CoUniverse: Framework for Building Self-organizing Collaborative Environments Using Extreme-Bandwidth Media Applications

Miloš Liška and Petr Holub

Laboratory of Advanced Networking Technologies
Faculty of Informatics
Masaryk University
Botanická 68a, 621 00, Brno
Czech Republic
xliska@fi.muni.cz,
hopet@ics.muni.cz

Abstract. In this paper, we present a framework called CoUniverse, designed for building real-time user-empowered collaborative environments to work primarily on high-speed networks with true high-bandwidth applications such as uncompressed high-definition video. The system is designed for unreliable experimental infrastructures and therefore its operation relies heavily on self-organizing principles—this is also useful approach for extending it to larger infrastructures. When media stream bitrate is comparable to a capacity of the links, the additive assumption no longer holds and the system needs to have a sophisticated scheduling. The scheduler is conceived as a flexible plug-in for the CoUniverse framework. In this paper, we present a formal scheduling model based on constraint programming including evaluation of its prototype implementation. CoUniverse is designed to utilize external media applications, so that a wide variety of existing tools can be used. The whole system has been prototyped and demonstrated, e. g., during international demonstration on the GLIF 2007 workshop.

1 Introduction

The Grid environment is nowadays understood not only as a manner how to share computational resources or data storage facilities but may be understood in a more general way as an infrastructure for sharing of various types of capacities and for virtual collaboration. In this context it also includes high-quality collaborative environment. High-quality collaborative environment must be able not only to transmit media streams with the best possible quality, but also it has to be capable of accommodating changes in the underlying infrastructure. While multipoint transmissions of low-latency uncompressed high-definition media streams at 1.5 Gbps provide the desired quality, they have very high demands and lack adaptivity to changing networking conditions. Bitrate of such media streams becomes comparable even to the current highest-speed network links

E. César et al. (Eds.): Euro-Par 2008 Workshops, LNCS 5415, pp. 339–351, 2009.

(10 GbE or OC-192) and thus scheduling of media streams to network links needs to be done carefully. Furthermore such an environment comprises large number of components which can become very hard to orchestrate manually. Manual orchestration of components makes virtually impossible reacting to network events in time short enough to minimize impact of events on the users' experience.

In this paper, we propose a self-organizing collaborative environment framework for real-time network transmissions called CoUniverse. CoUniverse is designed as an application middleware capable of orchestrating collaborative environments like the one described above. Careful separation of control plane from data plane within CoUniverse allows for optimization of these two networks for different purposes. For the control plane, CoUniverse framework uses peer-to-peer (P2P) network communication substrate which adds necessary robustness and reliability even on experimental infrastructures. We have designed CoUniverse as self-organizing system capable of automated user-empowered steering and encapsulation of legacy media applications (i.e., third-party components of the collaborative environment which are completely unaware of the middleware). Our framework is also capable of responding to changes and outages of the underlying Grid (network and processing) infrastructure. CoUniverse introduces a concept of pluggable scheduler to address the self-organization aspect of the framework by the means of planning the transmissions of the media streams over particular network links and creating corresponding configuration for respective collaborative environment components. Not only that the streams with constant parameters can be configured to individual links, but strategies for using alternative streams and/or adjusting stream parameters may be defined. This allows for using, e.g., 250Mbps compressed stream instead of 1.5Gbps uncompressed when links required for 1.5Gbps stream are not available.

This paper is further structured as follows. Basic design principles used for proposing architecture of CoUniverse are discussed in Section 2. Resulting proposed architecture including overview of basic components and organization of the network is described in Section 3. The system has been prototyped including preliminary version of the scheduler as discussed in Section 4. The system has already been demonstrated during several events and its evaluation especially with focus on performance of current version of the scheduler is given in Section 5. Because the field of collaborative environments is rapidly moving forward, we brief related work in Section 6. The paper is concluded by tackling future research tasks and proposing further applications for CoUniverse in Section 7.

2 Design Principles

The CoUniverse is organized as one or more *collaborative Universes*, where the actual collaboration takes place, and a *Multiverse*, used for registration and lookup of clients and Universes. The collaborative Universes are intended to accommodate collaborative groups of limited sizes [1] and thus can implement functionality that may be hard or impossible to deploy at large. This includes features like sophisticated scheduling and aggressive monitoring of components and network that provides basis for fast reaction to problems that may occur. On the contrary the Multiverse provides a very limited functionality, it has to scale well with respect to large number of participating nodes.

