Scalability Analysis of the Hierarchical Architecture for Distributed Virtual Environments

Michael Kwok Johnny W. Wong

Presentation by Alexander Pokluda

Cheriton School of Computer Science, University of Waterloo, Canada

IEEE Transactions on Parallel and Distributed Systems 2008

Outline

What is a Distributed Virtual Environment?

Definition

A Distributed Virtual Environment is a shared virtual environment where users at their workstations interact with each other over a network

Design and Performance of a Virtual Environment Infrastructure

- In terms of scalability, a promising system architecture is a **two-level hierarchical architecture**
- Although the two-level hierarchical architecture is believed to have good properties with respect to scalability, not much is known about its performance characteristics

Contributions

Queueing theory is used to develop a performance model for the two-level architecture and obtain analytic results on the workload experienced by each server

The authors also investigate the issue of consistency and develop a novel technique to achieve weak consistency among the copies of the virtual environments at the various servers

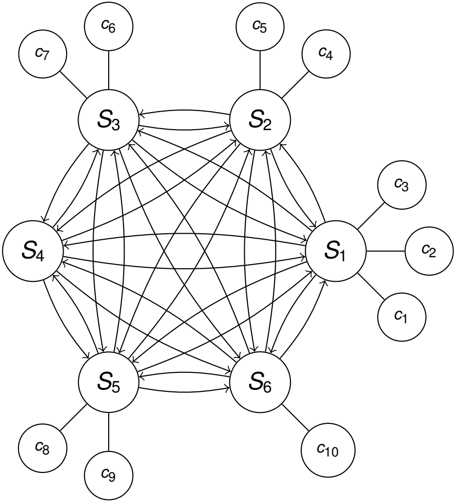
Two-Level Hierarchical Architecture

At the **lower level**

- users assigned to servers based on load-balancing consideration

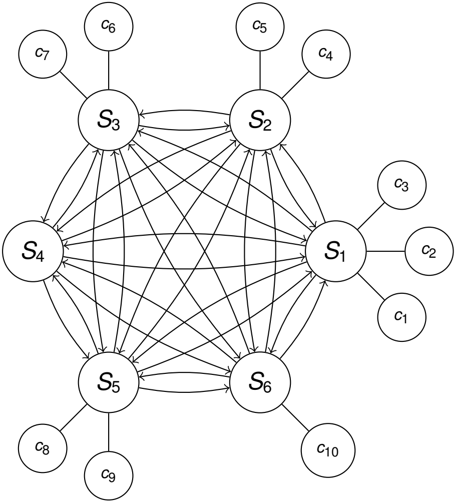
At the **higher level**

- servers communicate among themselves to ensure that updates are sent to affected users and that their VEs are as consistent as possible



Update Message Flow

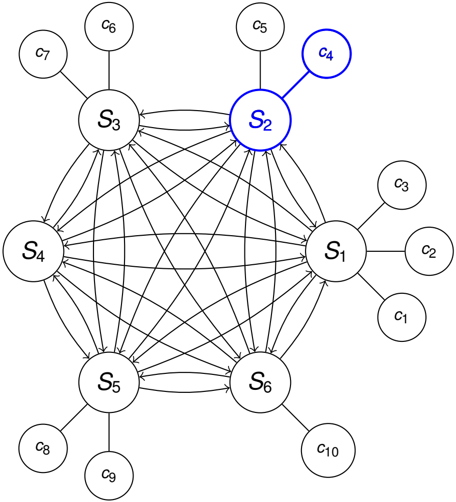
Suppose user c_4 is assigned to server S_2 and moves his/her avatar to user to a new location



Update Message Flow

Suppose user c_4 is assigned to server S_2 and moves his/her avatar to user to a new location

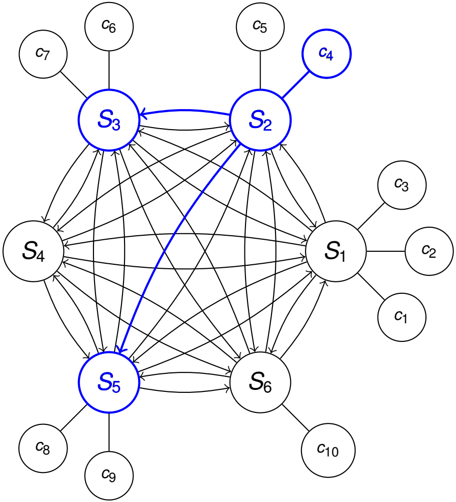
1. update packet is sent to local server



