an enormous growth of data volumes. The challenge now is to filter out those combinations that do not fit together and select only "valid" match candidates for further processing and classification. Straightforward approaches like the GAVO crossmatcher calculate all possible combinations of PMRs en bloc in main memory and subsequently filter only good match candidates with the $\chi^2_{reduced}$-metric. As already mentioned in Section 1, this approach will run out of main memory when matching too many catalogs and/or sources.

With StarGlobe being a distributed system, a different very promising approach can be taken. By distributing the combination phase of the spatial matching over multiple peers, processing can be parallelized. Many more PMRs can be joined at the same time and therefore, more catalogs can be included to produce match candidates containing even more information.

In addition, $\chi^2_{reduced}$-filtering, which is usually done *after* the deterministic matching in the workflow of SED assembly, is split up and relocated at the join operators of the deterministic matching process as shown in Figure 5. This results in a tight integration of the deterministic as well as the statistical process, improving selectivity and preventing bad match candidates from yielding unnecessary combinations at a very early stage. Also, network traffic is reduced and throughput of valid match candidates at single peers is increased. But one has to be careful not to filter out match candidates too early when their $\chi^2_{reduced}$ value lies slightly above the threshold. At first glance, these are bad match candidates. But as long as further counterparts could join a match candidate, the match candidate's $\chi^2_{reduced}$ value could drop below the threshold. This may happen when an existing match candidate is joined by another counterpart whose coordinates lie near or exactly on the optimal center of the spider. This would contribute to the "compactness" of the spider, such that its $\chi^2_{reduced}$ value decreases. Therefore, thresholds on filter operators placed at inner nodes have to be specified more generously. To assure that no possible good match candidate is dropped, the local χ^2 is divided by the maximum degree of freedom a match candidate can reach throughout the assembly process. This degree depends on the number of catalogs being spatially matched.

CATALOG	SPECTRAL BAND	# OBJECTS	FULL NAME
2MASS	near-infrared	470,992,970	Two Micron All Sky Survey
FIRST	radio	811,117	Faint Images of the Radio Sky at Twenty centimeters
GSC-2	optical	455,851,237	The Guide Star Catalog Version 2.2
NVSS	radio	1,773,484	1.4 GHz National Radio Astronomy Observatory Very Large Array Sky Survey
RASS-BSC	X-ray	18,806	ROSAT All-Sky Survey Bright Source Catalog 1RXS (1st ROSAT X Survey)
USNO B1.0	optical	1,045,913,669	Whole-Sky United States Naval Observatory B1.0 Catalog

Table 1. Catalogs used in spatial matching scenario

CATALOG	SEARCH RADIUS	TABLE	# PMRS	STREAM SIZE
2MASS	90 arcsec	II/246/out	611	229 KB
FIRST	90 arcsec	VIII/71/first	60	27 KB
GSC-2	90 arcsec	I/271/out	722	350 KB
NVSS	90 arcsec	VIII/65/nvss	32	14 KB
USNO B1.0	90 arcsec	I/284/out	666	260 KB

Table 2. Catalogs queried using an input list of RASS-BSC sources

Another feature of the distributed approach is that first results are returned rather quickly during processing in contrast to the straightforward approach where no results are returned as long as the calculation proceeds. This is due to the fact that StarGlobe constitutes a DSMS using non-blocking query operators which process data streams on-the-fly in a pipelined fashion.

In order to support astrophysical scenarios like the following, StarGlobe has been augmented with additional stream iterator implementations [11]. Stream iterators in StarGlobe are basically non-blocking operators for processing data streams. The additional operators comprise a *sigma enricher* for attaching necessary uncertainty information to data stream items, a generic, non-blocking, progressive *merge join* operator used for matching data items, and a $\chi^2_{reduced}$-*filter* operator for performing the filtering as described in Section 3. To realize real-life spatial matching scenarios for SED assembly, PMRs have been acquired from different catalogs shown in Table 1.

4.2. Spatial Matching Scenario

In our spatial matching scenario for SED assembly, we use 50 sources from RASS-BSC [23] as input list. Since there does not yet exist a module to directly query the VizieR catalog service [21] from Centre de Données astronomiques de Strasbourg (CDS) [6] and to stream the results automatically into StarGlobe, the PMRs are currently retrieved manually and converted to XML. The queried catalogs are shown in Table 2. The search radii used for querying the catalogs are quite large, yielding a comparatively large number of PMRs. This is due to the coordinates of the source objects taken from RASS-BSC having rather large uncertainties. The density of a catalog also has an impact on the result set. The higher the density, the more objects are returned within the search radius.

