Experiments on Network Services for the Grid

Christoph Barz, Peter Martini, Markus Pilz, Florian Purnhagen
University of Bonn
Institute of Computer Science IV
Bonn, Germany
Email: {barz, martini, pilz, purnhage}@cs.uni-bonn.de

***Abstract*—For demanding Grid applications that use resources at different cluster sites a reservation mechanism ensures the availability of resources to guarantee a certain level of QoS. Besides computational and storage resources we see the network as a reservable resource. Our approach to QoS is to book resources in the network for exclusive usage. For this purpose, a flexible reservation mechanism is needed to meet the requirements of Grid applications and schedulers. In this paper we introduce a wider notion of network service than previous work. Special focus is laid on a file transfer service that guarantees the timely availability of data at the different Grid sites. To cope with these requirements, different strategies and algorithms for a flexible network reservation service are evaluated systematically. Algorithms using a single path strategy are compared to algorihms using multiple paths. Our simulation results show that data transport via multiple paths can lead to a higher user and network provider satisfaction than single path transfers.**

I. Introduction

Research communities in areas like particle physics, numerical weather prediction, and bioinformatics use scientific applications with demanding needs for computational resources. A common work pattern is the analysis of huge data sets produced by large-scale simulations or scientific instruments. Supercomputers, compute clusters, and mass storage systems provide a starting point to allow for these scientific applications. A trend in the last years is to make these costly computational resources available to researchers around the globe via the Grid [1]. This trend is driven by national and international Grid activities (e.g. UK Grid, D-Grid, DEISA, egee). Beside universities and research institutes, commercial providers are starting to make mass storage and compute resources available (cf. Amazon's S3 and EC2).

Consequently, scientific applications can effectively use a set of compute clusters and mass storage systems available in a computational Grid. In order to support applications with a need for coordination of resources at different sites, advance reservations with dedicated quality of service (QoS) are used [2]. As local compute resource managers already provide support for coordinating a workflow of distributed applications by means of advance reservation (e.g. EASY, LSF, Maui, PBS Professional), the timely delivery of data sets needs to be regarded as well as network connectivity with a certain QoS during application or workflow execution. This enlarges the scope to networking aspects for scientific applications.

This paper focuses on issues to provide users and applications with an additional benefit of the Grid infrastructure by taking a look at network services for Grid applications. Users or applications drive the selection and reservation of network connections with dedicated QoS. A special focus lies on a particular service: Delivering files on time, i.e. the transmission is finalized at a specified deadline. This service takes care of the transfer of large-sized data sets between sites, e.g. for the pre- and post-processing of Grid tasks.

Network QoS often focuses on aggregated streams of end-to-end micro flows. We believe that in the domain of scientific applications few high demanding flows are present which are either used for streaming/pipelining, MPI connections [3] or delivery of large-sized data sets. If few sustained high capacity flows are using network resources, QoS concepts as studied in this paper can be considered. The fundamental approach in this paper is to book resources in the network for exclusive usage. In day-to-day business booking of resources like seats, rooms and working time is a common task. The objective is that booked resources are available when required.

The rest of this paper is structured as follows: Section II presents an overview of the related work. Section III briefly introduces the architecture of our reservation system. Section IV focuses on single-path strategies while section V presents a multi-path strategy. Section VI compares both strategies by means of a simulative evaluation. The last section concludes the work and outlines future research.

II. Related Work

To the best knowledge of the authors the first architecture that delivered end-to-end quality of service (QoS) and advance network reservations for Grid applications was the Globus Architecture for Allocation and Reservation (GARA) [2]. It allows for the co-reservation and co-allocation of cluster and network resources for networks with RSVP signaling and DiffServ. In [4] Guérin and Orda evaluated the impact of advance reservation of network resources on the path selection process and introduced new reservation policies like maximum duration or soonest completion. However, they focused mainly on fixed capacity requirements for the reservations. In [5] Burchard extended this work and introduced the notion of malleable reservations, a variant of advance reservation mainly applicable to a file transfer service. With malleable

The work in this paper is partially funded by the German Federal Ministry of Education and Research (BMBF) through the VIOLA project under grant 01AK605L and by the European Commission under the 6^{th} Framework Programme (6FP) through the IST-Phosphorus project under grant 034115.

reservations, the duration of the reservation is dependent on the network capacity that is allocated for the reservation. Additional constraints like a deadline for the completion of the transfer set limits to the timing of data transmission. Burchard evaluates data structures as well as algorithms to process this kind of reservations. As network model MPLS is assumed.

Our work is carried out in the context of a real-life network reservation system which is being developed in the VIOLA [6] project. One challenging topic in VIOLA is to fill the gap between the services a high-speed network based on MPLS or GMPLS can provide and services Grid applications need: Dynamic provisioning of a particular QoS. Within VIOLA the development of the network resource reservation system ARGON [7] (Allocation and Reservation in Grid-enabled Optical Networks) has started. ARGON significantly widens the notion of network service beyond earlier work (c.f. section III). In this paper, different strategies and algorithms for a flexible service with a special focus on file transfers are evaluated systematically. Additional work on the network reservation system ARGON is carried out in the IST-Phosphorus [8] project. It addresses the challenge to enable on-demand end-to-end network services for Grid applications across multiple domains.

