Using Epidemic Hoarding to Minimize Load Delays in P2P Distributed Virtual Environments

Ingo Scholtes[1], Jean Botev[1], Markus Esch[2], Hermann Schloss[1], and Peter Sturm[1]

[1] Systemsoftware and Distributed Systems, University of Trier, D-54286 Trier, Germany
{scholtes,botev,schloss,sturm}@syssoft.uni-trier.de
http://syssoft.uni-trier.de

[2] Faculty of Sciences, Technology and Communication, University of Luxembourg, L-1359 Luxembourg-Kirchberg, Luxembourg
markus.esch@uni.lu

Abstract. Distributed Virtual Environments (DVEs) have grown popular in various fields of application. Apart from providing great collaborational opportunities in an immersive setting, large-scale DVEs pose severe scalability challenges. Although P2P approaches have proven to be effective for tackling many of these issues, still load delay problems remain in regions with high object or avatar density. In this article we present and evaluate a hoarding approach that is suitable to minimize such delays in P2P-based DVEs with a real-time distribution of dynamic data. The prediction of what data shall be hoarded is based on an epidemic aggregation algorithm working solely with local knowledge. Evaluation results that have been obtained using a DVE simulation environment will be presented.

Keywords: DVE, P2P, Hoarding, Gossiping, Epidemic Aggregation.

1 Introduction

The potential of Distributed Virtual Environments (DVEs) for facilitating immersive and intuitive user collaboration is well-recognized. Until now being predominant in the form of Massively Multiplayer Online Games (MMOGs), they are increasingly used for other purposes like e.g. learning or telepresence. While today MMOGs with several thousand concurrent players constitute the largest DVEs, it is justifiable to think about future environments with hundreds of millions of concurrent participants. An interesting example for such a global-scale future DVE is a 3D representation of the real world in which all kinds of information are embedded. In analogy to Neal Stephenson's novel "Snow Crash", this vision of a virtual environment as major future Internet application is often called "MetaVerse".

E. Bertino and J.B.D. Joshi (Eds.): CollaborateCom 2008, LNICST 10, pp. 578–593, 2009.

The realization of this scenario poses severe technical challenges. Current MMOGs usually rely on several gigabytes of pre-supplied data. This approach is not practicable when considering future environments with huge amounts of dynamic objects. While the real-time distribution of these data is alluring, it poses severe scalability issues to any centralized infrastructure. Even with today's - comparably moderate - user numbers and pre-supplied data, server-based DVEs are suffering from scalability problems. In order to overcome these issues, P2P-based approaches have been proposed over the last few years. While these have effectively addressed server-side scalability issues, in crowded virtual regions there remain problems that result from inevitable limitations of the clients' bandwidth. Having motivated the resulting load delay issues in more detail in the following section 2, we describe a scalable probabilistic solution in section 3. It has been implemented within the HyperVerse project[1] [4], which investigates a P2P-based infrastructure for global-scale DVEs providing flawless user experience in face of unlimited user numbers and population densities. The proposed algorithm has been implemented in a simulation environment for DVEs. Evaluation results will be presented in section 4 of this article.

2 Motivation

In order to achieve scalability in face of an unlimited amount of objects and participants, it is crucial for P2P-based DVEs to minimize the state required in each component. Interest management (IM) schemes have been developed for this purpose [20]. Due to the implicit locality of interest, usually space-based approaches are used in which a client's knowledge about objects and other users is limited to a certain surrounding area. In the following we intend to motivate load delay problems that arise from the usage of a dynamic, space-based IM scheme in the HyperVerse infrastructure. Similar ones are widely deployed in many other DVEs. The scheme is based on Euclidean distance and defines two spheres around a user's position p in the virtual world. A sphere with radius d defines the user's Field of View (FoV), i.e. the range within objects and other avatars shall be visible. A second sphere with radius $d + \Delta$ is called Area of Interest (AoI) and represents the range within a client is aware of objects and other avatars. The motivation for using an AoI is to make clients aware of nearby objects and avatars before they enter the FoV so that there remains enough time to prefetch data necessary for rendering.

We assume that the AoI remains static unless the user moves more than a certain distance Λ away from its center. In this case (see e.g. position p'' in Figure 1) a new AoI centered around the user's current position will be computed and objects therein will be retrieved. Depending on the client's capabilities, motion speed and object density, the sizes of the FoV and the AoI as well as the threshold Λ need to be adapted. Since it minimizes a client's knowledge about its environment, the AoI size should be as small as possible. It must however be large enough to provide sufficient time for data retrieval.