In Figure 6, the grid-based network topology which is used for this spatial matching scenario is shown. At *peer-1* to *peer-5*, the PMRs are streamed into the network. The input list is injected at *peer-0*. In this scenario, the input list itself is included in the matching process, which is possible in SED assembly [2]. Note that the left outer join as shown in Section 3.2 has to be split up and distributed over several peers within the network. Thus, a transformation of the n-way left outer join becomes necessary to make sure no tuple (or match candidate) is accidentally dropped. The basic principle of the transformation is that left outer joins on certain inner nodes have to be replaced by full outer joins. The transformation is based on the following theorem.

Theorem 4.1. *Let A, B, and C be relations and let A.id, B.id, and C.id be their corresponding join attributes, respectively. Then the following applies:*

$$(A \ltimes\!\!\!\rtimes_{id} B) \ltimes\!\!\!\rtimes_{id} C \equiv A \ltimes\!\!\!\rtimes_{id} (B \ltimes\!\!\!\rtimes_{id} C) \qquad \square$$

PROOF. The proof can be found in [11]. ∎

In our example, the join

$$((((I_{RASS-BSC} \ltimes\!\!\!\rtimes_{id} P_{2MASS}) \ltimes\!\!\!\rtimes_{id} P_{FIRST}) \ltimes\!\!\!\rtimes_{id} P_{USNOB1}) \ltimes\!\!\!\rtimes_{id} P_{NVSS}) \ltimes\!\!\!\rtimes_{id} P_{GSC-2}$$

is transformed to

$$((I_{RASS-BSC} \ltimes\!\!\!\rtimes_{id} P_{2MASS}) \ltimes\!\!\!\rtimes_{id} (P_{FIRST} \bowtie\!\!\!\!\!\!\;\supset\!\!\!\subset_{id} P_{USNOB1})) \ltimes\!\!\!\rtimes_{id} (P_{NVSS} \bowtie\!\!\!\!\!\!\;\supset\!\!\!\subset_{id} P_{GSC-2}) .$$

The transformed join order is reflected by the network topology shown in Figure 6.

To demonstrate how important early filtering during spatial matching is, Table 3 shows two benchmarks. The first one is performed with and the second one without early $\chi^2_{reduced}$-filtering. The stream size represents the intermediate size of an output stream together with the number of match candidates directly after each join operator (see Figure 5), i. e., after the join but before the following filter operator in case of early filtering. Early filtering keeps the

	With early filtering		Without early filtering	
	Stream size	# Match candidates	Stream size	# Match candidates
After join-0	808 KB	611	808 KB	611
After join-1	1,874 KB	1,138	1,874 KB	1,138
After join-2	1,387 KB	826	1,387 KB	826
After join-3	6,355 KB	2,522	46,525 KB	15,489
After join-4	14,356 KB	3,815	1,838,648 KB	364,299
After filter-4	1,364 KB	318	1,364 KB	318
Duration h:m:s	00:02:58		02:46:00	