In terms of self-organization, CoUniverse is capable of reacting to events in the system, namely to events raised by users, nodes, and by the monitoring. It includes applications being started/terminated, network links being turned up/down, changes in link parameters (capacity, loss, latency, jitter), nodes being added to and removed from the Universe and nodes being reconfigured.

CoUniverse needs to have a scheduler to support applications with media streams comparable to network link capacity. The scheduling objectives may vary: for simple interactive applications with fixed quality, it usually includes minimization of media distribution latency and possibly minimization of number of nodes involved in the network. For more complex applications where quality is an adjustable parameter, maximization of the quality may also be included. Output of the scheduler has to include not only the plan itself, but also a *workflow* describing how to implement the plan, as there are many functional dependencies. For instance, network links need to be allocated prior to starting media applications that will send data over them.

Because the CoUniverse is designed to integrate high-bandwidth applications, it is necessary to interface with services provided by advanced networks like lambda services [2] or network resource allocators [3].

The whole system follows the user-empowered paradigm [4,5] as much as possible. The CoUniverse doesn't require administrative privileges especially over the network and components which means that the system is able to run entirely in user space.

3 Proposed Architecture

3.1 Network Organization

As discussed above, the CoUniverse is organized as one or more collaborative Universes and a Multiverse. From the networking point of view each Universe consists of a *control plane* used for control communication of all components of the Universe and one or more *data planes* used for actual data exchange between Universe components. Both control plane and data planes are forming an overlay networks on top of an actual physical network infrastructure.

The Multiverse and the control planes of collaborative Universes are based on a P2P networking substrate which provides necessary robustness for the Multiverse and the control planes. Moreover a P2P substrate provides functions like clients and Universes description, naming and addressing, lookups and reliable data transfers.

The data planes of the collaborative Universes are based on available physical networking infrastructure. The data planes are optimized for maximum performance and minimum latency when transmitting data between the components of the Universe. As data planes are virtual overlays over a physical networking substrate, they exist only in case when there is an Application Group (see bellow) to utilize it. The system is designed with user-empowered paradigm in mind and thus it naturally relies on using application-level media “routers” and distributors (reflectors, Active Elements [5]) for multipoint data distribution.

3.2 Collaborative Universe

Collaborative Universes, as shown in Figure 1, consist of nodes, each of which runs Universe Peer client. Universe peers are providing a base for communication among the Universe components, managing underlying node configuration and steering media applications configured on the very node. Nodes within the Universe are aggregated into *network sites*, usually representing all nodes of a single site participating in the collaborative Universe. To give more precise definition, a network site is a set of collocated nodes, where each site may have one or more users participating. Expressed using terminology defined below, typical property of all nodes within one site is that there are no consumers consuming data from producers from the same site (this definition doesn’t include media distributors).

Each network node is configured by specifying *(i)* a list of its physical network interfaces and their parameters, *(ii)* a list of Media Applications which are installed on the node, and *(iii)* a network site the node belongs to. A Media Application is any application which is used to create the collaborative environment and which produces or consumes a media stream (e. g., videoconferencing clients, audioconferencing clients, data distributing Active Elements (AE) [6], etc.). All Media Application producers (except AEs) are producing exactly one media stream which is then sent to exactly one consumer.

Media applications are organized into Application Groups (AG). AGs are then generalizing a particular functionality of the collaborative environment

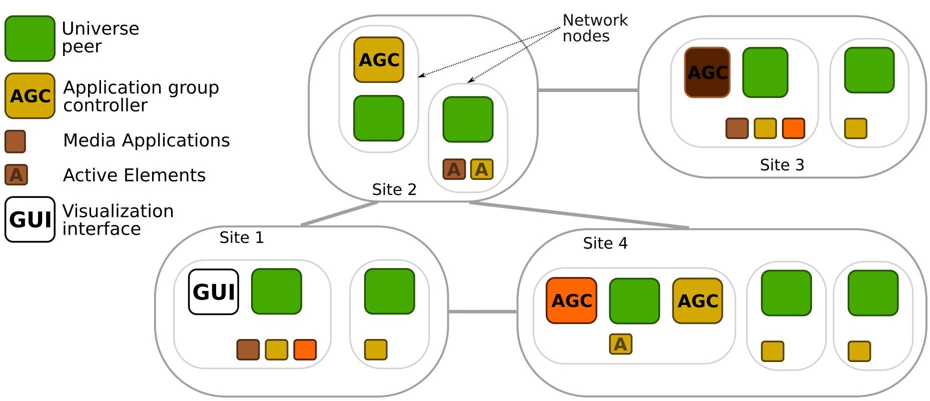

Fig. 1. Scheme of the Collaborative Universe with its components. Different colors for Media Application squares mean different media applications.