Our approach to QoS is to book resources in the network for exclusive usage or a certain share of the resources if exclusive usage is not possible or appropriate. However, when a certain amount of data is to be transferred during the reservation, there is a certain overhead due to the transport-protocol. For data transfers, TCP is the most popular transport-protocol in the Internet. It provides reliable transfer of data and a congestion control mechanism that controls the load imposed by IP packets sent by a host. As TCP is designed for data transmissions through the Internet it might not be favorable for communication via exclusively reserved resources. Experiments carried out in [9] show that standard TCP is not the best choice for communication via links with a high bandwidth delay product. This might lead to two approaches. The first approach is to predict the throughput of the standard TCP connections. In [10] two approaches are considered. One is a Formula-Based approach, the other is History-Based. Both estimators might work especially well in the context of exclusive resource usage. However, they might also rely on information about the source and sink of the transfer. The second approach is to use a transport mechanism that is better suited for the scenario of inter-cluster communication via dedicated, high capacity network resources. A candidate for this protocol is GridFTP [11] which is based on TCP. It uses parallelism via parallel and striped data transfers, is able to automatically negotiate the TCP buffer/window sizes while coordinating the transfer via collective operation. This approach may lead to a more efficient usage of the reserved network resources and a higher predictability. Another candidate is the transport protocol SCTP [12]. While this protocol was not explicitly designed for high capacity data transfers for the Grid, it also has a multi-streaming feature which allows for the partitioning of data into multiple, independent streams of data. Transmitting data via multiple independent streams also allows for a data transmission via multiple paths as described in section III-D. SCTP uses an adapted TCP congestion control mechanism.

The exact prediction of throughput is out of scope of this paper. In the following we will assume that either a component predicting the throughput for reserved network connections exists or that the efficiency is sufficiently high (e.g. via GridFTP).

III. Network Services for Grid Applications

This section describes the extension of a network resource reservation system for advance reservations to allow for a file transfer service supporting deadlines. Firstly, a system architecture of a network resource reservation system is presented. This system builds the basis for the file transfer service. Here, the underlying network technology has to support explicit routing of traffic flows. The following subsection describes data structures used to manage resources. Subsequently, strategies to meet file transfer requests are categorized and an outline of the scenarios for the simulative evaluation is presented

A. System Architecture

In order to meet the requirements of coordinated computation in an interconnected world, two types of reservations were identified: Immediate reservations and advance reservations. The latter is motivated by the fact that computing resources are booked in advance. Reservations describe certain QoS constraints that have to be mapped to services provided by the network. In order to provide advance reservations, a concept of time is needed. While options for QoS are present in many technologies, there is usually a lack of time-dependant information and configurations within forwarding entities and control plane concepts. Consequently, this task has to be done by the network resource reservation system. The challenge of advanced reservations is obvious: Without knowing the exact status of the network at future points in time it is difficult to decide whether a connection with a certain capacity can be accepted.

Network services that can be requested are the connection of sites, nodes, or particular networks in a point-to-point, point-to-multipoint or multipoint-to-multipoint manner with a dedicated QoS. Here, ARGON cooperates with two technologies: MPLS and GMPLS. Both technologies support concepts to perform allocation of network resources by means of the routing protocol OSPF with traffic engineering extension (OSPF-TE) and the signaling protocol RSVP-TE. OSPF-TE provides a way of describing the topology with traffic engineering options, e.g. capacity and administrative weights. MPLS Traffic Engineering (MPLS-TE) nodes usually allow for strict QoS guarantees, resource optimization, and fast failure recovery, but the current standard is limited to point-to-point connections.

There are different approaches to realize the establishment of a path that was precomputed by the network resource reservation system for a reservation: It is possible to use explicit route objects, which are supported by the MPLS and

GMPLS control plane. If explicit routes are established in a network domain exclusively used by advance reservations, interactions with additional traffic are avoided. On the other hand, the deployment of a Path Computation Server (PCS) is possible that queries the reservation system for the paths of reserved connections. In this paper we assume the former approach of explicit route objects.

Providing support for the envisioned file transfer service requires effort in different areas:

- resource management for advance reservations,
- admission control strategies and algorithms for file transfer requests,
- signaling of dedicated end-to-end paths with capacity constraints, and
- interaction between reservation system and file transfer applications.

Each area needs to be addressed in detail to establish the envisioned file transfer service. Here, topics like signalling paths with capacity constraints for point-to-point connections and getting topological information are only sketched. The paper mainly focuses on the first two topics as these address the core of an overall architecture.

B. Types of Reservation

A well-known type of reservation is used in the context of the public switched telephone network (PSTN): Immediate reservations. The basic parameters of an immediate reservation request are the source identifying the starting point (e.g. telephone, ingress router) and the destination (e.g. egress router) of the requested path. Additional constraints of the connection usually are implicit. The duration and therefore the ending time of the reservation is a priori unknown to the reservation system.

In the context of Grid computing another type of reservations is envisioned: Advance reservations. Reservations are planned in advance and the duration and capacity or the amount of data to be transmitted is known a priori. An advance reservation request for a source-destination pair (s, d) is defined as $r = (s, d, t_a, t_r, t_d, \mathcal{C})$, where t_a is the arrival time of the request, t_r is the release time (soonest start time possible), t_d is the deadline or due time of the request, and $\mathcal{C}$ specifies a set of constraints. Figure 1 shows the life cycle of an advance reservation.

When focusing on a file transfer service, the set of constraints $\mathcal{C}$ is reduced to a single parameter describing the file size, i.e. requests are given by $r = (s, d, t_a, t_r, t_d, v)$, where v specifies the amount of data to be transferred; The actual time for the file transfer may be one or more sub-intervals $[t_s, t_e[$ of $[t_r, t_d[$ with start time t_s and end time t_e and the transfer rate may change from interval to interval. Figure 1 shows the actual data transfer in $[t_s, t_e[$ with a duration of Δt still meeting the deadline t_d. In general, the release time and the deadline need not be tight, e.g. there might be more than one possible start time. This leaves room for the reservation system to efficiently plan the data transmission in different time slots with varying data rates. In this case the reservation system may require a feedback mechanism with the applications that transmit the data. As stated in the previous section, this mechanism is out of scope of this paper. In addition, the user or the application may define certain additional policies for the reservation, e.g. first-fit, best-fit, multi-path, single-path.

