Flexible Scheduling and Control of Bandwidth and In-transit Services for End-to-End Application Workflows

Mehmet Fatih Aktas, Georgiana Haldeman and Manish Parashar
Rutgers Discovery Informatics Institute (RDI^2)
Rutgers University, Piscataway NJ, USA Email: {mehmet.aktas, haldeman, parashar}@rutgers.edu

***Abstract*—Emerging end-to-end scientific applications that integrate high-end experiments and instruments with large scale simulations and end-user displays, require complex couplings and data sharing between distributed components involving large data volumes and varying hard (in-time data delivery) and soft (in-transit processing) quality of service (QoS) requirements. As a result, efficient data transport is a key requirement of such workflows. In this paper, we leverage software-defined networking (SDN) to address issues of data transport service control and resource provisioning to meet varying QoS requirements from multiple coupled workflows sharing the same service medium. Specifically, we present a flexible control and a disciplined resource scheduling approach for data transport services of science networks. Furthermore, we emulate an SDN testbed on top of the FutureGrid virtualized testbed and use it to evaluate our approach for a realistic scientific workflow. Our results show that SDN-based control and resource scheduling based on simple intuitive models can meet the coupling requirement with high resource utilization.**

I. Introduction

As scientific discovery is becoming increasingly data driven, scientific applications are moving towards end-to-end workflows that integrate coupled simulations with data sources such as instruments, sensor system and experiments, and with data analysis and visualization pipelines to facilitate online knowledge extraction. Furthermore, the execution of such workflows often involves geographically distributed resources with runtime interactions, coordination and data exchanges between processes running on these resources [1], [2].

Scientific workflows typically involve complex couplings between the workflow components, requiring sharing of large data volumes with varying hard (in-time data delivery) and soft (in-transit processing) quality of service (QoS) requirements, and efficient data transport is a key requirement. Specifically, the time between when the data is generated at producer and when it can be consumed at the consumer can have a significant impact on the execution of the workflow. For example, slower data delivery can throttle the consumer, or faster data delivery may require storing the data at the consumer. Some applications may require that data is delivered at the consumer within a tight time window (for example, data needed to control an experiment), which adds further requirements on the data transport. Additionally, there often exists natural mismatches in the way data is represented at producers and consumers, and the data has to be transformed in a timely manner before it can be consumed. As a result, the data transport has to address multiple challenging requirements based on data sizes, data production and consumption rates, strict constraints on data delivery time and data storage, and managing data transformations between producers and consumers. The data transport medium is also typically shared by multiple application workflows with possibly competing application specific coupling requirements, and best-effort solutions, which inherently can not offer service guarantees, will not be able to achieve high performance in an over utilized service medium.

The goal of this project is to address these challenges, to research flexible scheduling and to explore the control of bandwidth and in-transit services for end-to-end scientific application workflows. Specifically, in this paper, we leverage software-defined networking (SDN) to address issues of data transport service control and resource provisioning to meet varying QoS requirements from multiple coupled workflows sharing the same service medium. The specific contribution of this paper is to present a disciplined resource scheduling approach for data transport resources, which is both application and network aware, and it offers flexible control. We also develop a model for in-transit data staging and data processing using intermediate resources in the data path using the approach outlined in [2]–[4]. Finally, we emulate an SDN testbed on top of the FutureGrid virtualized testbed and use it to evaluate our approach for an end-to-end Fusion workflow. Our results show that SDN-based control and resource scheduling based on simple intuitive models can meet the coupling requirement with high resource utilization.

II. Problem Description – Enabling End-to-End Coupled Simulation Workflows

Emerging scientific applications integrate simulations with data sources such as experiments and instruments, and analysis and visualization pipelines, into end-to-end workflows. These workflows exhibit different and varying interaction, coordination and data coupling behaviors, such as: in *tight coupling* case coupled processes exchange data very frequently, in *loose coupling* case data exchange is relatively less frequent, asynchronous and possibly with some slack time for opportunistic intermediate on-the-fly data processing, and finally in *dataflow coupling* case data flows from producer to consumer using publish/subscribe/notify-like semantics, for example, in case of data processing and/or analysis pipelines, and often strict data transport constraints are involved such as delivery time, data integrity and reliability.