(e. g., audio or video conferencing, desktop sharing etc.). Media applications within an AG are orchestrated using an Application Group Controller (AGC). AGC is a service running on top of at least one of the regular Universe peers.

The purpose of the AGC is to collect node configurations from all peers within the collaborative universe, assemble a topology of universe data planes, invoke a scheduler to schedule the media streams of corresponding media applications to a physical network links, create a configuration for each media application based on scheduled media streams and finally send the configuration together with data plane topology to respective universe peer. The universe peer in charge then adjusts the configuration of steered media application so that it corresponds to respective scheduled media stream. The scheduler within the AGC is invoked either manually (especially for the first time) or automatically as a reaction to a change Collaborative Universe state (e. g., new node appeared, a node is not reachable using a particular network link etc.).

3.3 Monitoring

Each Universe peer comprises monitoring of steered media applications, network links of a physical networking substrate which might be used to build the data plane for the media applications and the network links that are actually part of some data plane. Monitoring of the data planes network links is more aggressive than monitoring of network links of generally available physical networking substrate since the links of data planes are actually used for media applications data exchange. At the same time, the links that are not used in any of the Universe data planes need to be monitored less frequently just so that the AGC eventually has a notion of their state when some event in the Universe occurs and those links might be used for some newly scheduled media streams.

3.4 Visualization

Visualisation gives an overview of an actual collaborative Universe state to the user. Our goal is to provide a dynamic visualisation displaying on one hand topology of the physical network between nodes of the collaborative Universe, which might be used to build the data planes, and on the other hand active (currently scheduled) media streams. Visualisation of active media streams is extremely useful especially when incorporating data from network and applications monitoring. Moreover, users can also easily find out whether the schedule chosen for a given network topology has the desired effect (i.e., users can see, talk to, or collaborate with each other in the way it was intended in a particular collaborative universe).

3.5 Scheduling Network Model

In order to describe the scheduling algorithms implemented into the CoUniverse framework, we need to introduce formal notation first. In this section, only the notation is described, while the actual constraints used for scheduling are available in Section 4.

Let I be a set of all network interfaces, $i \in I$ a network interface. Furthermore let N be a set of all nodes in the Universe, $n \in N$ a particular node. Then $\text{node}(i) = n$ where $n \in N$ is a node n with configured network interface i.

Let $l = (i, j)$ be a network link for $i, j \in I$. Then $L = I \times I$ denotes a set of all network links and we denote a particular network link as $l \in L$. We can define following properties of a network link l: $\text{begin}(l) = i$ such that $l = (i, j) \wedge i, j \in I$ is the originating interface i of the link l, $\text{end}(l) = j$ such that $l = (i, j) \wedge i, j \in I$ is the ending interface j of the link l. $\text{cap}(l)$ denotes the link capacity.

Finally, let P be a set of producers where $p \in P$ is a media application producer, C a set of consumers where $c \in C$ is a media application consumer and M set of media distributors where $m \in M$ is an Active Element (AE). Producers, consumers and media distributors are running on the nodes $n \in N$. Let $\text{consumers}(p)$ where $p \in P$ be a set of consumers for a particular producer p. Thus $\bigcup_p \text{consumers}(p)$ is a set of all active consumers, i.e., those that have requested a data stream from some producer. In the opposite direction, $\text{producer}(c)$ is the requested producer for the consumer c. Furthermore we define $\text{node}(p) = n$ where $n \in N \wedge p \in P$, $\text{node}(c) = n$ where $n \in N \wedge c \in C$ and $\text{node}(m) = n$ where $n \in N \wedge m \in M$ as a parent nodes of the producer p, the consumer c and the media distributor m. A media application producer $p \in P$ is producing a media stream with minimal bandwidth $\text{min\_b}(p)$ and maximal bandwidth $\text{max\_b}(p)$.