Update Message Flow

Suppose user c_4 is assigned to server S_2 and moves his/her avatar to user to a new location

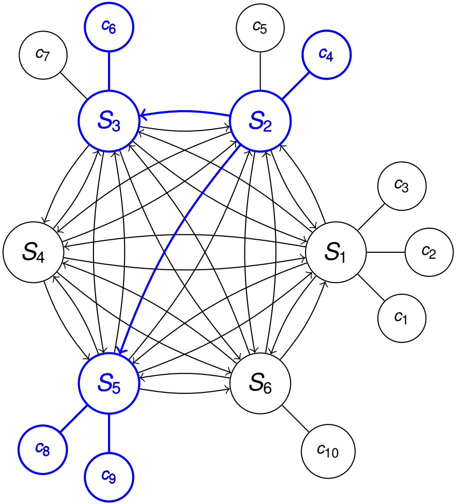
1. update packet is sent to local server
2. syn packet is sent to remote servers



Update Message Flow

Suppose user c_4 is assigned to server S_2 and moves his/her avatar to user to a new location

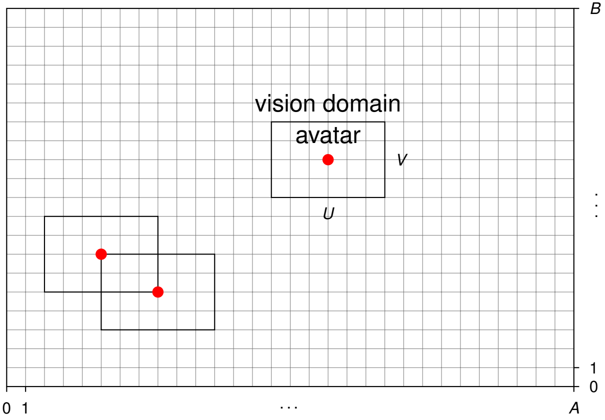
1. update packet is sent to local server
2. syn packet is sent to remote servers
3. update packet is sent to remote users



Avatars and Vision Domains

Our VE is modelled as a 2D unit square grid

- Avatars can only be located at a grid intersection (x, y)
- Each avatar is at the centre of its vision domain



User Movement

When the user makes a move, she can move up, down left or right according to a **probability distribution**

Assumptions

- Movement of each user is modelled by a Markovian chain
- Probability distribution is the same for all users
- Time until a user makes their next move is exponentially distributed and user moves are mutually independent

Let $q_{a,b;c,d}$ be the probability that a user moves from (a, b) to (c, d) in one step. It follows from the above assumptions that

$$P_{a,b} = \sum_{c=0}^{A} \sum_{d=0}^{B} p_{c,d} q_{c,d;a,b} \quad \text{for } a = 0, 1, ..., A;\ b = 0, 1, ..., B$$

where $\sum_{a=0}^{A} \sum_{b=0}^{B} p_{a,b} = 1$

Total Arrival Rate of Update and Syn Packets to a Server

- Let N_i be the number of logged on users at S_i, $i = 1, 2, ..., K$
- Let γ_i be the arrival rate of update packets to S_i
- Let $\eta_{k,i}$ be the arrival rate of syn packets from S_k to S_i, $k \neq i$

Arrival Rate of Update Packets

Let ϕ be the rate at which a user makes a move. The combined arrival rate of update packets to S_i is given by

$$\gamma_i = N_i\phi.$$

Arrival Rate of Syn Packets

1. Consider a *tagged* user at S_k
2. Let the probability that after the tagged user has made a move there are one or more users logged on to S_i who are within the tagged user's vision domain be $g_{k,i}$
3. Let $\xi_{k,i}(n)$ be the probability that exactly n users at S_i are within the tagged user's vision domain