Table 3. Results

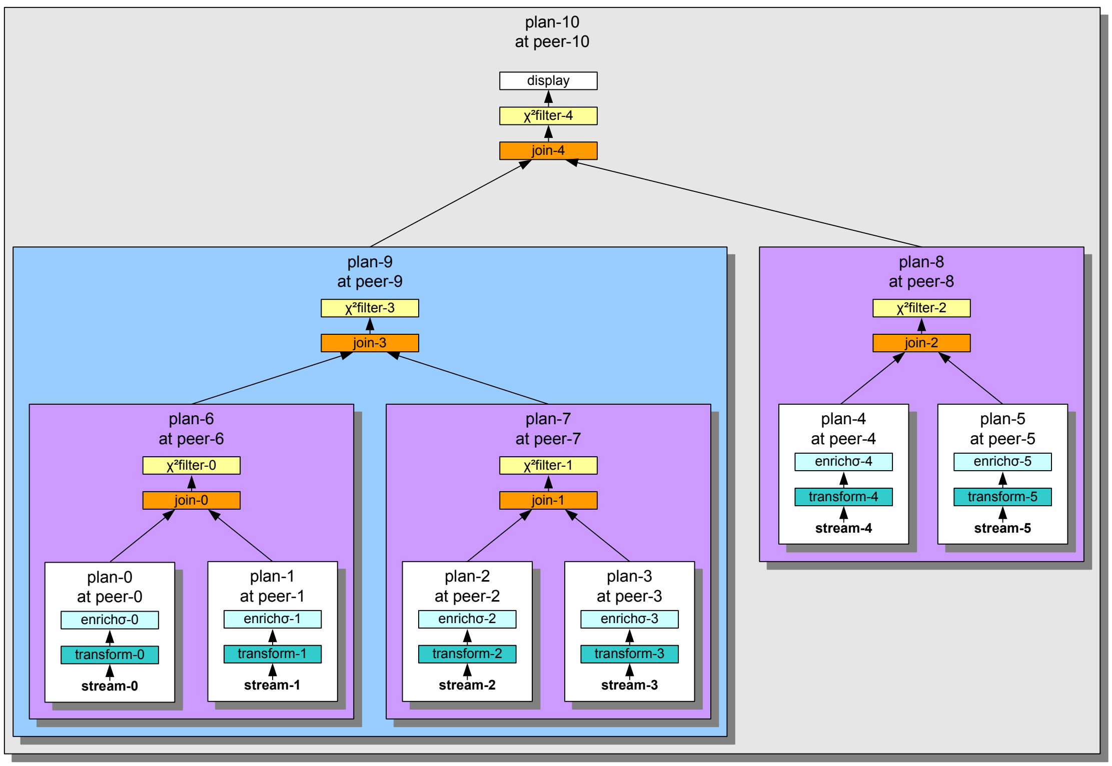


Figure 5. Query plan

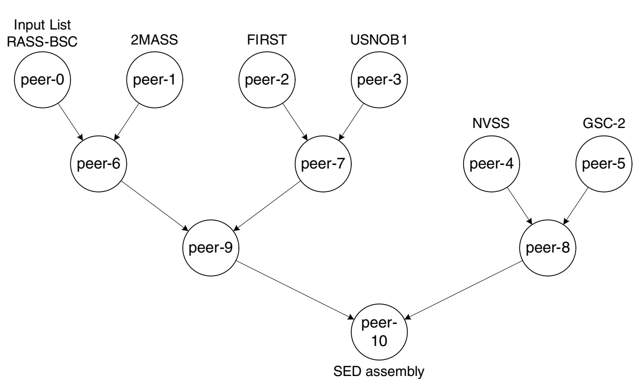


Figure 6. Network topology

number of match candidates considerably lower after *join-3* and *join-4* compared to the approach without early filtering. Since intermediate result sets are much smaller, performance with early filtering is much better. Without early $\chi^2_{reduced}$-filtering, large intermediate results with many irrelevant match candidates are generated and only filtered after the last join. Therefore, 99.9% of all match candidates in the result of *join-4* are dropped at *filter-4* in that case. This can be seen from comparing stream sizes after *join-4* and *filter-4* in Table 3. The loss of performance is obvious when comparing the durations[2] of both benchmarks. With early $\chi^2_{reduced}$-filtering, this astrometric matching scenario runs more than *50 times* faster than without!

[2]Durations may vary on different hardware. The benchmark has been conducted on a blade server using 11 blades, each equipped with an Intel Pentium IV XEON processor at 2.8 GHz and 1 GB of RAM.

	# MATCH CANDIDATES		FILTER RATIO
	Before filtering	After filtering	
At join-0	611	289	47.3%
At join-1	1,138	452	39.7%
At join-2	826	458	55.4%
At join-3	2,522	400	15.9%
At join-4	3,815	318	8,3%

Table 4. Filter ratios with early filtering

Table 4 shows the number of match candidates before and after a $\chi^2_{reduced}$-filter operator at every join node in the network. Additionally, the resulting filter ratio (selectivity) at each $\chi^2_{reduced}$-filter is shown.

Summarizing, our approach has proven to be higly beneficial in the presented scenario. It vastly reduces the negative effects of input coordinates with high uncertainties on overall performance by means of early filtering of intermediate results. Also, even for high quality input coordinates with small uncertainties, parallel and pipelined streaming execution of workflow operators further improve performance and increase convenience through delivering first results early on during processing.