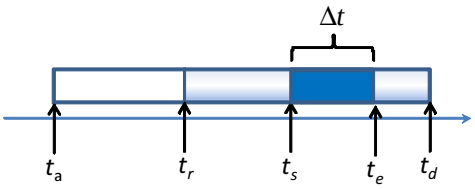

Fig. 1. Life Cycle of an Advance Reservation

C. Data Structures for Resource Management

Whenever a request is received by the reservation system, information about the allocation of resources for previously accepted requests is required in order to determine whether a request is feasible or not. In case of an advance reservation, it would be possible to request resources in the distant future. Generally, the reservation system needs to manage resource allocation information for an open-ended period. We assume a specific limit which bounds the interval for processing reservations. In the following, this interval is denoted as timeline $\mathcal{T}$ and the length of the interval is denoted as book-ahead.

A task of the resource information management is to keep track of residual capacities in the network topology. Every time a new reservation request is handled by the reservation system, it must check whether enough capacity is available to serve the processed request. In addition to the core topology information, it is necessary to manage information about the planned resource utilization in the network with respect to time. This information is available if every request regarding the network is handled by the reservation system.

The timeslot-based management of allocated resources is an established way to manage this information (cf. [2], [4], [13]). This approach is already used in different environments for advance reservations, e.g. denoted as timeslot table within GARA [14].

Using the concept of timeslot-based resource allocation management, the timeline $\mathcal{T}$ is divided into a sequence of timeslots $T_i = [t_i, t_{i+1}[, 0 \leq i \leq n-1$, where $[t_0, t_n[$ represents the timeline and $T_i \in \mathcal{T}$. In doing this, the reservation system manages utilization information individually for each timeslot, i.e. only accumulated values for every link are stored. Modeling the network topology as a capacitated, directed graph $G = (V, E, c_{max}), c_{max} : E \rightarrow \mathbf{R}^+$, the residual capacities of a link in a given timeslot are represented by $c_{res} : \mathcal{T} \times E \rightarrow \mathbf{R}_0^+$. When a new request is accepted by the reservation system, the residual capacities are updated.

In this paper we assume timeslots of dynamic length with a fixed granularity. For every new reservation, existing interval(s) may be split at the start time and the end time of

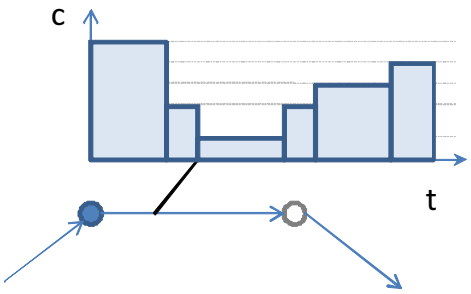

Fig. 2. Link annotated with the residual capacities within the Timeline

the reservation. Two different cases for the assigned starting and ending times can occur: Possibly, a slot boundary can be met where another accepted reservation or the timeline itself begins or ends. In this case, the timeline is already divided at this point in time. Otherwise, a point is met where no other reservation begins or ends. In this case, the corresponding timeslot must be divided at this point, as the resource reservation information is different on both sides of the new boundary. So, every time a new reservation is accepted, no more than two new timeslots can be created in addition to the already existing ones. Starting from the timeline as the first interval, this results in a timeline being segmented in at most $2k+1$ non-overlapping subintervals, where k is the number of accepted requests. It should be noted that even if the number of timeslots depends on the number of reservations, an upper bound is given by the length of the book-ahead and the fixed granularity.

Figure 2 sketches a timeline of a single link in the network managed by the reservation system. The x-coordinate represents the current timeline and the y-coordinate the residual capacities of the link. While the timeline is limited by the book-head, the capacities are limited by the maximum reservable capacity of a link (c_{max}). The different box widths demonstrate the usage of dynamic length timeslots, the different heights correspond to different levels of residual link capacity. The 6 timeslots depicted in figure 2 at 5 different levels of link capacity correspond to 4 or more accepted reservation requests that segmented the timeline.

D. Online Admission Control

The online admission control of the reservation system has to check whether reservation requests r can be realized by the residual capacities in the network given by c_{res}. Starting from dynamic timeslots with a fixed slotted granularity, a set of procedures is needed to handle requests. If a new request is processed, the reservation system has to determine all timeslots which are overlapping with the request. Subsequently, the residual capacities can be inspected on all links within these timeslots and the admission control is able to decide on the request. The result of the admission control is a set of reservation entities $R \subset \mathcal{T} \times P_{s,d} \times \mathbf{R}^+$, i.e. each reservation entity is determined by a timeslot ($\in \mathcal{T}$), a path from the source to the destination ($\in P_{s,d}$), and a capacity value ($\mathbf{R}^+$). Note that there may be more than one reservation entity per request. The reservation entities must be stored in the reservation system to allocate and release the resources at the specified points in time. Regarding the set of reservation entities the processing of file transfer requests span three inter-dependent domains:

- Spatial Domain: Which path(s) should be used to accept the request?
- Temporal Domain: Which time interval(s) should be used to accept the request?
- Capacity Domain: How much capacity should be used?

For a file transfer service supporting deadlines, the admission control has to find a set of reservation entities within these domains such that the sum of all capacity-duration products of the corresponding reservation entities equal the data amount specified by the request. Taking a look at the spatial domains, two possible strategies to process a request can be identified: (i) Single-Path: Only one path from the source to the sink is used to transfer data within a timeslot. (ii) Multi-Path: A set of paths from the source to the sink is allowed to transfer data within a timeslot.

Adding the temporal domain to the single-path and multi-path category leads to the question whether a spatial solution should be applied to all timeslots. Furthermore, adding the capacity domain the most contrary forms are the following:

- Constant Capacity Single-Path all Slots: The request uses the same path for the whole reservation time, i.e. the selected path is used for each timeslot. The reserved capacity between the source-destination pair is constant.
- Variable Capacity Multi-Path per Slot: The request can use a different set of paths for each timeslot. In addition, the reserved capacity between the source-destination pair is variable per timeslot.

We take these most contrary forms as a basis for an evaluation, the following two questions arise: Which algorithms can be used to perform an online processing of file transfer requests of the mentioned strategies? Is the mapping of a request on multiple paths favorable with respect to the user and provider satisfaction? Both questions are handled in the following sections.