[1] http://hyperverse.syssoft.uni-trier.de

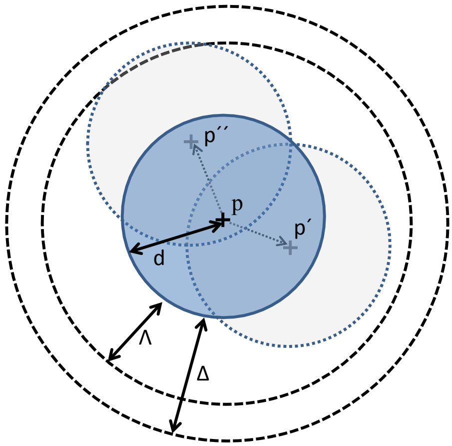


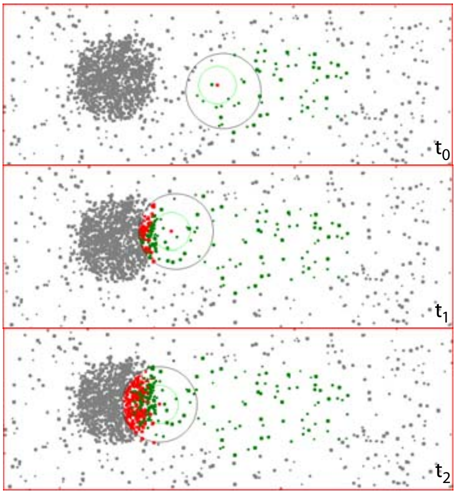
Fig. 1. Area of Interest (AoI) and Field of View (FoV) of a client

Using the IM scheme described above, load delays may occur when approaching densely populated areas, so-called hot spots. Such a situation is illustrated by the following example. Let's assume a client becomes aware of a hot spot within its AoI at time t_1. Let's further assume at t_2 the client has moved to a point where these objects are within its FoV. This situation can be observed in the simulation snapshots in Figure 2. The outer circle is the client's AoI, the inner circle represents its FoV.[2]

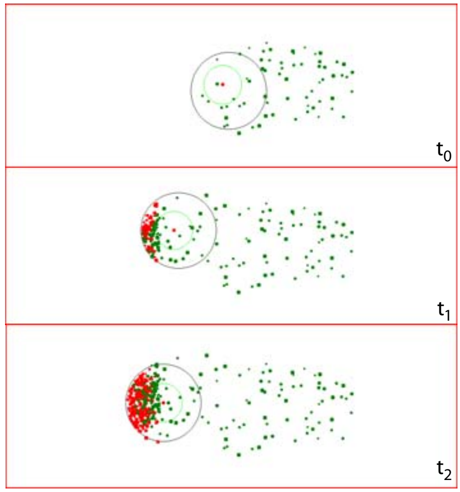
If - based on the client's downlink bandwidth - the time taken to retrieve data for all objects in the client's FoV at time t_2 exceeds $t_2 - t_1$, there will necessarily occur a load delay, even assuming unlimited resources at the data provider. Consequently, the question arises how such situations can be avoided. For this, clients must be made aware of hot spots at time $t_0 < t_1$, so that $t_2 - t_0$ is sufficient to retrieve all relevant data. As can be seen in Figure 2(b), at time t_0 the client has however no knowledge about the nearby hot spot.

One possible solution would be to increase the client's AoI, so that the hot spot can be identified at time t_0. Since it conflicts with the minimization of a client's knowledge horizon, this is however not a scalable solution. This becomes especially clear when considering dynamic objects and users. A hot spot resulting for instance from a user crowd can exist at the same position for a long time regardless of the dynamics of its constituents. Tracking a huge number of dynamic entities in a wide range in order to identify hot spots clearly is not a scalable solution. Since in P2P DVEs there is no central instance with global knowledge the question arises how the "crowd wisdom" of peers can be utilized to solve these issues. In the following section we describe a scalable solution for this problem.

[2] As can be seen in Figure 2(b), in the simulated situation the client is aware of some objects outside it's direct awareness radius: These are cached objects that have been discovered on its way towards the current position.



(a) Global View



(b) Local View of Client

Fig. 2. Problem Motivation

3 Epidemic Hoarding

The basic idea that can be used to prevent load delays when entering hot spot regions is to get rid of the inherently bursty traffic pattern that results from the IM scheme described in section 2: Traffic bursts occur whenever the AoI changes. In the following we will answer the question whether it is possible to distribute traffic across time better and thus equalize these bursts. For this we propose a hoarding mechanism in which a certain fraction of a client's bandwidth is constantly dedicated to speculatively prefetch data regardless of a client's AoI. If this bandwidth is used to selectively prefetch data only from within hot spot areas, load delays can be minimized or totally anticipated. For this, hot spots need to be identified based on a peer's local knowledge. In the following we will show how this information can be efficiently retrieved using a modification of the epidemic algorithm described in [15]. Since for this the P2P overlay topology of the HyperVerse infrastructure is used, we will briefly describe it in the following section.

3.1 P2P Overlay Topology and Data Distribution

For the sake of scalability in the face of a potentially unlimited number of users, within the HyperVerse infrastructure data are exchanged directly between peers whenever possible. For any communication between clients, a P2P overlay topology is used. An interesting property of DVEs is that communication most likely occurs between peers that are close to each other in respect of the virtual geography. Reasons for this include direct avatar to avatar interaction, mutual visibility or an exchange of data with geographic reference. Consequently, by maintaining direct connections between "nearby" peers, complex routing mechanisms can be avoided. Besides supporting scalability, a positive effect of this is that no deterministic structure for routing needs to be maintained. This allows for high peer dynamics and avoids network maintenance overhead.

