TECHNISCHE FAKULTÄT

CHAIR FOR COMPUTER SCIENCE 6
DATA MANAGEMENT

DEBS2010
4th ACM International Conference on Distributed Event-Based Systems

Event Semantics in Event Dissemination Architectures for Massive Multiuser Virtual Environments

Thomas Fischer, Richard Lenz

Scenario

Massive Multiuser Virtual Environments (MMVEs):

Massive — A certain scale has to be reached. Current Architectures serve up to 50.000 users in one world [1].
Multiuser — It is a distributed application with multiple users, which are represented by interconnected clients.
Virtual **E**nvironments — The system simulates a persistent world in real-time.

In MMVEs thousands of players share and compete in one persistent world. From a scientific point of view such a system is conceptually a discrete event based simulation in a distributed environment. Therefore the state changes of each client is published via events and the corresponding server computes consistent world states from those events, which are then published back to the clients via events.

In most MMVEs, the chosen event processing paradigm to implement those systems is a distributed publish/subscribe concept. This is the conceptual background under which we examine performance optimization under exploitation of event semantics.

Problem Statement

Without any optimization, the initial effort for the computation of a consistent state in an MMVE is $O(n^2)$, n being the number of clients and therefore also the number of messages considering broadcast semantics. Obviously this effort does not scale well. Therefore many optimization approaches have been proposed to reduce the number of messages needed. All of these approaches exploit the semantics of the events in some kind.

The key to large scale MMVEs to our believe lies in the minimization of the events needed to compute a consistent world state. Our approach aims for a framework for exploitation of event semantics to minimize required messages.

Centralized vs. Decentralized Architectures for MMVEs:

Client/Server Architectures: Current state of the art. These architectures only scale to the current level by usage of different optimizations:
-Cluster and Grid approaches (e.g. partitioning of game world via shards and regions).
-Gameplay optimization (e.g. low simulation frame rate).
-Event dissemination optimization (e.g. area of interest management).

Hybrid/P2P Architectures: Addresses the problem of scalability further. Resources of clients are used to scale beyond centralized approaches by exploitation e.g. of the locality of messages or the circumvention of physical limitations as well as the lower maintenance costs in contrast to a large cluster. But this approaches also have new introduced complexity, as clients may disconnect or be compromised.

In order to enable the development of a framework for optimization based on event semantics, we use a P2P based architecture, as a client/server system in our case is only a degenerated decentralized architecture.

Existing Approaches

We examined some existing P2P based approaches for MMVEs in respect to our classification. Each approach is denoted, whether it provides an optimization mechanism of some kind for the corresponding dimension of our classification. As the table shows, there is no approach, considering all proposed dimensions, despite their relevance in optimization of MMVE architectures.

	Context	Persistency	Synchronization	Validity	Delivery	Security
SimMud [8]	x	x				
Colyseus [4]	x		x			
VON [2]	x					
Rokkatar [9]	x					
Donnybrook [3]	x					
Virtual time [7]			x			
Ferretti [6]			x			
Dead Reckoning [12]					x	
Kabus [11]						x
Mammoth [5]		x				
MiddleSIR [10]		x	x			

Exemplary Classification

We give some common event type examples for MMVEs based on an analysis of the events used in Quake 3 Arena. An exhaustive classification is not possible at this point of our research. Each event type is classified along each dimension and associated with a certain class depending on available optimization strategies. Following selected event types have been found in the Quake 3 Arena source code (Q3A 1.32b Source Code: *http://www.idsoftware.com/business/techdownloads/*) and exemplary classified:

Movement events have a spatial context, transient persistency, causal synchronization, interval validity, desired delivery and no security.

Jump events have a spatial context, transient persistency, causal synchronization, interval validity, guaranteed delivery and preventing security.

Item Pickup events have a spatial context, persistent persistency, sequential synchronization, unlimited validity, guaranteed delivery and preventing security.

Team Message events have a multi-target context, transient persistency, weak synchronization, unlimited validity, desired delivery and no security.

Event Semantics

In order to exploit event semantics we strive for a multidimensional classification schema. To achieve this, two steps are required:

1. Identify all relevant dimensions of event semantics.
2. Define disjoint classes along those dimensions with corresponding optimization methods.

Following initial dimensions have been found for the domain of MMVEs:

Event

Delivery

Some events must reach their destinations, while others like position updates may have such a high frequency, that the loss of a single event does not cause any problems. Therefore the system may have to guarantee the delivery or prioritize it. Depending on different delivery characteristics, the systems may be optimized and reduce events.

