

## DSM-Communities in the World-Wide Web

Peter Schulthess, Oliver Schirpf, Michael Schoettner, Moritz Wende,  
Distributed Systems, Ulm University,  
schulthess@informatik.uni-ulm.de

**Abstract.** We contemplate extending the applicability of our current implementation of a DSM operating system from the locally connected PC cluster to large scale intranets and multiple federated DSM domains in a global network context. Potential gains in functionality, speed, consistency and elegance are expected.

### 1 Introduction

The grand vision of a globally interconnected information space has found its preliminary implementation in the world-wide-web. However, there is room for improvement in several aspects:

- connection oriented protocols do not efficiently transfer single information pages,
- www information pages are burdened with irrelevant advertisements,
- the consistency of the URL-pointers is currently not guaranteed,
- the world-wide connectivity represents a security hazard,
- there is no logical concept of a closed user group,
- world-wide web pages are read-only items,
- the world-wide web is too slow.

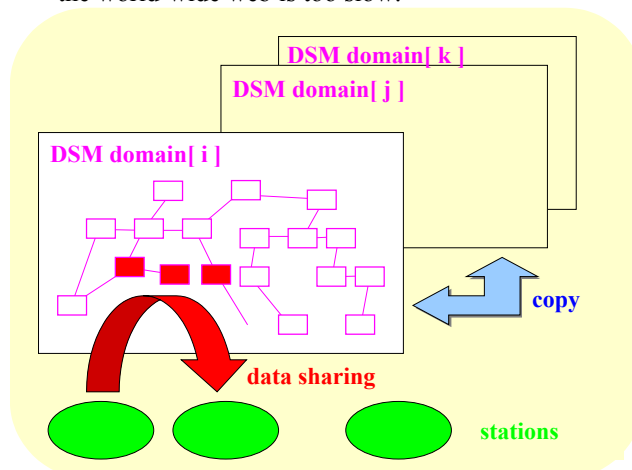


Fig. 1. Federated DSM-Communities

We certainly are enthusiastic about the advent of internet services and the word-wide web in particular and feel that our development in the area of distributed shared memory (DSM) might present a partial solution to the problems quoted above. We suggest the establishment of a multitude of separately managed domains of interest, so called DSM-communities. Topics for such domains might range from "Scientific Computing Resource Pool" or "Twenty First Century Lifestyle Conversion" or "North American Baseball News" to "Bavarian Oktoberfest Attractions" or "Sears Roebuck Departments Stores" or "Distributed Data of Peter Schulthess".

The information of one domain of interest (DSM-Community) might be stored in a single distributed memory space and the links between information texts (hypertext) can be regular pointers. Appropriate transaction protocols guarantee the consistency of objects in the shared memory space. Within each address space information retrieval and distributed applications are easily programmed obeying a simple sequential execution model (Figure 1). Objects in external memory spaces are accessed using "copy semantics" instead of a "shared data semantic".

## 2 Mode of operation

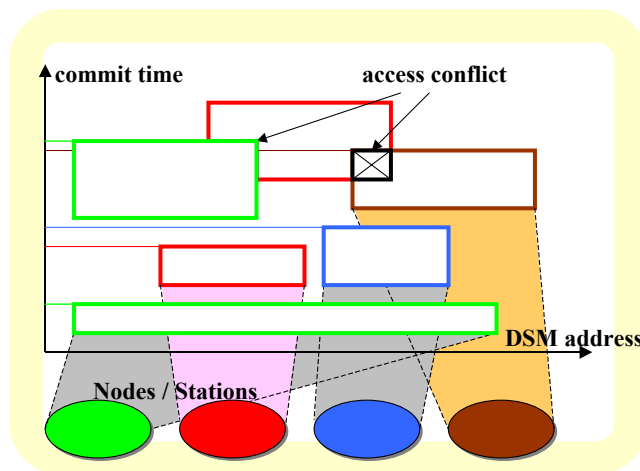


Fig. 2. Transactional DSM Storage

The envisaged DSM-communities in the world-wide web shall be implemented as multiple disjoint DSM address spaces. DSM (Distributed shared memory) is an interesting alternative to message-based construction of distributed systems. A major challenge lies in keeping the objects in DSM storage consistent according to some selected consistency model [1]. However, as long as a DSM page remains unmodified multiple copies of it may freely reside in separate nodes. This is particularly convenient for web pages and program code. We imply that if a web space is realized as a DSM communities it will work faster, more securely and offer additional functionalities.

Our Plurix operating system uses the model of transactional consistency [2] for the DSM storage areas (Figure 2): Competing transactions in different nodes access local copies of DSM storage and attempt to commit their actions at the end of each transaction. A commit request distributes its read/write-set onto the network invalidating all modified pages in eventual partner nodes if it is successful. Because the system classes are very light-weight user interactions tend to be comparatively short transactions. In case of a collision with another transaction the aborted transaction is automatically restarted.