The P2P topology of the HyperVerse infrastructure is built in a way that maintains direct connections between peers with intersecting AoI. Such peers will most likely have a common interest in data or can potentially collaborate via an object present in both of their AoI and might thus be required to communicate. In order to make real-time distribution of data scalable, clients in the HyperVerse infrastructure first ask neighbored peers whenever data from within their AoI needs to be retrieved. Only if this fails, data distribution will fall back to a federated backbone of seed servers. Since it is beyond the scope of this paper, we refer to [4] for a detailed description of the data distribution and neighbor discovery mechanism.

Currently, the application of additional rules (as e.g. described in [22]) which effectuate power law properties in the resulting overlay are being investigated. The main motivation for using such schemes are the favorable global properties that can be attributed to the power law property of networks [10]. Furthermore, the theory of critical phenomena in complex networks [11] provides means for a self-organized monitoring and adaptation of these properties [23].

3.2 Epidemic Hot Spot Aggregation

In this section we will present an algorithm that utilizes the P2P overlay described above in order to identify hot spots within a definable range based on a client's local knowledge. The algorithm uses an epidemic aggregation mechanism similar to the one described in [15]. The main advantage of this scheme is that it does not require any network structure, making it suitable for networks with highly dynamic constituents and topologies. The basic idea of our epidemic hot spot aggregation approach is to consider objects and avatars as particles with a certain mass, their mass reflecting the transmission effort involved when retrieving associated model and texture data. In order to describe the algorithm in more detail, we rely on the following definitions.

Definition 1. *Let p_i be a client with n renderable objects and users within its AoI. With $s_1, \ldots, s_n$ denoting their transmission sizes in bytes, we define the mass M_i of the client's AoI as $M_i = \sum_{j=1}^{n} s_j$. With $r_1, \ldots, r_n$ being the objects' (or users') positions in virtual space, a peer p_i's center of mass C_i is defined as $C_i = \frac{\sum_{j=1}^{n} r_j \cdot s_j}{M_i}$.*

Our aggregation scheme involves peers computing cumulative mass and center of mass for objects and avatars in their AoI as well as exchanging this information with a random neighbor. For this, we assume that each peer p_i keeps a local fixed-size vector S_i of "most crowded spots". Each entry (C_k, M_k) in S_i consists of an AoI mass M_k and the associated center of mass C_k. Furthermore, each peer p_i defines a maximum lookahead distance L_i which determines the range for which it desires to aggregate the most densely populated hot spots.

The algorithm requires each client to be aware of its nearest neighbors (according to the overlay topology described above) along with their AoI and lookahead radii. Each peer p_i applying the epidemic scheme as described in [15], in random intervals the entry in S_i with maximum mass is exchanged with a random neighbor p_j in the overlay. The following algorithm describes the aggregation scheme when information is exchanged between two peers p_i and p_j:

1. p_i selects the entry (C_k, M_k) from its local vector S_i with $dist(C_k, p_j) < L_j$ and maximum mass M_k. If there is no such entry or the mass M' of p_i's current AoI (with center of mass C') is bigger than M_k, (C', M') will be sent to p_j, otherwise (C_k, M_k) will be sent.
2. On reception of (C, M), p_j will add it to the size-constraint vector S_j, possibly replacing an existing entry with smaller mass. It will then select the entry (C_k, M_k) from its vector S_j with maximum mass M_k and with $dist(C_k, p_i) < L_i$. If such an entry does not exist or if the mass M' of p_j's current AoI (with center of mass C') is bigger than M_k, the local information (C', M') will be sent back to p_i. Otherwise (C_k, M_k) will be sent.

The scheme resembles the decentralized gossip-based maximum aggregation as proposed in [16] but it has been extended by some additional rules. First of all information that is aggregated is dynamic, since a client's AoI as well as the objects and avatars within (and thus the total mass as well as the center of mass) can change at any time. In order to account for this dynamics, on each gossip iteration clients check whether the mass of their current AoI is bigger than their currently known maximum. Furthermore, range constraints have been introduced: By having a client check whether the maximum center of mass falls within the random neighbor's lookahead range, only information from within the client's lookahead will be aggregated.

The main advantage of using epidemic aggregation is that is has proven to produce fast-converging results in highly dynamic networks [16] with constant communication cost. In our case, each client will retrieve a fixed-size set of most crowded places that are within its lookahead radius and which are known to some peer in its connected component. For this, only a small number of information exchanges is required. Since only aggregated information are exchanged in regular intervals, this involves only a small constant communication overhead. In particular, in each gossip iteration a 3D position and a mass value needs to be exchanged with a single neighbor.

From a "thermodynamic" argumentation, one can infer that aggregate information on hot spots is likely to remain stable in spite of the dynamics of its constituents. Thus an important contribution of our algorithm is that it allows clients to efficiently retrieve stable, aggregate information on hot spot areas without being burdened with dynamic details on objects or users within.