E. Comparing Strategies

After setting the scene by outlining the system architecture, fundamental data structures, and strategies to process file transfer requests, the basis for a comparison of the algorithms described in sections IV and V is given. Basically, a set of data transfer requests R is handed to the reservation system in an online fashion, i.e. the system decides on a request without knowledge of upcoming requests. Intuitively, two basic metrics can be used to characterize the performance of the algorithms which correspond to two different optimization goals:

- User Satisfaction: Assuming that the user satisfaction is independent of the file size, accept as many requests as possible.

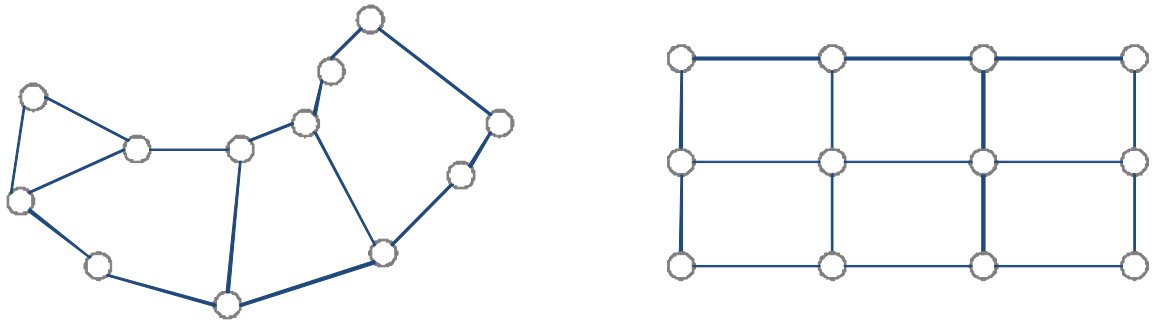

Fig. 3. Topology of the Abilene Backbone (left) and Grid Graph (3,4) (right)

- Network Provider Satisfaction: Maximize the profit, i.e. in a simple setting the file size describes the profit for the provider and the sum of the sizes of accepted files should be maximized.

The corresponding metrics are entitled Request Blocking Ratio (RBR) and Bandwidth Blocking Ratio (BBR). In the presented context of requests spanning a time period with a certain capacity, Volume Blocking Ratio (VBR) would also be appropriate in the latter case.

The RBR is defined as $|\bar{R}|/|R|$, where $|\bar{R}|$ denotes the cardinality of the set of rejected and $|R|$ the cardinality of the set of all requests presented to the reservation system. The Bandwidth or Volume Blocking Ratio can be defined in a similar fashion [15]. In this paper a basic job model is used in order to focus on the differences of path selection algorithms and differences of the single- and multi-path strategies. The requests generated in the simulation framework are restricted to a single file size, i.e. except of a constant factor the RBR is equivalent to the BBR. As a consequence, the performance of the algorithms is solely measured by the RBR. Obviously, the values of RBR can range from 0 to 1, where a smaller value represents a better performance. The generated requests are uniformly distributed between all source-destination pairs. We assume independent requests and model the inter-arrival time of the requests as being exponentially distributed.

In addition to the job model, topologies are an important parameter. In figure 3 two of the three topologies used in the scope of this paper are shown. The graph on the left side represents the Abilene core topology, which is a cross country backbone in the USA. On the right a grid graph (or lattice) with 12 vertices is shown. Furthermore, a complete graph (or fully connected graph) with 6 vertices is used for the simulative evaluation of our algorithms. It is to mention that the multi-path strategy has advantages in a richer connected topology as different paths can be used for the data transfer at the same time. In order to examine this property, topologies with an increasing nodal degree are used (2–3 for Abilene, 2–4 for the grid graph, and 6 for the complete graph).

IV. Algorithms for the Single-Path Strategy

In this section we concentrate on a single-path strategy, i.e. only one request is processed at a time and only one path is used in the network at a time to convey data. Therefore, the first set of algorithms maps a request r to a single path. After presenting the algorithm, an evaluation with different path selections is presented.

A. A Heuristic for Constant Capacity Single-Path All-Slots

This approach is derived from an algorithm presented in [5] and tries to map a file transfer request to an advance reservation given by a single path and a constant capacity. It is a heuristic and avoids the super-polynomial runtime of the variable capacity approach (cf. advance cumulative reservations [4]). Although Burchard describes a similar algorithm in [5] that maps requests according to the mentioned heuristic, details of the algorithm are different (e.g. we are using timeslots of dynamic length). Therefore our algorithm is described in detail. The overall approach is to start the search for configurations in the temporal domain. The algorithm maps a request to a contiguous capacity block in the network resulting in a single reservation entity. It includes the following phases:

1) Determine potential time intervals for reservation entities (potential configurations),
2) compute a path for each potential configuration, and
3) select a potential configuration meeting the demand or reject the request.

In the first phase potential configurations are determined by timeslots of already accepted reservations, the release time t_r and the deadline t_d of the processed request. Futhermore, the maximum capacity given by the widest path between the source s and destination d according to c_{max} is considered. Given k reservations, this leads to $\sum_{i=1}^{2k+2} i \in O(k^2)$ configurations at most. Taking a look at figure 4, the potential configurations starting at time $t_r = t_1$ are presented at the right side. The corresponding timeslots are taken from a set of example reservations presented at the left side of the figure. Additionally, two default configurations are added based on the maximum capacity with respect to the widest path (cf. $[t_1, t_{min}[$ in figure 4) and the minimum capacity regarding the release time and deadline (cf. $[t_1, t_d[$).