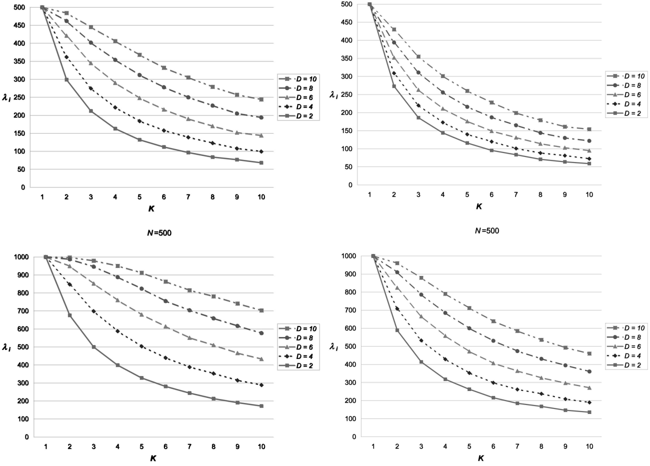
$$\xi_{k,i}(n) = \sum_{a=0}^{A} \sum_{b=0}^{B} \left[\binom{N_i}{n} (h(a,b))^n (1 - h(a,b))^{N_i - n} \right] p_{a,b}$$

where $h(a,b) = \sum_{x=x'}^{x^*} \sum_{y=y'}^{y^*} p_{x,y}$

4. Then $g_{k,i} = 1 - \xi_{k,i}(0)$ and $\eta_{k,i} = g_{k,i} N_k \phi$ for N_k users at S_k
5. Summing over all other servers, $\eta_i = \sum_{k=1,k\neq i}^{K} \eta_{k,i}$

The total arrival rate to S_i is the sum of γ_i and η_i, $\lambda_i = \gamma_i + \eta_i$.

Total Arrival Rate Results



N=500 N=500

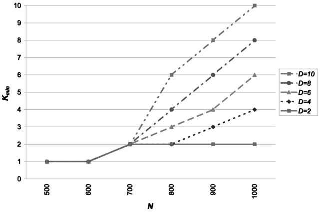
N=1000 N=1000

λ_i, total arrival rate at S_i, for a VE with size 100×100 and 150×150 and various D, where D is the width and height of the vision domain

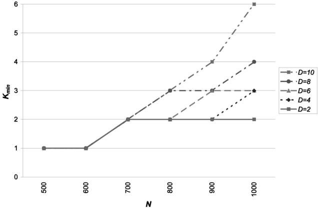
Total Arrival Rate Discussion

- In all cases, we observe a reduction in the total arrival rate λ_i when more servers are used
- A larger vision domain leads to a higher total arrival rate
- The fact that λ_i is a decreasing function of K indicates that the two-level architecture has good properties with respect to scalability

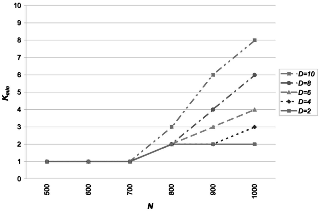
Scalability



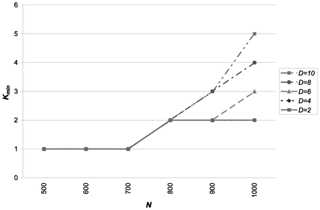
y = 0.8



y = 0.8



y = 0.9



y = 0.9

K_{min}, minimum number of servers required to support N users while $\lambda_i/\mu_i \leq y$ for VE with size 100×100 and 150×150

Scalability Discussion

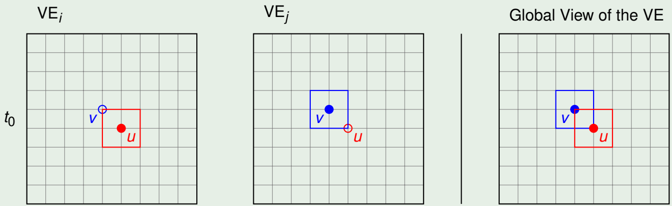
- K_{min} increases almost linearly with N
  - This is a good property with respect to scalability
- Rate of increase of K_{min} is affective by the size of the vision domain D
  - A large D has a negative impact on scalability
- A larger VE means a lower density of avatars and smaller rate of syn packets generated

Inconsistency Scenario 1

Scenario 1: User u's vision domain contains users who should not be there

Example

- At t_0, users u and v are in each other's vision domain

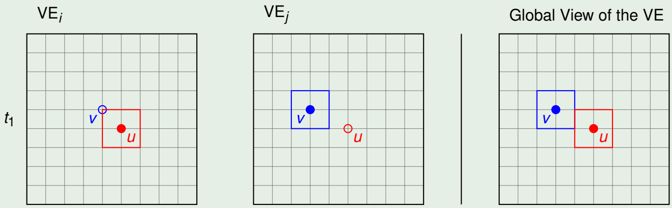


Inconsistency Scenario 1

Scenario 1: User *u*'s vision domain contains users who should not be there

Example

- At t_0, users *u* and *v* are in each other's vision domain
- At t_1, user *v* at S_j moves left by one step (no syn sent)