4 Prototype Implementation

A prototype implementation of CoUniverse[1] uses a current stable version of JXTA [7] P2P framework to implement CoUniverse control plane. Both Multiverse and collaborative Universes are implemented as user name and password authenticated private JXTA peer groups separated from public JXTA P2P network. Current implementation of Multiverse lacks most of the functionality mentioned in previous section and is used just for Universe registration and static lookup.

Current prototype implementation of the CoUniverse uses just one AGC to orchestrate all applications within the collaborative universe. We are using a single AGC to simplify the implementation and to avoid synchronization issues between several AGCs running at the same time. We implemented an interface for steering of generic media applications. In the current implementation of CoUniverse, the Universe Peer is able to control a variety of UltraGrid flavors [8] for both uncompressed and compressed full 1080i HD video transmissions—compared to description in [8], bitrates from 250 Mbps to 1.5 Gbps are now also supported, based on several compression and bitrate reduction algorithms. Amongst other supported applications are: VideoLan Client[2] for HDV video transmissions, VIC[3] for low bandwidth videoconferencing (used as a fallback for building of the collaborative environment) and RAT[3] tool for audio transmissions.

[1] Java sources and JAR archive of the CoUniverse are available at https://www.sitola.cz/CoUniverse

[2] http://www.videolan.org/

[3] http://mediatools.cs.ucl.ac.uk/nets/mmedia/

Media streams scheduler was implemented as a constraint-based solver using a Choco solver library[4]. The solver searches for a solution which is a mapping of media streams on particular network links. Formally we are looking for a set of *stream links* $SL = L \times P$. Scheduler plans the stream links so that $(l, p) = 1$ where $(l, p) \in SL$ for a stream link that is planned to be actively used for the data distribution in the Universe and $(l, p) = 0$ where $(l, p) \in SL$ for an unused stream link. For sake of brevity in the text below, we say that stream link (l, p) *exists* iff $(l, p) = 1$.

Speaking in terms of network model given in previous section the constraints for the solver look as follows:

Stream links constraints. Parent network link l of the stream link must have sufficient capacity to transmit the media stream p. Each stream link must have producer or media distributor on its beginning node and each stream link must have a consumer receiving data using the stream link.

Producer constraints. More than one consumer for a particular producer means that there cannot be any direct stream link between consumers and respective producer as the producer has to send the media stream through at least one media distributor.

Consumer constraints. The media stream for each active consumer is received using exactly one stream link. There are no media streams for any of inactive consumers (i.e., those that hasn't requested any data from any producer) and each active consumer has to be covered by the requested producer either directly or through some media distributor.

Data distribution tree constraints. The number of used stream links for producers with only one consumer is greater or equal to the number of producers. That means data may go either directly, or through some forwarding media distributor (typically in case that direct sending is not available for one reason or another). The number of stream links obviously must not exceed the number of all the media distributors in the network plus one. Moreover a minimal number of used stream links is greater or equal to the number of consumers for given producer plus one for a multipoint data distribution.

AE constraints. A single media distributor instance can only serve for distribution of data from a single producer. Any media distributor is not scheduled together with another consumer for the same producer on a single node and there has to be at least the same number of egress media streams as ingress media streams for a particular AE.

Link capacity constraint. A single constraint for link capacities is stressing that the bandwidth requirements of all the scheduled stream links (l, p) must not exceed the capacity of the link l the stream links are bound to.

[4] http://choco-solver.net/

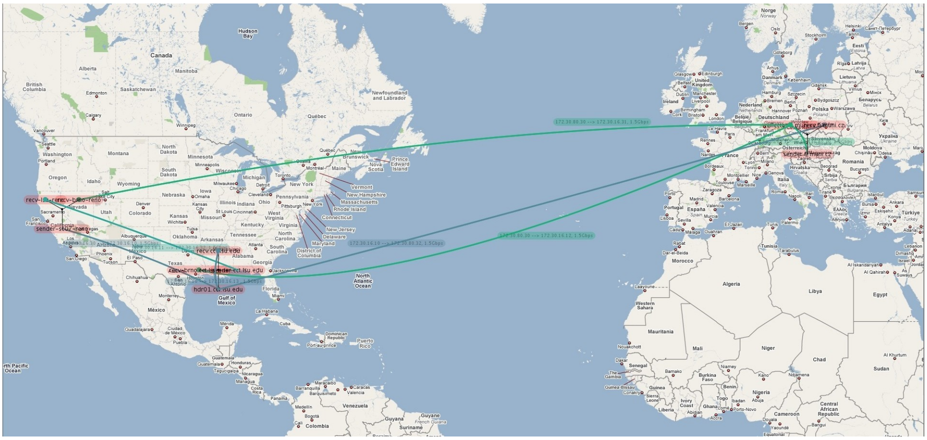

Fig. 2. Visualization of scheduled stream links in CoUniverse during SC'07 demonstration