3.3 Hoarding of Hot Spot Data

A peer p_i's vector S_i resulting from above epidemic aggregation scheme can be utilized to prevent load delays. Entries of S_i representing the most crowded areas within p_i's lookahead radius, their aggregate mass and center of mass provide an estimation of required bandwidth and loading time. The fact that users exhibit a high probability to go to these most crowded (and thus most popular) areas calls for a foresighted hoarding of data from within these regions. In order to prioritize the hoarding of data for multiple hot spots in S_i, additional client-side information like motion trajectories, speed and history can be used.

We propose that clients dedicate a certain fraction of idle downlink bandwidth for speculatively hoarding data from hot spots and thus mitigating delays. This fraction needs to be adapted to the client's lookahead radius: A larger lookahead radius provides for smaller hoarding bandwidth, since there is more time for prefetching data. It is important to note that, by virtue of the epidemic scheme described above, increasing the lookahead radius does not involve additional communication overhead. Further arguments on which lookahead radii and thus hoarding bandwidths are suitable to be used in practice are currently under investigation.

4 Evaluation Results

In order to implement and evaluate the proposed hoarding scheme, comprehensive simulation support is required. For this, TopGen[3] [24] - an open-source topology generator created within our working group - has been extended by DVE simulation facilities. For realistically simulating large-scale distributed virtual environments it provides a deterministic, event-based simulation of avatar mobility based on different models. For the following simulations, a DVE-specific model has been used which will be described in more detail in the following section. Apart from users with simulated mobility, objects with different transmission sizes can be placed in the virtual world. By this means it is possible to realistically simulate the object retrieval traffic resulting from different mobility patterns, IM and prefetching schemes.

An important feature of TopGen is the possibility of extending it by so-called experimental modules. By hooking into certain simulation events (join and exit of clients, avatars becoming mutually visible, refresh of AoI, etc.), simulations can be extended with own code. For the evaluations in this section we have created an experimental module which implements interest management, object retrieval and epidemic hoarding as described in sections 2 and 3. In each simulation step, every peer was allowed to transfer a limited amount of data in order to simulate client-side bandwidth limitations.

[3] http://syssoft.uni-trier.de/˜scholtes/

4.1 Preferential Way Point Mobility Model

An important aspect when performing DVE simulations is the choice of a realistic model for avatar mobility. The most simple model known e.g. from the field of ad hoc networks is random way point [19]. Here each simulated agent selects a random point with uniform probability. The agent then starts moving towards this point and selects a new target once having reached it. Due to lacking a notion of Points of Interest (PoIs), the model is not very realistic. Since it seems intuitive that users move to certain PoIs with higher probability, we propose a variation of the random way point model that respects objects, users and their distribution. For this we assume that DVE users are collaborative by nature, i.e. that they are primarily interested in interacting with objects and other users.

Our "preferential way point" mobility model is identical to the random way point model, except for the fact that movement targets are selected among all possible object and user positions with a probability proportional to the object and user density in the surrounding region. Users will thus preferentially move to densely populated hot spot regions. The name has been chosen in resemblance of the "preferential attachment" generation model for power law graphs [1] in which newly added nodes create links to existing nodes with probability proportional to the target's link number. While here every new edge will further increase the "attractiveness" of a node, a similar effect occurs in our mobility model: Each user that moves towards a certain position increases the probability that other users go there and thus further the attractiveness of the surrounding region.

4.2 Simulation Results

In this section we provide simulation results of the hoarding scheme that have been retrieved using the TopGen simulation environment. For this a background set of 200 objects is randomly distributed in a virtual region of 1000 x 350 pixels in size. Additionally two hot spots consisting of 400 objects each are created. Synthetic data transmission sizes between 1 and 5 units are randomly assigned to objects and the users' avatars. The simulation comprises 200 randomly distributed avatars moving according to the preferential way point model. In each simulation step, each of the simulated clients can download a maximum of 15 data units from a fictional data provider. Furthermore each client uses a cache limited in size to 3000 data units. As described in section 3.1, clients with intersecting AoI are interconnected via an edge in the overlay topology. The initial distribution of objects and peers in the simulated setting is shown in Figure 3.

Two simulations consisting of 1250 iterations have been performed. For these, the IM scheme described in section 2 is used with a FoV radius of $d = 50$ pixels, an AoI radius of $d + \Delta = 75$ and $\Lambda = 12$ pixels. The motion speed of peers is set to a maximum of one pixel per simulation step, i.e. the AoI of a client is refreshed at most every 12 simulation steps. The simulated traffic arising from object retrievals is recorded for each peer. In order to capture and evaluate situations with visible load delays, the amount of object data from within the FoV which could not be loaded in time is recorded for each peer in every simulation step.