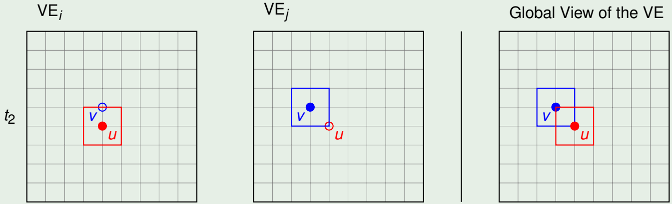


Inconsistency Scenario 1

Scenario 1: User u's vision domain contains users who should not be there

Example

- At t_0, users u and v are in each other's vision domain
- At t_1, user v at S_j moves left by one step (no syn sent)
- At t_2, user u at S_i moves left by one step (syn packet sent)

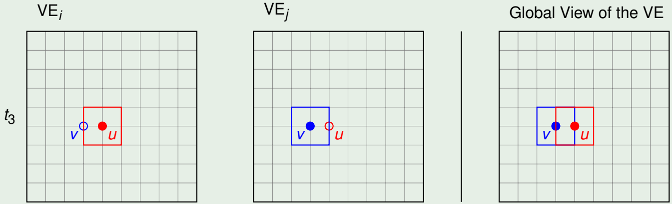


Inconsistency Scenario 1

Scenario 1: User u's vision domain contains users who should not be there

Example

- At t_0, users u and v are in each other's vision domain
- At t_1, user v at S_j moves left by one step (no syn sent)
- At t_2, user u at S_i moves left by one step (syn packet sent)
- At t_3, v moves down by one step (consistency restored)

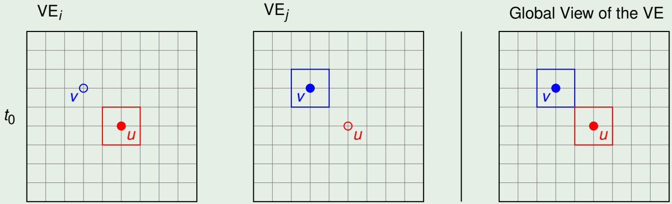


Inconsistency Scenario 2

Scenario 2: User *u*'s vision domain does not contain users who should be there

Example

- At t_0, both users in consistent state, not in each other's VD

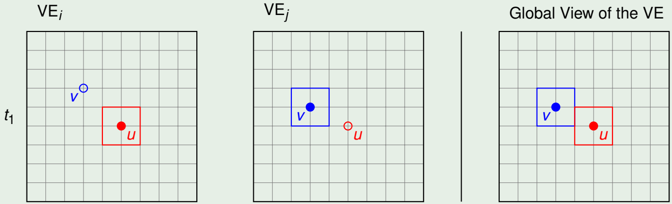


Inconsistency Scenario 2

Scenario 2: User u's vision domain does not contain users who should be there

Example

- At t_0, both users in consistent state, not in each other's VD
- At t_1, user v at S_j moves down by one step (no syn sent)

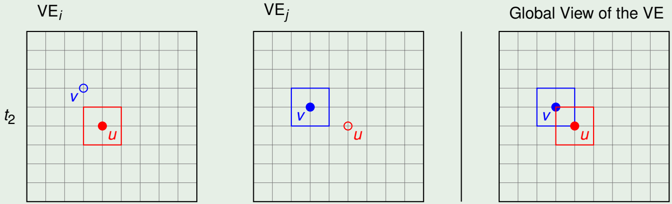


Inconsistency Scenario 2

Scenario 2: User u's vision domain does not contain users who should be there

Example

- At t_0, both users in consistent state, not in each other's VD
- At t_1, user v at S_j moves down by one step (no syn sent)
- At t_2, user u at S_i moves left by one step (no syn sent)