Context

Each event in an MMVE has a certain context in which it is relevant or valid. This context may be spatial, social or one or more targets defined by certain metrics. An example for an event with a spatial context is the opening of a box in the virtual environment. Only clients in visual range need to be subscribed to such an event. In general the context of an event in an MMVE reduces its recipients to a certain subset. Most optimization algorithms in this field may be summarized under the topic Area of Interest Management.

Validity

Whilst synchronization describes the order of, or more general the relationship between events, validity is strictly limited to one event, for example an effect on a player which is active for a certain time may be modeled by one event with the corresponding validity.

Synchronization

Some events have certain temporal or causal dependencies and therefore require synchronization. For example a position update may have no synchronization requirements, due to its high update rate. An event representing the pickup of an item from a chest on the other hand needs defined synchronization semantics, because there are causal interdependencies if another player also wants to pick up this item at the same time. There exist many different approaches, all providing different synchronization semantics for different requirements.

Security

A secure event is not tempered and represents the initial event. Especially in distributed MMVE architectures it is important to at least detect cheating clients. Prevention would be the optimum, but in most cases it is too expensive to guarantee cheat-free operation. Because of its impact on the performance of event dissemination we see security of events as a semantically relevant dimension in this context.

Persistency

In contrast to normal multiuser virtual environments, MMVEs provide a persistent world, which leads to the consequence, that some events like e.g. the gain of money have to be persistent in some way. There are two major solutions to this problem: Replication of the state, to ensure enough hosts are always online to restore the state in case of failure or the storage in a centralized database.

References

1. Halldor Fannar, Victoria Coleman, Randy Breen, and Brandon Van Slyke. The server technology of eve online: How to cope with 300,000 players on one server. Talk on GDC Austin, 2008.

2. Shun-Yun Hu, Jui-Fa Chen, and Tsu-Han Chen. VON: A Scalable Peer-to-Peer Network for Virtual Environments. IEEE Network, 20(4):22-31, July 2006.

3. Ashwin Bharambe, John R. Douceur, Jacob R. Lorch, Thomas Moscibroda, Jerey Pang, Srinivasan Seshan, and Xinyu Zhuang. Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games. SIGCOMM Comput. Commun. Rev. (CCR), 38(4):389{400, 2008.

4. Ashwin Bharambe, Jerey Pang, and Srinivasan Seshan. Colyseus: A Distributed Architecture for Online Multiplayer Games. In 3rd Symposium on Networked Systems Design & Implementation (NSDI), pages 155-168, Berkeley, CA, USA, 2006. USENIX Association.

5. Kaiwen Zhang, Bettina Kemme, and Alexandre Denault. Persistence in Massively Multiplayer Online Games. In 7th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames), pages 53-58, New York, NY, USA, 2008. ACM.

6. Stefano Ferretti. A Synchronization Protocol For Supporting Peer-to-Peer Multiplayer Online Games in Overlay Networks. In 2nd International Conference on Distributed Event-Based Systems (DEBS), pages 83-94, New York, NY, USA, 2008. ACM.

7. David R. Jeerson. Virtual Time. ACM Transactions on Programming Languages and Systems (TOPLAS), 7(3):404-425, July 1985.

8. B. Knutsson, Honghui Lu, Wei Xu, and B. Hopkins. Peer-to-Peer Support for Massively Multiplayer Games. In IEEE INFOCOM, volume 1, pages 96-107. IEEE, 2004.

9. Jens M uller and H. Sergei Gorlatc. Rokkatan: Scaling an RTS Game Design to the Massively Multiplayer Realm. ACM Computers in Entertainment (CIE), 4(3):1-14, 2006. Article No. 11.

10. Yi Lin, Bettina Kemme, Marta P. Martinez, and Ricardo J. Peris. Applying Database Replication to Multiplayer Online Games. In 5th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames), New York, NY, USA, 2006. ACM. Article No. 15.

11. Patric Kabus, Wesley W. Terpstra, Mariano Cilia, and Alejandro P. Buchmann. Addressing Cheating in Distributed MMOGs. In 4th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames), New York, NY, USA, 2005. ACM. Article No. 6.

12. Sandeep Singhal and Michael Zyda. Networked Virtual Environments: Design and Implementation. Addison- Wesley Professional, New York, NY, USA, 1999.

Friedrich-Alexander-Universität
Erlangen-Nürnberg

ACADEMIAE FRIDERICO ALEXANDRINAE SIGILLUM

Address
Friedrich-Alexander-University
Erlangen-Nuremberg
Department of Computer Science
Chair for Computer Science 6 (Data Management)
Martensstrasse 3
91058 Erlangen, Germany

Office
Phone: +49 9131 85 27893
office@i6.cs.fau.de

Contact
Phone: +49 9131 85 27901
thomas.fischer@cs.fau.de