To meet such complex and varying coupling requirements, it is essential for the data transport service to be application aware and self-optimizing in that it must autonomously adapt based on dynamics application requirements and resource states. Furthermore, the overheads of adaptation and service management on the application should be acceptable.

In this paper, we focus on control and scheduling of data transfers (across network switches and links) and in-transit service (intermediate staging and processing hosts) resources to meet varying data transport requirements of differing coupling behaviors, such as those described above. In this work, we assume that the different coupling requirements of the application workflow are provided to the scheduler before the coupled data streamed, as further discussed in Section III-B.

SDN is an initiative to make network control simpler and more flexible using well-defined abstractions for forwarding and switch configuration. The main idea behind SDN is to replace the existing distributed control with centralized control. This is achieved by implementing network control programs on top of the network operating system (NOS), which in turn provides the control programs with a global view of the network and interfaces for communicating with switches, enabling the control programs to configure the network state.

In this research we explore how the data transport service can benefit from SDN and its characteristics such as (1) open and programmable control, (2) faster innovation in networking layer, (3) easy customization and optimization of network resource scheduling due to flexible control, can be used to enhance data transport service management. Overall, by leveraging SDN-based networking control, we propose to use a centralized and flexible framework for data transport service control and scheduling for end-to-end coupled application workflow.

III. Flexible Scheduling and Control of Bandwidth and In-transit Services

In this section, we present the three key components of this research, i.e., (1) A flow-based in-transit service model, (2) Centralized layered architecture for data transport service control, and (3) Application and network aware resource scheduling for workflow couplings.

A. *In-transit Service Model*

We observed that in the packet-based intermediate processing/staging model, the end-to-end system design arguments are violated by the interception of network packets on-the-fly (as further discussed in [5], [6] and [7]), which significantly impacts the existing network stack performance. We propose an in-transit service model to achieve intermediate data processing over the application data in a flow-based manner. For that purpose, switch-host coordination can be used, in a similar manner to gateway-host coordination in Phoebus architecture [8], to break the end-to-end connections into subconnections over intermediate hosts where the application data is treated as a flow. The autonomic switch-host coordination for creating and managing sub-connections can be handled by the granular forwarding control of SDN, which is explained in Subsection III-B. Flow-based in-transit processing/staging service can be implemented between transport layer (TCP) and the application layer using an in-transit service protocol (IT-SP). The IT-SP header enables the flow of service specific information (e.g., the list of functions to opportunistically execute over the data, how long data is staged) between the producer, the consumer and intermediate hosts.

B. *Data Transport Service Control*

In order to manage the autonomic switch-host coordination and ensure efficient in-transit service control for flow-based intermediate processing and staging, our approach is to extend SDN-NOS to a Data Transport OS (DT-OS) that offers support for in-transit service management with the help of a controller (scheduler) running above it. The scheduler contains the control logic that is required to decide on the appropriate actions necessary to optimize the allocation of resources. Similar to NOS, DT-OS provides a centralized view of the service medium to the scheduler, and in turn, the scheduler tells DT-OS what resources to allocate to individual coupling sessions according to their tolerances for hard requirements and demands for soft requirements. The resulting control architecture consisting of the DT-OS and the scheduler preserves the centralized and layered features of the SDN architecture. Overall, our framework adds two additional abstractions to the existing SDN abstractions: (1) Switch-host coordination for flow-based intermediate data processing/staging and (2) Configuration of in-transit processing/staging pipelines at intermediate hosts.