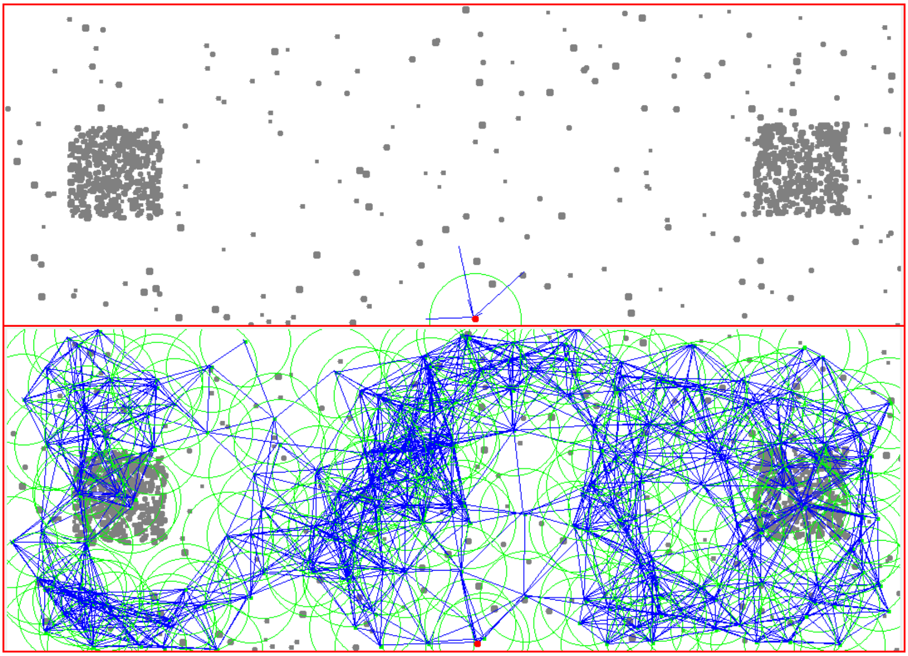


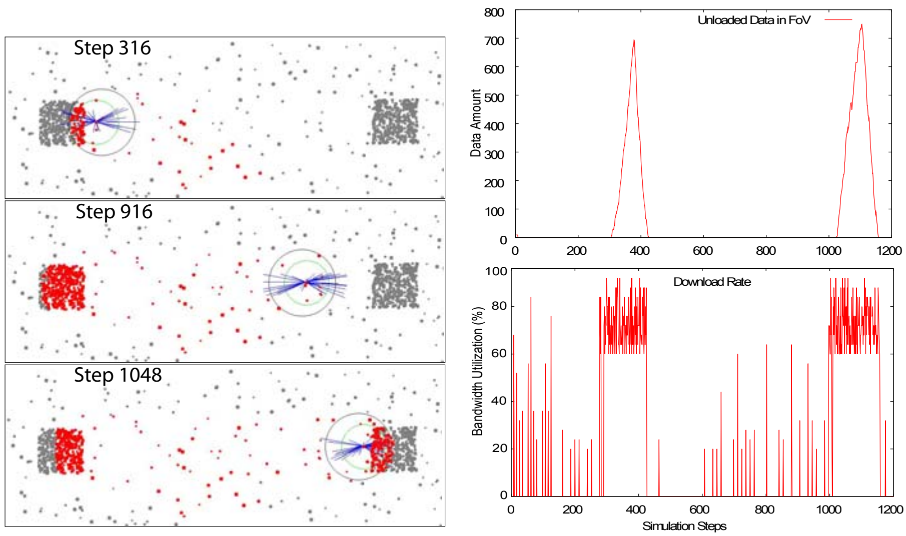
Fig. 3. Initial situation with all clients and P2P overlay (bottom) and selected peer only (top)

This value can be used to identify situations as motivated in section 2, i.e. where the client's bandwidth is not sufficient to retrieve data in time. The following paragraphs present results of a single random client. The initial position of this client can be seen in the top part of Figure 3.

Simulation without Epidemic Hoarding. Figure 4 shows simulation results of a random client with no epidemic hoarding being used. For the sake of clarity only the selected client is shown in Figure 4(a) at three selected simulation steps. Links to peers in the overlay topology (i.e. nearby clients with intersecting AoI) are indicated by edges. Based on the preferential way point mobility model, the selected client first moves towards the hot spot which can be seen in the left part of the simulated area in Figure 4(a). Having reached this in step 316 of 1250, it proceeds to the hot spot that can be seen in the right part of the simulated area. The client reaches this hot spot in step 1048. A video of the simulation can be found at the website of one of the authors[4].

Looking at the client's bandwidth that is utilized for retrieving data within its AoI, in the bottom part of Figure 4(b) one recognizes numerous peaks which occur whenever the AoI changes. Usually - because the AoI is bigger than the FoV - there remains enough time to load objects coming into the FoV shortly. In this case there are no objects with unloaded data in the client's FoV. In the