5. Related Work

StarGlobe is a version of the StreamGlobe [16, 20] DSMS featuring application-specific functionality to suit the needs of the astrophysics community. StreamGlobe itself serves as a research platform for investigating new techniques for distributed data stream management, processing, and optimization. One of the main research directions in StreamGlobe is *data stream sharing*, an optimization technique reducing resource consumption, e. g., network traffic and peer load, in the network of a distributed DSMS [14, 15]. Queries in StreamGlobe are processed using the *FluX* query engine [12, 13]. FluX is a query processor for continuous queries over XML data streams. It uses schema information of data streams to minimize memory consumption and is able to execute window-based queries over possibly infinite data streams.

The functionality presented in this paper is also provided by the GAVO crossmatcher [4]. However, as has been pointed out before, the approach pursued in the crossmatcher does not use early filtering, parallelization, and pipelined stream processing. It therefore imposes strict limits on the sizes of computable problems because of excessive resource consumption. If main memory is not large enough to hold all the necessary data, this approach becomes infeasible. Also, the entire result is delivered to the user at the end of the processing which can take considerable time to complete. Our StarGlobe approach reduces processing time and memory consumption drastically by using parallel and pipelined stream processing and delivers the first result item immediately after it has been generated.

The SkyQuery system [18] and its current redesigned version OpenSkyQuery [19] are federated databases which are based on web service technology. They enable users to query distributed catalogs called (Open)SkyNodes and to perform crossmatching of various catalogs in a similar way as presented in this paper. However, OpenSkyQuery currently has some limitations, e. g., in the number of rows that can be returned in an answer to a query. This is due to performance issues since, in contrast to our approach, OpenSkyQuery does not yet employ optimization techniques like parallelization and pipelined stream processing. We have recently started an active collaboration with Alex Szalay and his group to investigate possibilities of integrating our work with theirs.

In the past, many different data stream management systems have been proposed [1, 5, 7, 8, 24]. In particular, GATES [9] provides a grid-based middleware for processing data streams in a distributed fashion. But, as far as we know, none of these systems has been used to specifically support real-life e-science applications. To the best of our knowledge, this is the first work to investigate the impact of employing a grid-based distributed data stream management system for supporting and improving an actual (astrophysical) e-science workflow.

6. Conclusion

This work demonstrates the possible benefits of combining research efforts in computer science and other scientific disciplines, e. g., astrophysics, for the scientific community. Recent research efforts in computer science in the field of data streams enable new solutions for existing problems. StarGlobe, as presented in this paper, is a new platform for modeling actual astrophysical e-science scenarios. This is demonstrated by the successful installation of the astrometric matching scenario as a prerequisite for the SED assembly and classification process, which is an important task in astrophysics today. In this scenario, extensive processing of multiple data streams is combined with the application of possibly complex transformations and statistical methods in order to efficiently cross-match astrophysical data catalogs. The result data stream can be subscribed at any peer connected to the grid-based StarGlobe network and used for further processing, e. g., for supervised classification to discover new astronomical objects like, for example, obscured neutron stars or hidden galaxies.

The research community benefits from our approach in many ways. First, larger problem sizes can be handled without running into problems concerning available computing resources, especially in terms of main memory. With early filtering and parallelization techniques to reduce memory consumption and processing time, StarGlobe is prepared for the challenges of the anticipated data explosion of the next decade. Second, using pipelined stream processing, e-science workflows can deliver first results early on in a pipelined fashion while the computation of the remaining results is still running. This enables scientists to check the correctness of their parameter settings and to start working on the results early on. Of course, our approach is also applicable to many other matching scenarios in all fields of science, business, and engineering.

As far as future work is concerned, automatic retrieval of PMRs would eliminate the need to query astronomical catalogs like VizieR/CDS manually. Within the scope of the development of the GAVO crossmatcher, our cooperation partners in astronomy from the Max-Planck-Institut für extraterrestrische Physik (MPE) have developed a component which performs the PMR acquisition. They intend to make the component available for integration into StarGlobe in the future. As an intelligent content provider, the adapted component could then be used to query various remote catalogs in parallel and stream the results into the StarGlobe network. Furthermore, scientific workflows are currently specified in the form of manually written XML documents in StarGlobe. This workflow description contains specifications of operators, operator placement on peers, and the data flow that has to be established between operators and peers. An optimizer for automatic plan generation, e. g., from queries formulated in the astronomical data query language (ADQL), would relieve researchers from having to write their own query plans which is desirable especially for complex scenarios.

References

[1] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Çetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik. The Design of the Borealis Stream Processing Engine. In *Proc. of the Conf. on Innovative Data Systems Research*, pages 277–289, Asilomar, CA, USA, Jan. 2005.