In the second phase, these configurations are analyzed by a path selection algorithm on edges that meet the given capacity constraint on all timeslots from the start to the end. As the

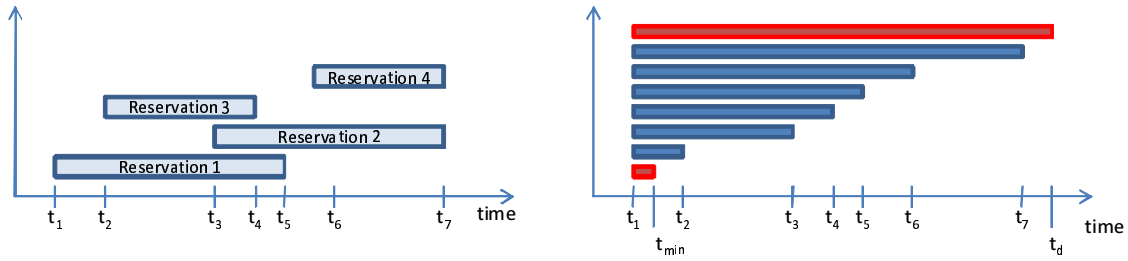

Fig. 4. Time Line with respect to accepted Reservations (left) and Potential Configurations starting from t_1 (right)

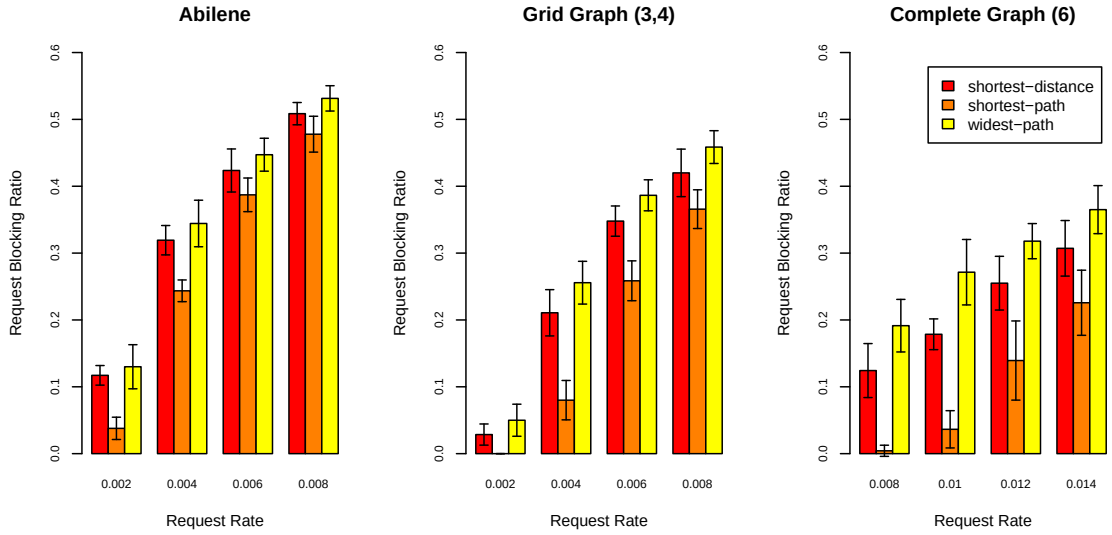

Fig. 5. Simulation Results for the Single-Path all Slots Algorithm

capacity constraint is a link constraint in the sense of QoS routing a preprocessing time of $O(k|E|)$ is needed, where $|E|$ is the number of edges in the graph and k the number of accepted requests. Subsequently, a path can be computed on the preprocessed graph. Here, a basic Dijkstra algorithm with a running time of $O(|V|^2)$ is used to find a path.

In the third phase, three tie-breaking rules are used to select a configuration. The following rules are processed with decreasing priority: (i) prefer the configuration according to the path selection metric (e.g. the configuration with the overall shortest path is selected), (ii) prefer the longest duration, and (iii) select a configuration at random. In other words, the presented algorithm processes a single request by aligning it to the already accepted requests while using network resources determined by a path selection strategy.

B. Experiments on the Single-Path all Slots Strategy

The second phase of the previously described heuristic can utilize a variety of path selection algorithms. These algorithm determine the network resources for the potential configurations. In order to identify a favorable path selection algorithm, we take a look on the performance of the following algorithms:

- Select the shortest path (measured in hop count).
- Select the widest path with respect to the residual capacities.
- Select the shortest distance given by the sum of the reciprocal of the residual capacities.

It is to mention that the corresponding implementations use a randomized graph traversal. This leads to varying paths of equal cost.

Figure 5 shows the RBR (with 0.95 confidence interval) as a function of the request rate. The reciprocal value of the request rate specifies the mean inter-arrival time. The initial link capacity is identical on all link of the Abilene topology, the grid graph, and the complete graph. The generated requests are uniformly distributed between all source-destination pairs and a single value for the data amount is used with a duration of approximately 4 times the minimal transmission time. The minimal transmission time is given by the widest path between the source and the destination. As a result, 4 requests share the capacity of a link in a timeslot at most. Overall, the RBR is decreasing from the Abilene topology to the complete graph as the capacity of the graphs ($\sum_{e \in E} c_{max}(e)$) is increasing and the structures provide more paths between arbitrary pairs of vertices.

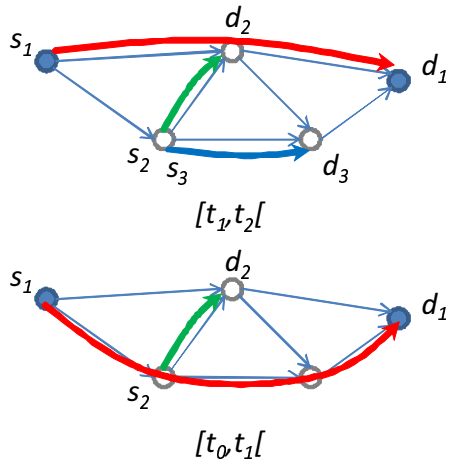

Fig. 6. Switching a single path between timeslots

In all aggregated simulation runs presented in figure 5 the shortest path algorithm performs best. The widest path and shortest distance algorithm tend to allocate longer paths and therefore use more resources. The usage of longer paths blocks links needed for following requests. Increasing the network load results in smaller performance differences between the path selection algorithms. The shortest distance algorithms performs slightly better than the widest path algorithm in our scenarios.