[4] http://syssoft.uni-trier.de/~scholtes/VideoA.avi



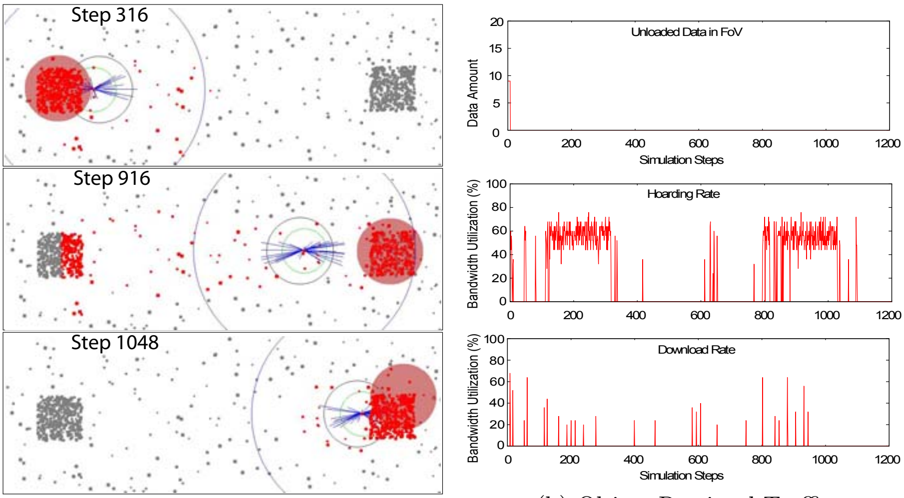
(a) Visualization of Selected Steps

(b) Object Retrieval Traffic and Pending Data Units

Fig. 4. Simulation Results without Epidemic Hoarding (Lines are drawn to guide the eye)

top part of Figure 4(b), one recognizes that this is the case during the first 315 simulation steps. In step 316, the client's FoV enters the crowded area. Due to the density of objects, now data can not be retrieved in time. The amount of unloaded data in the FoV that is shown in the upper part of Figure 4(b) exhibits two sharp peaks when the client enters the hot spot areas in simulation steps 316 and 1048. The maximum value of pending data was 750 units. Based on the maximum download bandwidth of 15 units per simulation step, in this situation at least 50 additional time steps would have been required to load all objects within the FoV in time. In a real setting this would have resulted in visible delays. Another effect than can be seen in Figure 4(b) is that beginning in simulation step 280, the client's available download bandwidth is saturated until it leaves the hot spot in simulation step 410. In reality this can affect other communication protocols (voice chat, avatar interaction, etc.) and thus degrade user experience over a long period.

Simulation with Epidemic Hoarding. Figure 5 shows the results of another simulation run for the same client using the same random seed. Here the epidemic hoarding algorithm described in section 3 is used. No client-side information (like e.g. movement trajectory) is utilized to improve the prediction which data shall be preloaded, i.e. all data from within predicted hot spot areas are hoarded. Pending data transfers resulting from the IM scheme are prioritized, i.e. only



(a) Visualization of Selected Steps

(b) Object Retrieval Traffic

Fig. 5. Simulation Results with Epidemic Hoarding (Lines are drawn to guide the eye)

otherwise unused bandwidth is used for hoarding. Clients use a lookahead range of 5 times their AoI size. The maximum bandwidth utilization for hoarding is set to 70 %. In case of a full client cache, an additional rule disables the hoarding of objects that are farther away than all objects in the cache. This prohibits thrashing situations in which hoarding displaces entries in the cache that are repeatedly required for rendering. Again, a video of this simulation can be retrieved from the website of one of the authors[5].

Figure 5 shows results obtained in this simulation. In Figure 5(a), the same selected simulation steps as in 4(a) are visualized. The lookahead radius is indicated by the outermost circle around the selected client. The client's local hot spot prediction resulting from epidemic aggregation is visualized by the shaded circle. Due to hoarding, all data have already been retrieved when the client enters the left hot spot in step 316. As soon as the right hot spot is in the client's lookahead range in step 916, the hot spot is correctly identified by the epidemic aggregation algorithm and hoarding of data begins. Finally, when the client's FoV enters the hot spot region in step 1048, again all data have already been retrieved and no load delays occur.

These claims are substantiated by the results shown in Figure 5(b). The middle diagram shows the rate at which data have been speculatively hoarded. Comparing both diagrams to Figure 4(b), one recognizes that periods with saturated bandwidth could be avoided. A positive side-effect of this is that it leaves more resources for other communication protocols. The most interesting result, the amount of unloaded data in the client's FoV, is shown in the topmost diagram.

[5] http://syssoft.uni-trier.de/˜scholtes/VideoB.avi

	Data Transferred	Unloaded Data in FoV	Canceled Transfer Data
without hoarding	5794	85109	94
with hoarding	6462	9	0

Fig. 6. Integral Transfer Statistics

Due to epidemic hoarding, load delays could be avoided since at no moment there are objects with unloaded data in the client's FoV[6].

Although these results look promising, an important question is the overall additional cost resulting from data hoarding. In order to assess this, the integral amount of data retrieved in both simulation has been recorded for the client considered above. The result is shown in Figure 6. One recognizes a roughly 11% increase in the total amount of data that have been transferred. An important question is how much of this difference results from situations in which a client not using epidemic hoarding leaves a hot spot before all objects have been loaded. This results in any queued downloads being canceled, thus underestimating the amount of total data that actually need to be transferred. A further evaluation of simulation data has shown that this accounts for a difference of 94 data units. After accounting for canceled transfers, one concludes that hoarding leads to a 9 % overhead in transferred data for the simulated setting.