Control operations of DT-OS for realizing a single coupling session is briefly as follows. Application pushes a request for a coupling session including application specific requirements and the request is forwarded to scheduler by DT-OS. After a feasibility check on application requirements, scheduler generates rules for data walk route and optimized resource allocation for the session. DT-OS realizes scheduling rules by setting up forwarding and in-transit service tables at switches and intermediate hosts. Finally, a reply containing scheduling information is sent to application by DT-OS. Once application receives the reply, it immediately starts streaming data and data is served autonomically over the scheduled resources. The scheduler treats every application request as a new coupling session. When there are multiple active coupling sessions, the scheduler may need to reallocate resources between them, dynamically update the resource states, and inform the applications about any changes.

C. *Resource Allocation*

We have addressed the resource allocation by systematically formulating it as a convex optimization problem, since convex optimization problems have unique globally optimal solutions that can be obtained with several existing efficient, reliable and robust algorithms [9].

Our scheduling scheme assumes that application requirements are autonomically received from the applications and intermediate processing routines do not change the data size. We use cumulative user satisfaction as the scheduling objective, which in this case means that scheduler tries to maximize satisfaction of all coupling sessions' requirements by allocating resources accordingly. Applications make a request for a coupling session by informing scheduler about the optimal transport time and tolerance for in-time delivery, and in-transit processing routines, and then demand for on-the-fly completion of those routines. Given the coupling session requirements and resource availability, the scheduler schedules a data walk on a network path with allocated bandwidth, in-transit processing rate and staging duration, and determines how much of the given in-transit processing routines are to be completed on-the-fly. For every request the scheduler receives or every time a session ends, resources are dynamically reallocated to achieve highest cumulative user satisfaction.

IV. Experiments and Results

To evaluate the framework described in this paper, we used the ”Plasma Disruption Analysis” workflow [10] from the KSTAR[1] project to develop synthetic use case scenarios. In this application, plasma disruptions occur due to loss of stability and/or confinement of tokamak plasmas and cause a fast thermal and/or current quench within sub-milliseconds, which can damage the expensive (multi-billion dollars) tokamak device. As a result, finding precursors and early prediction of tokamak operation anomalies is a very active research field. One such research effort is the plasma visualization diagnostics system designed to provide direct 2D/3D visualizations of the plasma in a tokamak. Visual plasma images are obtained via monitoring technologies such as Soft X-Ray (SXR), Microwave Imaging Reflectometry (MIR), or Electron Cyclotron Emission Imaging (ECEI). Diagnostic routines are then run over the plasma images to detect precursors of plasma disruption.

For our experiments, we used data, diagnostic and visualization routines for ECEI to compose synthetic scenarios consisting of multiple coupling sessions with varying requirements sharing a service medium. ECEI generates high resolution 2D images of radiated electron temperature, which provide visualization of plasma instabilities – specifically, a 24x8 float matrix image is generated every 2μs, i.e., 5,000,000 images are generated in 10 seconds resulting in $\approx$ 3.5GB of data. The over aching goal of KSTAR is to enable a remote scientist, for example, in the US, to monitor the plasma visualizations to monitor tokamak stability and to take regulatory actions if necessary. This requires diagnostic and visualization routines to be run either at US site or opportunistically over the available intermediate resources.

For our experimentation testbed, we implemented DT-OS by extending POX, which is a networking software platform [11], and the scheduler by using POX API. Further, we used CVXPY [12] as the modeling language for the optimization problem for resource allocation. Finally, we implemented the IT-SP layer on top of TCP at the producer, consumer and intermediate hosts.

To emulate a coupling session with producer-consumer pairs, network switches-links, and in-transit hosts in our experiments we used mininet [13]. Running the entire system on a single machine is convenient but imposes limitation on switching and in-transit processing capacity. For example, if the machine has 3GHz of CPU, mininet can switch at most 3 Gbps simulated traffic. Moreover, overall available emulation capacity is shared by multiple coupling sessions and resources. Therefore, we used multiple mininets running on different VMs interconnected via GRE tunnels on top of Infiniband. We then deployed our testbed on a distributed OpenStack cloud as part of the FutureGrid testbed [2].