Another available constraint is based on an elimination of intra-site links (i.e., links (l, p), where node(begin(l)) and node(end(l)) belong to the same site). This can speed up the scheduling up to 10× for many scenarios, but it may also disable some useful solutions, e. g., those where media distributors are collocated in the same site with producers and/or consumers. In case of need, this can be however circumvented by moving media distributors to a separate site.

Based on its settings, the solver can return just a single first match solution or a solution optimized for a minimal media streams distribution latency between the nodes. Based on the network topology and configured media applications the solver may also return a number of equivalent (and even optimal) solutions. In such case the first solution is used and deployed within the collaborative universe.

Because Java lacks any reliable tools for network connectivity monitoring, we have implemented a custom client-server based ping tool. The tool measures not only availability of the peers through the native network, but also network round-trip time, which is an important parameter for latency minimization in our scheduler model. Each universe peer is running the server part implicitly and then is pinging all other known universe peers. In section 3, we mentioned that we need more aggressive monitoring of those network links which are part of some data plane and are used for media applications data exchange than of those network links which are just generally available in the network substrate. This is implemented by a priority and default classes of links which are monitored. A ping client is invoked each second for each network link with scheduled media stream (which is put into the priority class) and each 10 seconds for network links in the default class.

In our prototype, we have implemented a semi-static visualization (see Figure 2) of the collaborative Universe. The visualization is updated with every new scheduling of media streams within the Universe. Currently the visualization shows only active scheduled media streams with some rudimentary description and static parameters of the media streams. However, even such a simple visualization is helpful to check that the collaborative Universe is started up and configured as was intended to.

5 Prototype Implementation Evaluation

Performance and scalability of the CoUniverse environment heavily relies on the scheduler, therefore we have performed a number of simulations with various network topologies and data distribution schemes and measured the time necessary to obtain a schedule for given network topology and distribution scheme.

We chose a full mesh m:n, 1:n tree and direct 1:1 data distribution schemes as a test cases for evaluation of the scheduler performance and scalability. The network topologies were given by the data distribution schemes and a number of sites in the collaborative universe. The m:n distribution scheme test case topology was generated so that each site had one node with an UltraGrid media application producer a node with UltraGrid consumer for each other site and a node with AE. This scenario simulates full-mesh collaboration among peers. The 1:n tree distribution test case was generated so that one site had an UltraGrid producer node and UltraGrid consumers node for all other sites in the topology, every other site comprised of one UltraGrid producer node and one UltraGrid consumer node. This scenario is realistic, e.g., for virtual classroom type environment, where the lecturer gives his talk in multiple remote rooms in parallel. A corresponding number of AE nodes was generated with respect to the fact that one AE can replicate 1,5 Gbps media stream from an UltraGrid producer to at most 6 UltraGrid consumers where 10 Gbps network link is available. Finally direct 1:1 data distribution was a simple test case with generated pairs of UltraGrid producer nodes and UltraGrid consumer nodes, where each UltraGrid consumer was receiving the media stream from a particular preconfigured UltraGrid producer. This is sort of an artificial scenario to show scalability limits. All nodes had one 1 Gbps and one 10 Gbps network interface configured in all three test cases.

All measurement results were obtained on a 2 GHz Pentium M machine with 1 GB of RAM running a Linux operating system. A 1.2.05 version of Choco solver library was used. Table 1 shows excerpt of measured times necessary to find feasible plans for above mentioned test cases with the Choco solver set up to return all feasible solutions and the corresponding times measured for the Choco solver set up to return just the first feasible solution and exit immediately. The table shows that Choco solver scales reasonably for 1:n and direct 1:1 data distribution schemes with up to 25 nodes in the network topology. The worst scheduler performance was observed for m:n data distribution scheme. For such a scheme we were able to obtain a schedule in a reasonable amount of time for up to 12 nodes aggregated into 3 sites.