### 3 Limitations of domain size

A major limiting factor for the size of a DSM domain is the achievable transaction rate. Transactions on individual stations will execute concurrently but commission of the modified pages must be propagated to all participating nodes. In a local PC-cluster propagation of commit requests to all stations may be fast and in the order of 50 microseconds - leading to a limit of 20000 transactions per second. In a wide-area cluster, however, the propagation delay may be 50 milliseconds or higher. The resulting rate of 20 transactions per second is clearly unacceptable. Therefore in a wide-area DSM domain each node must overlap the current transaction with the commit request of the previous transaction (Figure 3: Pipelined transactions). These pipelined transactions carry the potential of achieving high transaction rates even in a wide-area DSM domain but require that not only the current but also the previous transaction is kept restartable. Restarting a previous transaction will automatically restart subsequent transactions.

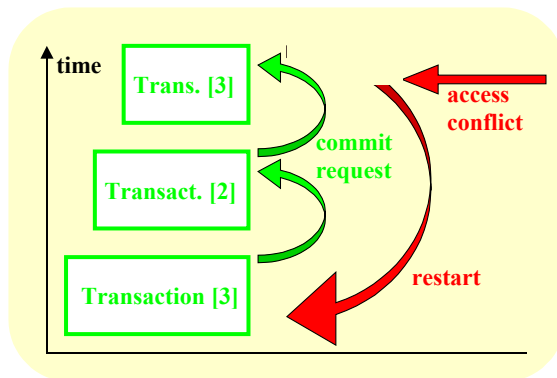


Fig. 3. Pipelined transactions

The maximum size of a DSM domain is given by the address length of the pointers. The current implementation therefore limits its domain size to 4 Gigabytes. By using separate segment descriptors current Intel CPUs can extend their addressing capability to 46 bit. However, since segment descriptors are not supported by the virtual memory mechanism we decided against including them into the container size

of a pointer in our prototype. With the advent of 64 bit architectures all practical addressing limitations will be removed. Instead of a physical addressing restriction the desire of the owning institution to control content and access rights of a DSM domain will most likely limit its size to less than a  $2^{64}$  bytes.

#### 4 Advanced Application 1: Telecooperation

Telecooperation (within a DSM Community) might serve as an example of the applicability of our approach beyond traditional web browsing. Transactional structures are natural for telecooperation scenarios. Often several participants will jointly edit a document. The document will reside in DSM storage and is easily shared. Modifications are directly written to the document and propagated to all participants during the commit phase. Conflicting modifications are detected and restarted in the context of their transaction. Depending on the modification history reformatting on the screens will be initiated.

Beyond editing of shared documents DSM storage will also be appropriate for the audio and video streams which support the telecooperation scenario. Various options may be selected when designing the telecooperation scenario: It can be a mere teleconference, windows on the screen may be shared or the shared object may be a document or an application [3]. Document sharing is depicted in Figure 4. Telecooperation may be useful within a research team, for teleworkers, for business meetings, for remotely teaching courses and for a plethora of other situations.

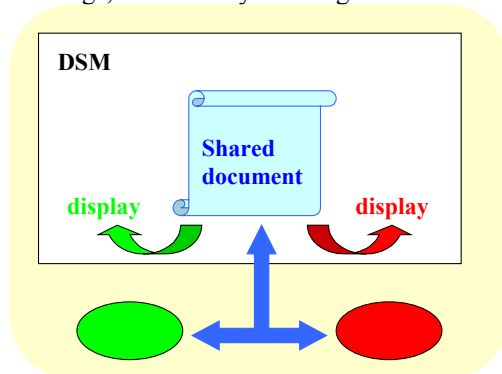
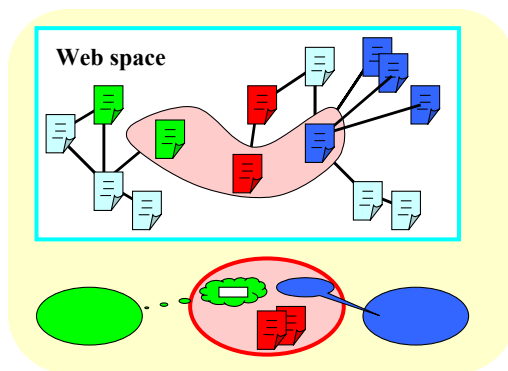


Fig. 4. Document sharing in a DSM environment

#### 5 Advanced Application 2: Collaborative browsing

The CoBrow project [4] implements a communication model which deviates from the traditional connection setup paradigm. Meetings in the net (in cyberspace) occur

spontaneously and not by explicit invocation of a telecooperation session. People will accidentally meet when visiting specific web pages and the presence of other people near the current page is indicated by appropriate applets or plug-ins. Contacts are made possible using chat-, voice- or video tools. DSM storage may hold additional dialog boxes for video, audio and chat communication. Underlying the CoBrow system is a complex database storing the location, the preference profiles and the personal communication page of individual users. Based on this information the virtual distance between users is computed and an appropriate subset is displayed to each participant. This vicinity database is subject to a relaxed consistency scheme only and can easily reside in DSM storage (Figure 5).