[2] H.-M. Adorf. The GAVO crossmatcher – features and algorithms. Unpublished manuscript, July 2005.

[3] H.-M. Adorf, F. Kerber, G. Lemson, A. Micol, R. Mignani, T. Rauch, and W. Voges. Assembly and Classification of Spectral Energy Distributions – A new VO Web Service. In *Proc. of the Conf. on Astronomical Data Analysis Software & Systems*, pages 365–369, Pasadena, CA, USA, Oct. 2004.

[4] H.-M. Adorf, G. Lemson, and W. Voges. The GAVO Cross-Matcher Application. In *Proc. of the Conf. on Astronomical Data Analysis Software & Systems*, pages 695–698, San Lorenzo de El Escorial, Spain, Oct. 2005.

[5] 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.

[6] Centre de Données astronomiques de Strasbourg, Sept. 2006. http://cdsweb.u-strasbg.fr.

[7] 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.

[8] 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.

[9] L. Chen, K. Reddy, and G. Agrawal. GATES: A Grid-Based Middleware for Processing Distributed Data Streams. In *Proc. of the IEEE Intl. Symp. on High-Performance Distributed Computing*, pages 192–201, Honolulu, HI, USA, June 2004.

[10] The Globus Alliance, Sept. 2006. http://www.globus.org.

[11] S. Huber. Grid-based Processing of Multiple Data Streams in E-Science. Master's thesis, Technische Universität München, Oct. 2005.

[12] C. Koch, S. Scherzinger, N. Schweikardt, and B. Stegmaier. FluXQuery: An Optimizing XQuery Processor for Streaming XML Data. In *Proc. of the Intl. Conf. on Very Large Data Bases*, pages 1309–1312, Toronto, Canada, Aug. 2004.

[13] 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.

[14] R. Kuntschke and A. Kemper. Data Stream Sharing. In *Proc. of the Intl. Workshop on Pervasive Information Management*, pages 769–788, Munich, Germany, Mar. 2006.

[15] R. Kuntschke, B. Stegmaier, and A. Kemper. Data Stream Sharing. Technical Report TUM-I0504, Technische Universität München, Apr. 2005.

[16] R. Kuntschke, B. Stegmaier, A. Kemper, and A. Reiser. StreamGlobe: Processing and Sharing Data Streams in Grid-Based P2P Infrastructures. In *Proc. of the Intl. Conf. on Very Large Data Bases*, pages 1259–1262, Trondheim, Norway, Aug. 2005.

[17] LOFAR, Sept. 2006. http://www.lofar.org.

[18] T. Malik, A. S. Szalay, T. Budavári, and A. Thakar. SkyQuery: A Web Service Approach to Federate Databases. In *Proc. of the Conf. on Innovative Data Systems Research*, Asilomar, CA, USA, Jan. 2003.

[19] W. O'Mullane, T. Budavári, N. Li, T. Malik, M. A. Nieto-Santisteban, A. S. Szalay, and A. R. Thakar. OpenSkyQuery and OpenSkyNode – the VO Framework to Federate Astronomy Archives. In *Proc. of the Conf. on Astronomical Data Analysis Software & Systems*, pages 341–345, Pasadena, CA, USA, Oct. 2004.

[20] 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.

[21] The VizieR Catalogue Service, Sept. 2006. http://vizier.u-strasbg.fr.

[22] H. Völk. Gamma-Astronomie mit abbildenden Cherenkov-Teleskopen – Erste Ergebnisse und Pläne für die Zukunft. *Sterne und Weltraum*, pages 1064–1070, Dec. 1999.

[23] W. Voges, B. Aschenbach, T. Boller, H. Bräuninger, U. Briel, W. Burkert, K. Dennerl, J. Englhauser, R. Gruber, F. Haberl, G. Hartner, G. Hasinger, M. Kürster, E. Pfeffermann, W. Pietsch, P. Predehl, C. Rosso, J. H. M. M. Schmitt, J. Trümper, and H. U. Zimmermann. The ROSAT All-Sky Survey Bright Source Catalogue. *Astronomy and Astrophysics*, 349(2):389–405, July 1999.

[24] Y. Yao and J. Gehrke. Query Processing for Sensor Networks. In *Proc. of the Conf. on Innovative Data Systems Research*, Asilomar, CA, USA, Jan. 2003.