5 Related Work

An important characteristic of DVEs is the locality of access with respect to virtual geography that results from human cognition. The main motivation for the geography-based overlay topology presented in section 3.1 is to respect this fact and thus guarantee that required data are available in a peer's neighborhood. Similar techniques have been investigated to improve search performance in general P2P systems [21] [2] [9]. The basic idea of these approaches is to adapt overlay topologies with respect to content that peers provide or are interested in.

Several P2P-based approaches to improve the scalability of large-scale DVEs have been proposed. Some of these approaches specifically address non-uniform access patterns. The P2P framework ATLAS [18] introduces a user-specific object popularity based on repeated access which is used to improve prefetching and caching. While this can mitigate delays for frequently visited areas, it does not work for hot spots that have not yet been accessed or that form dynamically. Another way of minimizing load delays is to predict avatar behavior. The utilization of local knowledge (e.g. motion trajectory) to predict future object access has been investigated in [17] and [7]. Although the proposed schemes do not consider heterogeneous distributions of dynamic objects, they might be suitable to prioritize hot spots predicted by our epidemic aggregation approach.

Most solutions to problems related to hot spots - or flash crowds - have been developed with a focus on the data provider side. Dynamic partitioning schemes

[6] The reason for the small peak in the first simulation step is, that in the initial situation the client was set to a position where some objects were in its FoV already.

like the one presented in [6] can e.g. be used to equally distribute objects or avatars from densely populated regions across data providers. The approach presented in [26] combines octrees with the Chord [25] protocol to achieve the same task. In order to maintain a minimum quality of service for at least a limited number of users in server-based MMOGs, [8] propose an "early warning system" that detects hot spots by monitoring performance degradations. In case a hot spot is identified, no more users are admitted to the identified crowded regions. None of the aforementioned approaches addresses the problem of the limited bandwidth of clients resulting in load delays regardless of the data provider's resources. The work that has been done in this area focuses on traffic resulting from mutual avatar visibility. [3] deals with avatar interactions in crowded DVEs and introduces a scheme that aggregates individual avatars to crowds in order to maintain scalability. [12] describes a group-based filtering mechanism that minimizes motion updates of nearby avatars in crowded areas.

Another active field of research are scalable interest management schemes for large-scale DVEs. A survey of such schemes for MMOGs has been performed in [5]. A common practice for scalable IM schemes is to reduce the AoI size when object or avatar density reach a certain threshold. This technique is e.g. used in the Voronoi-based clustering that can be found in VON [14]. Without the usage of AoI-independent hot spot detection mechanisms like the one presented in this article, reducing the AoI will however worsen the problem of delays resulting from object retrieval traffic. The usage of adjustable AoI shapes is another direction of research, being investigated e.g. for the P2P DVE infrastructure FLoD [13].

6 Conclusion and Future Work

In this article we have presented an efficient probabilistic and localized approach to identifying hot spot regions in DVEs that rely on real-time data distribution. It can be used to enrich interest management with hot spot predictions, so data can be speculatively hoarded before clients are too close for data still being retrieved in time. The only knowledge required for this prediction is a list of objects and avatars in the client's AoI. By virtue of using an epidemic aggregation approach, the communication overhead involved is limited by a small constant. Another advantage is the fact, that neither user mobility nor peer dynamics constitute a problem, since epidemic aggregation makes no assumptions about the structure of the overlay topology. In DVEs, hot spot regions most likely remain at a certain position even though the constituting objects and avatars can be highly dynamic. The approach presented in this article is well-suited to efficiently retrieve inertial aggregate hot spot information without burdening clients with the details of the underlying dynamics. A disadvantage of existing prediction-based prefetching schemes is the fact that they do not consider dynamic hot spots. Rather than requiring an exact prediction of user mobility, our approach efficiently yields those regions that will potentially incur load delays. Prioritizing hot spots retrieved by means of prediction is considerably easier than obtaining an exact prediction of the user's movement.

From our simulations we draw the conclusion that the proposed scheme can efficiently be used to avoid load delays in DVEs with real-time data transfer and thus improve user experience. In the simulated setting the hoarding scheme resulted in a 9 % increase of a client's total data traffic while totally eliminating load delays. It remains to investigate how local knowledge like movement trajectories, motion speed or favorite venues can be used to further improve predictions and thus minimize this overhead. Since avatar traces of real-world DVEs are slowly becoming available, we plan to investigate the usage of such heuristics as well as the viability of our mobility model.

Another future direction is the optimization of what regions are treated as hot spots by clients. For this, we plan to apply a second epidemic aggregation algorithm which efficiently determines the average mass density within the virtual world. By this, only those hot spots that significantly exceed average density can be advertised by gossiping, thus improving prediction quality. While the scheme currently only works for a continuous avatar motion, we plan to investigate extensions that consider avatar relocation through teleportation. Finally, another direction for improvements is the consideration of additional information while aggregating information: When moving in a certain direction, e.g. gossip messages received from peers in this direction can be prioritized.

Ultimately, it has to be investigated whether interest management as a whole can be replaced in favor of a self-organizing and probabilistic prediction similar to the hot spot identification presented in this article. For this a fixed fraction of bandwidth could be dedicated to hoarding while a self-organizing process ensures that all required data are available in time.