**Fig. 5.** Vicinity in "Cyberspace"

## **6 Plurix Implementation status**

The Plurix project at Ulm University implements a native DSM operating system [5]. The resources of a cluster of PC machines and its distributed shared memory is managed by a set of approximately 20 Java classes (Figure 4 below). All objects reside in persistent DSM storage and can be accessed via regular pointers. A native Java compiler [6] compiles application classes and operating system classes directly to 32-bit Intel code. Additional Java language constructs simplify the programming of drivers and interrupts and provide direct access to the hardware. The core classes support transactions in the DSM context and soon the splitting and fusion of separate domains. The overall system structure is patterned after the archetypical Oberon system developed by Wirth & Gutknecht [7]. It uses a central event loop in each station and a simple cooperative multitasking concept. Currently a single station will only support one DSM domain within a 32 bit address space. The system has been successfully demonstrated at various trade fairs and proved to be fast, compact and reliable.

## **7 Perspective**

The Plurix DSM operating system is intended as a generic operating system including special support for distributed operation within multiple separate domains. Future work concentrates on streamlining the current prototype and extending the address space to 64 bit. This implies a rewrite of the compiler and of the memory management component. Stations supporting preemptive tasking will be able to participate in several DSM domains simultaneously. Further evolution of the Internet will establish communication with guaranteed low latency within a closed user group. To boost the transaction rate the concept of pipelined transactions and more elaborate DSM protocols are considered. Introducing DSM communities requires a migration path from the current world-wide web to a more integrated solution. Initially DSM operation might be embedded in commercially available browsers later on partitions of the internet might develop which are entirely based on the principle of DSM storage, interaction and consistency. On a longer time scale identifying and managing individual domains in the context of a consistent network operating system will be more realistic and acceptable than a uniform global information space.

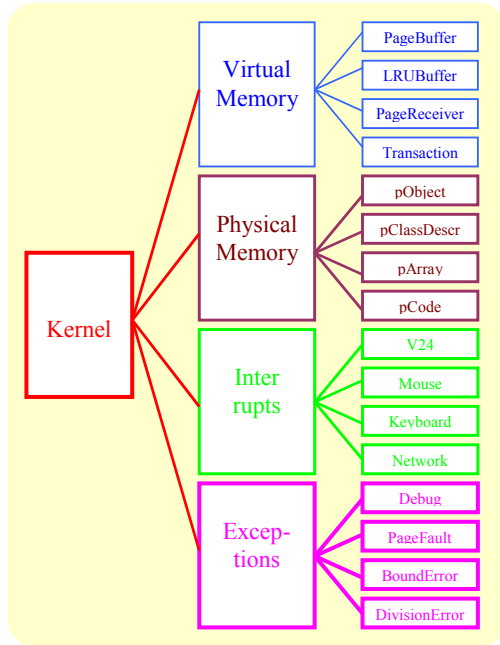


Fig. 6. Object oriented architecture of the Plurix System

## 8 Literature

1. Mosberger, D.: Memory consistency models Operating Systems Review Vol. 27. No. 1 pp. 18-26. ACM press.
2. Schoettner M., Traub S. and Schulthess P.: A transactional DSM Operating System in Java. Proceedings of the 4th International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 1998
3. Dermler G., Froitzheim K.: JVTOS - A Reference Model for a New Multimedia Service; 4th IFIP Conference on High Performance Networking (hpn 92). Liège, 1992.
4. Sidler G., ETH Zurich; Scott A., Lancaster University; Wolf H., University of Ulm; Collaborative Browsing in the World Wide Web. Proceedings of the 8th Joint European Networking Conference, Edinburgh, May 12.-15. 1997 .
5. Recent Plurix status: <http://www-vs.informatik.uni-ulm.de/projekte/plurix.html>
6. Schoettner M., Schirpf O., Wende M. and Schulthess P.: Implementation of the Java language in a persistent DSM Operating System. Proceedings of the 5th International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 1999
7. Wirth, N., Gutknecht J.: Project Oberon. ACM Press, Addison-Wesley New York 1992