Table 1. MatchMaker evaluation

Distribution scheme	Sites	Nodes	Network links	Media applications	Active Elements	Scheduling time (first solution only) [s]	Scheduling time (all feasible solutions) [s]
m:n	2	6	60	6	2	0,308	0,178
m:n	3	12	264	12	3	0,447	0,510
m:n	4	20	760	20	4	1986,047	1970,540
1:n	2	5	40	5	1	0,169	0,181
1:n	4	11	220	11	1	0,285	0,360
1:n	6	17	554	17	1	0,758	0,753
1:n	7	20	760	20	1	0,924	1,110
1:n	8	24	1104	24	2	3,747	8,914
1:n	10	30	1740	30	2	17,518	37,299
1:1	2	4	24	4	0	0,187	0,187
1:1	5	10	180	10	0	0,343	0,333
1:1	8	16	480	16	0	0,862	0,979
1:1	11	22	924	22	0	1,900	2,009
1:1	14	28	1512	28	0	3,382	3,344
1:1	17	34	2244	34	0	5,745	6,160
1:1	20	40	3120	40	0	9,727	10,161

5.1 Demonstrations

A prototype implementation of CoUniverse was evaluated during SuperComputing'07 event and a demonstration at GLIF 2007 meeting. The CoUniverse was used to orchestrate a network of twelve nodes using a high quality, high bandwidth HD video transmissions and audioconferencing to create a multipoint-to-multipoint collaborative environment connecting three sites (Louisiana State University, USA with Charles University, Czech Republic and Academia Sinica, Taiwan).

Creating such a collaborative environment means in praxis configuring and steering of more than two dozens of media applications and Active Elements to bring up the media streams connecting all the sites. Configuring all media applications and AEs at dozen of machines presents a huge amount of manual work which is overwhelming for users of such environment. Moreover there must be at least one user of the collaborative environment having precise idea how to create the media streams between all media applications and AEs based on knowledge of available physical network substrate between all participating sites. Last but not least the users are not able to ensure resiliency and fast recovery of such an environment in case of any network, node or media application failure, because it might mean even newly configuring of all nodes and applications.

Both issues were well addressed deploying CoUniverse. Although creating node configurations for all nodes in the Universe is initially quite time consuming as well, users have to create just a local configurations describing network interfaces of the local machine and the location of local media applications. The SuperComputing'07 demonstration showed that CoUniverse is also able to respond to changing networking conditions when parts of 10 GbE infrastructure used for the HD video transmissions went down and back up for a couple of times during the demonstration.

6 Related Work

As mentioned in the introduction, some extent of self-organization is usually built into the all but the simplest collaborative tools. H.323 and SIP tools that are considered a sort of industrial standard as a videoconferencing platform can accommodate changes in available link capacity by changing compression parameters of media streams. Isabel [9] platform has similar properties by means of flow server and also features programmable floor control [10], which is however on the level of GUI programmability only.

Probably closest to CoUniverse idea is currently VRVS EVO [11], which allows self-organization of the collaboration network. It is however a closed system that doesn't incorporate external tools and namely it is designed to work only with a low and standard-definition media streams that have bandwidth requirements significantly lower than the link capacity. From user perspective, VRVS EVO can be viewed as a system similar to Skype in terms of both self-organization of the network and usage of low quality media streams.

Another important videoconferencing platform is AccessGrid [12]. AccessGrid is capable of providing high-definition media streams. However, AccessGrid doesn't have any self-organizing properties. The fail-over mechanisms are only very simple and have to be initiated manually by the user, e. g., by selecting unicast media transport instead of multicast. Compared to the other systems, it may seem simple, but it follows several of CoUniverse design principles which the other systems are not compliant with: user-empowered paradigm at least for the collaborative system components (which are open-source and may be installed by end-users arbitrarily) and it is also extensible and incorporates external applications (e. g., UltraGrid to support high-definition media streams).

7 Conclusions and Future Work

In this paper, we have designed a framework for advanced self-organizing collaborative environments called CoUniverse and described its prototype implementation. The system is targeted to incorporating high-end multimedia tools while utilizing advanced high-speed networks with their specialized services.