As the usage of *shortest path* is the most suitable in the presented simulation scenarios, this path selection algorithm is preferred for comparing the single-path algorithm against the multi-path algorithm.

C. Further Single-Path Strategies

An additional single-path strategy is a topic for further studies. This strategy allows for reservation entities changing the path per timeslot. One approach for a single-path per slot strategy is described in [5]. The advantage of a single-path per slot strategy is depicted in figure 6. In the subsequent timeslot denoted by $[t_0, t_1[$ two commodities are present. The next timeslot $[t_1, t_2[$ has three commodities. If the path for the source-destination pair (s_1, d_1) can be switched between the timeslots – for example by using path switching capabilities of MPLS nodes – additional alternatives can be regarded in order to lower the RBR.

V. Algorithms for the Multi-Path Strategy

In this section the processing of a file transfer request with a multi-path strategy is presented. The algorithm maps a request to a network flow and chooses paths form this flow. This section concludes with an evaluation of different path selection strategies.

A. Maximum Flow Approach

The main idea of the multi-path approach is to construct a time expanded graph representing the scheduled load in the network. This graph covers the timeslots of the new request. The possible reservation entities are limited by the maximum flow from the requested source to the destination in this graph.

Starting from this idea, the admission decision of a multi-path request is split into three phases:

1) Create a time expanded graph representing the available resources during the requested time period,
2) compute the maximum flow in the time expanded graph, and
3) select a set of paths from the network flow meeting the demand or reject the request.

In the first phase, a time expanded graph is constructed as basis for a flow computation. Briefly, the time expanded graph contains a copy of the basic graph for each timeslot considered. The edges of each copy are annotated with the residual capacity of the corresponding timeslot. Note that each copy represents the network connectivity in terms of residual capacities in a specific timeslot with a certain length. A path traversing one of the copies can be interpreted as certain amount of data that can be transferred in this time interval. An example of a time expanded graph as defined in this paper for a request $r = (s, d, t_a, t_r, t_d, v)$ is shown in figure 7. Let $[t_0, t_1[$, $[t_1, t_2[$, and $[t_2, t_3[$ with $t_0 = t_r$ and $t_3 = t_d$ be the timeslots which are candidates for data transmission of r. A "virtual" source s' and a "virtual" destination d' are defined which are connected to the sources and destinations of the different timeslots s_i,d_i respectively. The capacity of the edges from the two virtual nodes to each copy has to be selected in accordance to the duration of the timeslot to configure the maximum amount af data that should be transmitted in the timeslot. The size of the time expanded graph depends on the number of timeslots covered by the request, i.e. up to $2k + 1$ copies of the graph can be present. In the following we will refer to the indexed set of copies as layers.

In the second phase the maximum flow from the virtual source s' to the virtual destination d' is computed. This can be done by any maximum flow algorithm [16]. The Edmonds-Karp algorithm (which is based on the Ford-Fulkerson method) has been used in the context of this paper. If the flow value – which relates to a data amount – is larger or equal to the requested amount of data, a potential solution to the reservation request has been found. If the flow value is less than the data amount to be transferred, the request is rejected. In the case of equality a decomposition of the flow into paths is a valid solution. If the maximum flow is larger, a sub-flow with a flow equal to the requested data amount has to be chosen.

In the third phase, the flow is decomposed into a set of paths. In order to decompose the flow, different approaches can be used. In our approach, paths are always selected from the maximum flow and the flow graph is updated by subtracting the data amount from the corresponding edges. We use the Dijkstra algorithm to perform the decomposition into a set of paths. If a sub-flow satisfies the request, heuristics are used to choose a subset of all available paths from the maximum flow. This subset will be used to specify the reservation entities (cf. section III-D). Note that paths and flows in certain layers of the