References

1. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
2. Bawa, M., Manku, G.S., Raghavan, P.: Sets: search enhanced by topic segmentation. In: SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, USA, pp. 306–313. ACM, New York (2003)
3. Benford, S., Greenhalgh, C., Lloyd, D.: Crowded collaborative virtual environments. In: CHI 1997: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, USA, pp. 59–66. ACM, New York (1997)
4. Botev, J., Esch, M., Hoehfeld, A., Schloss, H., Scholtes, I.: The hyperverse - concepts for a federated and torrent based "3d web". In: Proceedings of the First International Workshop on Massively Multiuser Virtual Environments at IEEE Virtual Reality 2008 (March 2008)
5. Boulanger, J.-S., Kienzle, J., Verbrugge, C.: Comparing interest management algorithms for massively multiplayer games. In: NetGames 2006: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, 6 p. ACM, New York (2006)
6. Bouras, C., Giannaka, E., Tsiatsos, T.: Partitioning of distributed virtual environments based on objects' attributes. In: DS-RT 2007: Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications, Washington, DC, USA, pp. 72–75. IEEE Computer Society, Los Alamitos (2007)

7. Chan, A., Lau, R.W.H., Ng, B.: Motion prediction for caching and prefetching in mouse-driven dve navigation. ACM Trans. Interet Technol. 5(1), 70–91 (2005)
8. Chen, X., Heidemann, J.: Flash crowd mitigation via adaptive admission control based on application-level observations. ACM Trans. Interet Technol. 5(3), 532–569 (2005)
9. Chiou, H., Su, A., Yang, S.: Interest-based peer selection in p2p network. In: International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 549–554 (2008)
10. Dorogovtsev, S., Mendes, J.: Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press, Oxford (2003)
11. Dorogovtsev, S.N., Goltsev, A.V., Mendes, J.F.F.: Critical phenomena in complex networks (2007)
12. Han, S., Lim, M., Lee, D.: Scalable interest management using interest group based filtering for large networked virtual environments. In: VRST 2000: Proceedings of the ACM symposium on Virtual reality software and technology, pp. 103–108. ACM, New York (2000)
13. Hu, S.-Y.: A case for 3d streaming on peer-to-peer networks. In: Web3D 2006: Proceedings of the eleventh international conference on 3D web technology, New York, USA, pp. 57–63. ACM, New York (2006)
14. Hu, S.-Y., Chen, J.-F., Chen, T.-H.: Von: a scalable peer-to-peer network for virtual environments. IEEE Network 20, 22–31 (2004)
15. Jelasity, M., Montresor, A.: Epidemic-style proactive aggregationin large overlay networks. In: Proceedings of the 24th International Conference on Distributed ComputingSystems (ICDCS 2004), Tokyo, Japan, pp. 102–109. IEEE Computer Society, Los Alamitos (2004)
16. Jelasity, M., Montresor, A., Babaoglu, O.: Gossip-based aggregation in large dynamic networks. ACM Trans. Comput. Syst. 23(3), 219–252 (2005)
17. Lau, R.W.H., Chim, J.H.P., Green, M., Leong, H.V., Si, A.: Object caching and prefetching in distributed virtual walkthrough. Real-Time Syst. 21(1/2), 143–164 (2001)
18. Lee, D., Lim, M., Han, S.: Atlas: a scalable network framework for distributed virtual environments. In: CVE 2002: Proceedings of the 4th international conference on Collaborative virtual environments, pp. 47–54. ACM, New York (2002)
19. Lin, G., Noubir, G., Rajaraman, R.: Mobility models for ad hoc network simulation. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 1 (2004)
20. Morse, K.L., Bic, L., Dillencourt, M.: Interest management in large-scale virtual environments. Presence: Teleoper. Virtual Environ. 9(1), 52–68 (2000)
21. Ramanathan, M.K., Kalogeraki, V., Pruyne, J.: Finding good peers in peer-to-peer networks. In: IPDPS 2002: Proceedings of the 16th International Parallel and Distributed Processing Symposium, Washington, DC, USA, P. 158. IEEE Computer Society, Los Alamitos (2002)
22. Rozenfeld, A.F., Cohen, R.: Scale-free networks on lattices. Physical Review Letters 89, 218701 (2002)
23. Scholtes, I., Botev, J., Esch, M., Hoehfeld, A., Schloss, H.: Awareness-driven phase transitions in very large scale distributed systems. In: Proceedings of the Second IEEE International Conferences on Self-Adaptive and Self-Organizing Systems (SaSo). IEEE, Los Alamitos (2008)

24. Scholtes, I., Botev, J., Esch, M., Hoehfeld, A., Schloss, H., Zech, B.: Topgen - internet router-level topology generation based on technology constraints. In: Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools) (February 2008)
25. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: SIGCOMM 2001: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 149–160. ACM Press, New York (2001)
26. Tanin, E., Harwood, A., Samet, H.: Using a distributed quadtree index in peer-to-peer networks. VLDB 16(2), 165–178 (2007)