Initial results from our experiments using the virtualized emulation testbed described above and the synthetically generated use case scenarios validate the following:

- Overheads introduced by the scheduling process are minimal. The round trip time between application request and scheduler response, and the time to solve the optimization problem are significantly smaller than the duration of the coupling session, and for large data cases this overhead is almost negligible. For example, the scheduling overhead is less than 1% of the overall data transport time.
- The disciplined resource allocation optimization formulated in this paper enables the scheduler to satisfy application requirements efficiently. If the application does not have any tolerance in the on-time data delivery requirement, then the gap between optimal and actual data transport times is managed to be small. If the application demands in-transit processing, then in-transit processing routines are aggressively scheduled and completed on-the-fly.
- The scheduler achieves high resource utilization of network and in-transit resources, while also meeting application requirements.

V. Related Work

In [2], Bhat et al. use adaptive buffer management based on proactive and/or online user-defined policies for the QoS management of a self-managing data streaming and in-transit processing service for Grid-based data intensive workflows. In this work, the streaming and in-transit processing components work cooperatively (using feedforward and feedback messages) to meet overall application requirements and constraints. This solution is similar to our solution in that it uses QoS management to meet user-defined requirements. However the approach in this work does not address the management of data transport resources, while our research is founded on scheduling and controlling of the transport resources. Cooperative management strategies presented in [2] could be used to extend our framework to work efficiently on a commodity service medium.

Active networking is a communication pattern [14] for tailoring network service to user requirements as explained in [15]. A network is called active when processing can be done within the network over active elements such as switches that have processing capability. Programming active switches according to application specific needs and taking advantage of packet-based processing within the network has been a key focus of active networking research. In [16], Lefevre et al. present an active network architecture (A-Grid) that attempts to provide QoS management for Grid data transport services in addition to other data transport services such as reliable multicast and dynamic service deployment. Their architecture employs QoS management at intermediate active routers, and

[1] Korea Superconducting Tokamak Advanced Research

[2] FutureGrid is part of NSF’s high-performance cyberinfrastructure – see futuregrid.org

in principal, it is similar to the opportunistic in-transit processing employed by our solution. However, our solution makes use of intermediate hosts to do flow-based processing rather than packet-based processing over active network elements. Moreover, their solution can only provide QoS management per application, but not per workflow, because their architecture lacks a global view of the service medium and of simultaneously running user applications, which is another feature of our solution.

Network resource reservation systems such as ESNET's OSCARS [17] and UltraScience Net [18] provide on-demand dedicated bandwidth channels to user applications. They introduce a virtual single-switch abstraction on top of networks which employ both a bandwidth reservation system and SDN concepts in [19]. The work presented in this paper is different but complementary. In this paper, we presented a disciplined scheduling of bandwidth and in-transit processing/staging capacity to meet application requirements for data transport in the context of data-intensive coupled workflows, and utilized SDN-based centralized control concepts to manage resource states according to scheduling decisions. Network reservations and virtualization systems can be used in a complementary manner to improve performance of the scheduling and flexibility of the resource control that we introduced.

VI. Conclusion and Future Work

This paper presented the architecture and design of a flexible control/scheduling framework for data transport service management of data intensive scientific workflows. Our frameworks leverages software-defined networking (SDN) to address issues of data transport service control and resource provisioning to meet varying QoS requirements from multiple coupled workflows sharing the same service medium. Furthermore, we addressed both, data transfer and in-transit processing/staging services. The presented framework has three key components: (1) A flow-based opportunistic in-transit processing/staging service model, (2) SDN-based centralized architecture for data transport service control, and (3) Application and service medium aware scheduling. Overall, the framework attempts to address complex, dynamic and varying data transport requirements of coupled application workflows by scheduling bandwidth and in-transit processing/staging capacity. Scheduling is formulated by using intuitive models for data transfer and in-transit processing/staging, and implemented as a disciplined convex optimization problem.