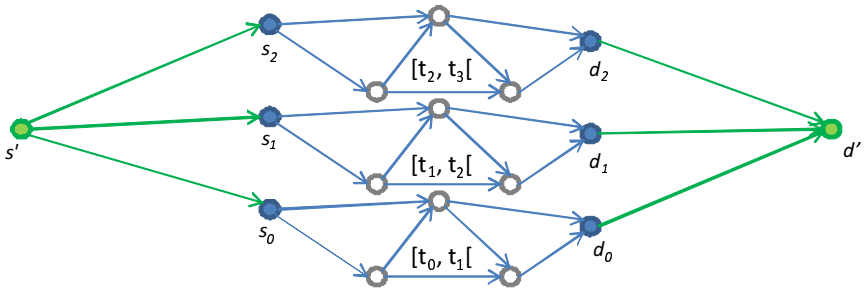

Fig. 7. Time-Expanded Graph to solve the Maximum Flow Problem

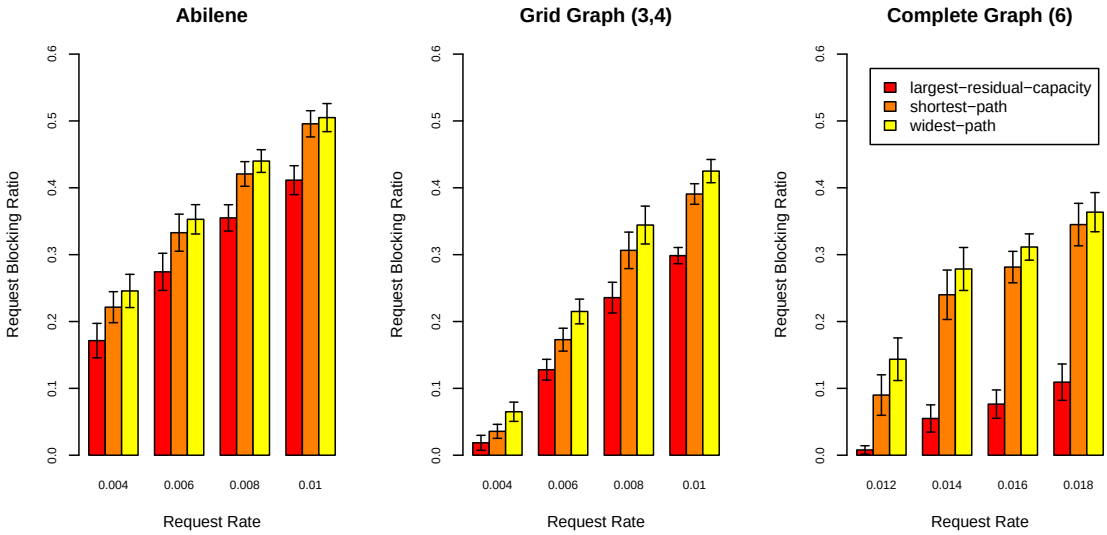

Fig. 8. Simulation Results for the Multi-Path per Slot Algorithm

time-expanded graph are associated with the duration of the associated timeslot. These paths and flows can be interpreted as a certain amount of data.

B. Experiments on the Multi-Path per Slots Strategy

The maximum flow value computed in the second phase of the presented algorithm can identify a larger data amount than requested. In this section two families of heuristics are presented to choose a subset of the decomposed maximum flow. One family follows the shortest path approach, i.e. the heuristic always chooses the shortest path (measured in hop count) from the set of available paths, until the set of chosen paths meets the requested data amount. The other family follows the widest path approach, i.e. repeatedly choose the path which corresponds to the largest data amount transferable. This is done with the intention to minimize the total number of paths and therefore minimize the resource configuration effort.

1) Shortest Path: This approach always chooses the shortest path in the set of the residual available paths. If two or more paths have equal length, the one found first is taken.

2) Largest Residual Among Shortest: Again, this approach is based on the shortest path strategy. If two or more paths with equal length are present, this strategy considers the capacity instead of the data amount. Remember that in the context of the time expanded graph the flow refers to the amount of data that can be transferred. This strategy chooses a path, which has the largest residual capacity on its bottleneck link in the given timeslot.

3) Widest Path: This approach always chooses the path corresponding to the largest data amount. This leads to a low number of paths to meet the data amount specified by the request.

Figure 8 shows the results of the simulative comparison of the different path selection algorithms. Again, the RBR (with 0.95 confidence interval) is given as a function of the request rate. The simulation parameters are used in accordance to the previous sections. Although the duration interval is approximately 4 times the minimal transmission time w.r.t widest path between the source and the destination, more than 4 requests can share the capacity of a link in a timeslot. This is due to the fact that the demand can be meet by multiple paths. Again, the RBR is decreasing from the Abilene topology to the complete graph because the capacity of the graphs is increasing.

In all three scenarios the strategies based on the shortest path approach generally perform better than the widest path

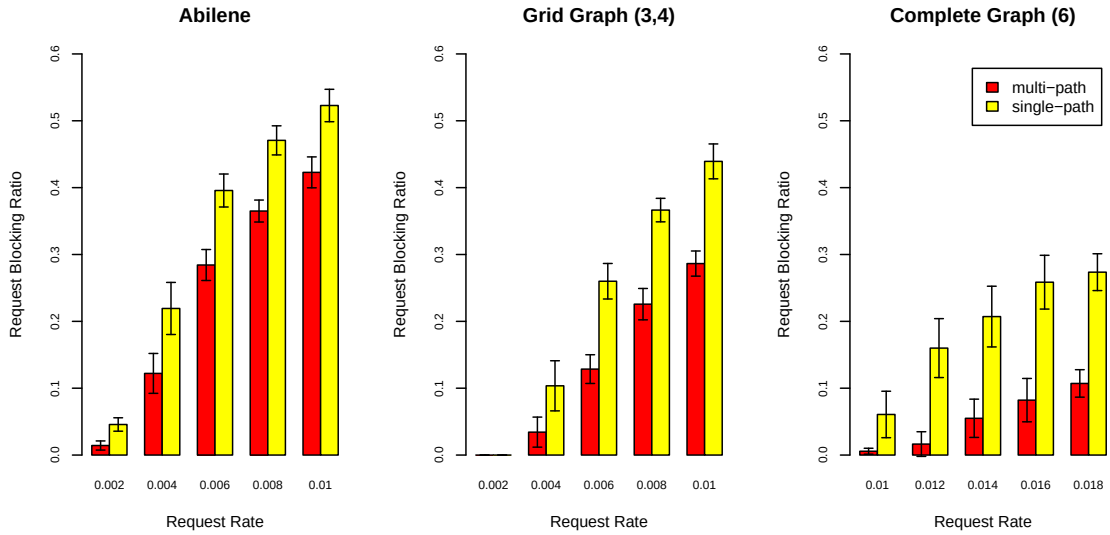

Fig. 9. Comparison of the Single-Path all Slots and Multi-Path per Slot Algorithms

approach. These simulation results are in accordance with the findings of [15]: QoS routing strategies using paths with fewer hops generally perform better than those neglecting the path length. In all aggregated simulation runs presented in figure 8 the *Largest Residual Among Shortest* algorithm performs best. The algorithm results in good performance, because the network load is distributed in a way that avoids the blocking of links. This increases the chance of successful admission of subsequent requests. The widest path strategy suffers from its greedy behavior. It takes the widest available paths, which results in long paths, and therefore in unnecessary resource usage which lowers the chance of successful admission of further requests.

Due to the results of these simulations, the *Largest Residual Among Shortest* heuristic is chosen as the default heuristic and is used for the comparison in section VI.

C. Further Variations on the Multi-Path Strategy

Additional strategies for multi-path reservations are under consideration. A variant is to avoid gaps in reservations by using at least one path per timeslot. This allows for transmitting data continuously and making multiple paths transmissions more apt for protocols which cannot handle transmission interruptions. An alternative approach is to use the same or at least a limited number of paths per timeslot. This might be important if the client handles multi-path transfers by using a dedicated TCP or UDP connection per path.