While the CoUniverse has been designed primarily with the high-end videoconferencing systems in mind, it can be very useful beyond this domain. Any component-based applications with real-time orchestration requirements can be supported. For example, if a scientific instrument, that is generating real-time data, is needed to be incorporated into the Grid infrastructure and the data is supposed to be distributed to one or more locations in real-time, the CoUniverse can be used to control the data distribution, including the components along the path: data source (i.e., some component that is a direct interface from the instrument to the computer network), data distributors, as well as data receivers (be it storage or real-time visualization systems). It can also allocate dedicated network circuits (e.g., lambda services) prior to starting data distribution and deallocate them after the data transmission is finished. All that is needed to create such an application workflow is to implement a CoUniverse modules for the respective services.

Even though we have already implemented and successfully demonstrated a prototype of the CoUniverse, it still leaves many unanswered questions stated in the introduction to this paper. One big issue is optimization of the scheduling algorithms in order to support larger infrastructures. It should also better utilize knowledge of network structure, even if it is only partial. We want to include scheduling for native multipoint applications. Another issue that needs to be further investigated is programmability of the whole system by its users. This is also important in the context of the scheduler, which may need to be able to incorporate user-defined constraints on its behavior.

Acknowledgments

This project has been kindly supported by the research intent “Parallel and Distributed Systems” (MŠM 0021622419).

References

1. Arrow, H., McGrath, J.E., Berdahl, J.L.: Small Groups as Complex Systems. Sage Publications, Thousand Oaks (2000)
2. Travostino, F., Mambretti, J., Karmous-Edwards, G.: Grid Networks: Enabling Grids with Advanced Communication Technology. John Wiley & Sons, Chichester (2006)
3. MacLaren, J.: Co-allocation of compute and network resources using harc. In: Proceedings of Lighting the Blue Touchpaper for UK e-Science: closing conference of ESLEA Project, vol. PoS(ESLEA)016 (2007)
4. Hladká, E., Holub, P., Denemark, J.: User empowered programmable network support for collaborative environment. In: Freire, M.M., Chemouil, P., Lorenz, P., Gravey, A. (eds.) ECUMN 2004. LNCS, vol. 3262, pp. 367–376. Springer, Heidelberg (2004)
5. Holub, P., Hladká, E., Matyska, L.: Scalability and robustness of virtual multicast for synchronous multimedia distribution. In: Lorenz, P., Dini, P. (eds.) ICN 2005. LNCS, vol. 3421, pp. 876–883. Springer, Heidelberg (2005)
6. Hladká, E., Holub, P., Denemark, J.: An active network architecture: Distributed computer or transport medium. In: 3rd International Conference on Networking (ICN 2004), Gosier, Guadeloupe, March 2004, pp. 338–343 (2004)
7. Traversat, B., Arora, A., Abdelaziz, M., Duigou, M., Haywood, C., Hugly, J.-C., Pouyoul, E., Yeager, B.: Project JXTA 2.0 super-peer virtual network, `http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf`

8. Holub, P., Matyska, L., Liška, M., Hejtmánek, L., Denemark, J., Rebok, T., Hutanu, A., Paruchuri, R., Radil, J., Hladká, E.: High-definition multimedia for multiparty low-latency interactive communication. Future Generation Computer Systems 22(8), 856–861 (2006)
9. De Miguel, T.P., Pavon, S., Salvachua, J., Quemada Vives, J.: ISABEL—experimental distributed cooperative work application over broadband networks. In: Steinmetz, R. (ed.) IWACA 1994. LNCS, vol. 868, pp. 353–362. Springer, Heidelberg (1994)
10. Quemada, J., de Miguel, T., Pavon, S., Huecas, G., Robles, T., Salvachúa, J., Ortiz, D.A.A., Sirvent, V., Escribano, F.: Isabel: An application for real time collaboration with a flexible floor control. In: CollaborateCom 2005 (2005)
11. Galvez, P.: Evo: Enabling virtual organizations. In: CHEP 2007, Victoria, Canada (2007)
12. Childers, L., Disz, T., Hereld, M., Hudson, R., Judson, I., Olson, R., Papka, M.E., Paris, J., Stevens, R.: ActiveSpaces on the Grid: The construction of advanced visualization and interaction environments. In: Engquist, B. (ed.) Proceedings of Simulation and visualization on the grid: Parallelldatorcentrum, Kungl. Tekniska Högskolan, seventh annual conference, Stockholm, Sweden. Lecture Notes in Computational Science and Engineering, vol. 13, pp. 64–80. Springer, New York (2000)