The paper also presented an experimental evaluation using an emulated SDN testbed on top of the FutureGrid virtualized testbed, and we used synthetic application workflow scenarios (derived from real Fusion simulations workflows) to demonstrate that our framework achieved high cumulative user requirement satisfaction and high resource utilization, and can meet the coupling requirements.

Our current work is focused on deploying the framework in a HPC Grid environment and using it to support real application workflows. We are also working on getting the optimization problem to work for more complex situations, such as with in-transit routines changing data size.

References

[1] L. Zhang, C. Docan, and M. Parashar, "The seine data coupling framework for parallel scientific applications," *Advanced Computational Infrastructures for Parallel and Distributed Adaptive Applications*, p. 283, 2010.

[2] V. Bhat, M. Parashar, and S. Klasky, "Experiments with in-transit processing for data intensive grid workflows," in *Proceedings of the 8th IEEE/ACM International Conference on Grid Computing*. IEEE Computer Society, 2007, pp. 193–200.

[3] V. Bhat, M. Parashar, H. Liu, M. Khandekar, N. Kandasamy, and S. Abdelwahed, "Enabling self-managing applications using model-based online control strategies," in *Autonomic Computing, 2006. ICAC'06. IEEE International Conference on*. IEEE, 2006, pp. 15–24.

[4] V. Bhat, S. Klasky, S. Atchley, M. Beck, D. McCune, and M. Parashar, "High performance threaded data streaming for large scale simulations," in *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*. IEEE, 2004, pp. 243–250.

[5] J. S. Plank and M. Beck, "The logistical computing stack–a design for wide-area, scalable, uninterruptible computing," in *Dependable Systems and Networks, Workshop on Scalable, Uninterruptible Computing (DNS 2002)*. Citeseer, 2002.

[6] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research," *Communications Magazine, IEEE*, vol. 35, no. 1, pp. 80–86, 1997.

[7] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura, "Active networking and the end-to-end argument," in *Network Protocols, 1997. Proceedings., 1997 International Conference on*. IEEE, 1997, pp. 220–228.

[8] A. Brown, E. Kissel, M. Swany, and G. Almes, "Phoebus: A session protocol for dynamic and heterogeneous networks," *University of Delaware, Tech. Rep*, vol. 2008, p. 334, 2008.

[9] J. Mattingley and S. Boyd, "Automatic code generation for real-time convex optimization," *Convex optimization in signal processing and communications*, pp. 1–41, 2009.

[10] J. Lee, J. Kim, C. Kessel, and F. Poli, "Simulation study of disruption characteristics in kstar," in *APS Meeting Abstracts*, vol. 1, 2012, p. 8047P.

[11] "Pox," http://www.noxrepo.org/pox/about-pox/.

[12] S. Diamond, E. Chu, and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization, version 0.2," 2014.

[13] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010, p. 19.

[14] D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," in *DARPA Active NEtworks Conference and Exposition, 2002. Proceedings*. IEEE, 2002, pp. 2–15.

[15] T. M. Chen and A. W. Jackson, "Commentaries on" active networking and end-to-end arguments"," *Network, IEEE*, vol. 12, no. 3, pp. 66–71, 1998.

[16] L. Lefèvre, C.-d. Pham, P. Primet, B. Tourancheau, B. Gaidioz, J.-P. Gelas, and M. Maimour, "Active networking support for the grid," in *Active Networks*. Springer, 2001, pp. 16–33.

[17] C. Guok, D. Robertson, M. Thompson, J. Lee, B. Tierney, and W. Johnston, "Intra and interdomain circuit provisioning using the oscars reservation system," in *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*. IEEE, 2006, pp. 1–8.

[18] N. S. Rao, W. R. Wing, S. M. Carter, and Q. Wu, "Ultrascience net: Network testbed for large-scale science applications," *Communications Magazine, IEEE*, vol. 43, no. 11, pp. S12–S17, 2005.

[19] I. Monga, E. Pouyoul, and C. Guok, "Software defined networking for big-data science," *SuperComputing 2012*, 2012.