VI. Comparison of the Single- and Multi-Path Algorithms

Figure 9 shows simulation results for the single- and multi-path algorithm. Again, the RBR (with 0.95 confidence interval) is given as a function of the request rate. The set of request rates are chosen with respect to the simulations results presented in section IV and V. Again, simulation parameters are used in accordance to the previous sections.

Overall, the RBR of the multi-path algorithm is lower compared to the single-path algorithm. It is to mention that the benefit of the multi-path algorithm in figure 9 increases regarding the different topologies from Abilene to the complete graph. This is due to the fact that more alternative paths exist between arbitrary source-destination pairs. The minimum link capacity that can be used by the single-path algorithm is determined by the maximum duration and the specified data amount. Links providing a lower residual capacity can only be used by the multi-path strategy by splitting the demand onto several paths. Therefore, the most significant benefit of the multi-path approach is achieved in the complete graph scenarios.

Although the multi-path approach performs best, a drawback of the multi-paths approach is obvious: The network, transport, and application layer have to cooperate to facilitate a transfer on multiple paths. In a circuit-switched network flows can be partitioned at the ingress and mapped onto different paths with corresponding capacities, e.g. by protocol header information. Beside the striping approach used by GridFTP (cf. section II), the transport layer protocol SCTP in conjunction with its multi-streaming and multi-homing feature could be used to distribute data across multiple end-to-end paths [12].

VII. Conclusion and Further Work

In this paper, a flexible reservation mechanism for network resources for demanding, distributed Grid applications was introduced with a special focus on a file transfer service. This file transfer service guarantees the timely availability of data at the different Grid sites even for very large files. To cope with

these requirements, an architecture for advance reservations of network resources and different strategies and algorithms for a flexible network reservation service were defined. These algorithms can be classified in two fundamentally different strategies. The first one uses a single path per file transfer request while the second class of strategies splits the data transfer for a request onto several distinct paths at the same time. For the single path strategy we described a heuristic realizing the most rigid approach, i.e. using a constant capacity during the whole transmission without interruption. The multi-path approach is based on the decomposition of the maximum flow between the source and the destination into a set of paths and in general uses different data rates during the transmission. Our simulation results show that data transport for a single file via multiple paths can lead to a higher user and network provider satisfaction than single path transfers. The results were strongly dependent on the structure of the underlying network topology and suggest that with a higher nodal degree the advantage of the multi-path strategy becomes larger. The most significant gain was achieved in a fully connected topology.

Further work on the filer transfer service is planned in a practical and theoretical manner. On the one hand, the described algorithms are integrated in the network reservation system ARGON. While clients using the constant capacity single-path approach to transfer files can easily be integrated in an overall architecture, the integration of file transfer applications for the multi-path approach is under consideration. On the other hand, algorithms only sketched in section IV and V are going to be designed and evaluated. Including these variants into a set of potential algorithms for the file transfer service and answering questions on performance and evaluating the potential usability is the topic of ongoing work.

REFERENCES

[1] D. Clery, "Can grid computing help us work together?" *Science*, vol. 313, no. 5786, pp. 433–434, 2006.

[2] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy, "A distributed resource management architecture that supports advance reservations and co-allocation," in *Proceedings of the International Workshop on Quality of Service*, 1999. [Online]. Available: citeseer.ist.psu.edu/foster99distributed.html

[3] B. Bierbaum, C. Clauss, T. Eickermann, L. Kirtchakova, A. Krechel, S. Springstubbe, O. Wäldrich, and W. Ziegler, "Reliable orchestration of distributed mpi-applications in a unicore-based grid with metampich and metascheduling," Institute on Resource Mangement and Scheduling, CoreGRID - Network of Excellence, Tech. Rep. TR-0052, August 2006. [Online]. Available: http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0052.pdf

[4] R. Guérin and A. Orda, "Networks with advance reservations: The routing perspective." in *INFOCOM*, 2000, pp. 118–127.

[5] L.-O. Burchard, "Advance reservations of bandwidth in computer networks," Ph.D. dissertation, Technical University of Berlin, 2004. [Online]. Available: edocs.tu-berlin.de/diss/2004/burchard_lars.pdf

[6] "VIOLA: Vertically Integrated Optical Testbed for Large Applications in DFN." [Online]. Available: http://www.viola-testbed.de/

[7] C. Barz, F. Hommes, W. Moll, M. Pilz, C. Rosche, and J. Schon, "Argon - allocation and reservation in grid-enabled optical networks," BMBF-VIOLA Project, Tech. Rep. B2.4.1, August 2005. [Online]. Available: http://www.viola-testbed.de/content/fileadmin/VIOLA/reports/Report_B2.4.1.pdf

[8] "IST Phosphorus." [Online]. Available: http://www.ist-phosphorus.eu/

[9] B. Dobinson, R. Hatem, W. Hong, P. Golonka, C. Meirosu, E. Radius, and B. S. Arnaud, "Transatlantic native 10 gigabit ethernet experiments: Connecting geneva to ottawa." in *HSNMC*, 2004, pp. 108–119.

[10] Q. He, C. Dovrolis, and M. Ammar, "On the predictability of large transfer tcp throughput," in *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2005, pp. 145–156.

[11] W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link, "The globus striped gridftp framework and server," in *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. Washington, DC, USA: IEEE Computer Society, 2005, p. 54.

[12] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent end-to-end paths," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, 2006.

[13] L.-O. Burchard, "Analysis of data structures for admission control of advance reservation requests," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 413–424, 2005.

[14] A. Roy and V. Sander, "Gara: a uniform quality of service architecture," pp. 377–394, 2003.

[15] Q. Ma and P. Steenkiste, "Quality-of-service routing for traffic with performance guarantees," in *IFIP Fifth International Workshopon Quality of Service, NY, NY*, 1997, pp. 115–126. [Online]. Available: citeseer.ist.psu.edu/ma97qualityservice.html

[16] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1993.