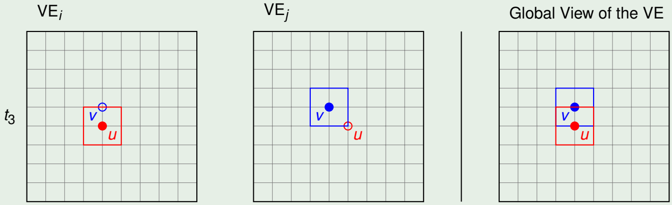


Inconsistency Scenario 2

Scenario 2: User u's vision domain does not contain users who should be there

Example

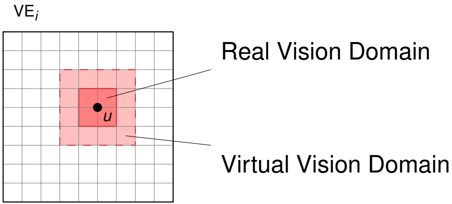
- At t_0, both users in consistent state, not in each other's VD
- At t_1, user v at S_j moves down by one step (no syn sent)
- At t_2, user u at S_i moves left by one step (no syn sent)
- At t_3, v moves right by one step (consistency restored)



Weak Consistency with the Virtual Vision Domain

A Virtual Vision Domain technique is introduced that achieves weak consistency The basic idea is to extend the vision domain to a larger size:

- The virtual vision domain is used to determine if a syn packet should be sent
- The real vision domain is used to determine a user's state



A virtual vision domain that is 2 units larger that the real vision domain prevents inconsistency in the above examples!

Experiment Setup

A simulation study was done to answer the following questions:

1. What is the amount of inconsistency if we rely only on future user movements to restore consistency?
2. If the virtual vision domain technique is used, how much improvement in consistency can be achieved, and what are the additional resources required?

Metrics to measure inconsistency:

- *f* fraction of users in the inconsistent state
- *t* mean time that a user spends in the inconsistent state
- *m* mean number of avatars that are incorrectly seen by a user

Experiments

- G1 relies only on future user movements to restore consistency
- G2 uses a virtual vision domain with size = $D + 2$
- G3 uses a virtual vision domain with size = $D + 4$

Experiment Data

$E = 100$ and $N = 800$

D	Strategy	f	K_{min}	t	m
2	G1	0.1794	2	2.0922	1.1551
	G2	0.0308	2	1.2919	1.0365
	G3	0.0025	2	1.0091	1.0128
4	G1	0.1536	2	1.8253	1.1320
	G2	0.0165	2	1.1939	1.0306
	G3	0.0010	2	1.0000	1.0110
6	G1	0.0809	2	1.5396	1.0935
	G2	0.0061	2	1.0945	1.0246
	G3	0.0031	3	1.2049	1.0281

$E = 100$ and $N = 1000$

D	Strategy	f	K_{min}	t	m
2	G1	0.1930	2	1.9367	1.1603
	G2	0.0669	3	1.4726	1.0728
	G3	0.0166	4	1.3143	1.0414
4	G1	0.2852	3	2.7986	1.2653
	G2	0.0817	4	1.6136	1.1184
	G3	0.0341	6	1.7132	1.0826
6	G1	0.2964	4	2.8236	1.3431
	G2	0.1219	6	1.9330	1.2017
	G3	0.0540	8	1.9425	1.1246

Experiment Data Continued

$E = 150$ and $N = 800$

D	Strategy	f	K_{min}	t	m
2	G1	0.1592	2	2.5167	1.1248
	G2	0.0420	2	1.7757	1.0345
	G3	0.0087	2	1.4837	1.0107
4	G1	0.1390	2	2.4896	1.1219
	G2	0.0384	2	1.7478	1.0370
	G3	0.0062	2	1.4887	1.0144
6	G1	0.1207	2	2.2335	1.1071
	G2	0.0264	2	1.5959	1.0327
	G3	0.0037	2	1.4302	1.0163

$E = 150$ and $N = 1000$

D	Strategy	f	K_{min}	t	m
2	G1	0.1346	2	2.3286	1.1339
	G2	0.0404	2	1.6246	1.0348
	G3	0.0199	3	1.6495	1.0240
4	G1	0.1646	2	2.3217	1.1264
	G2	0.0819	3	2.0434	1.0776
	G3	0.0343	4	2.0924	1.0473
6	G1	0.2573	3	2.6904	1.2213
	G2	0.1106	4	2.2693	1.1184
	G3	0.0443	5	2.7360	1.0675

Experiment Results

- When a virtual vision domain is used, f is significantly reduced
  - f is in the range of 15 to 30% under strategy G1
  - f is less than 9% for most cases under strategy G2
  - f is less than 2% for most cases under strategy G3
- A larger vision domain generally requires more servers. For example, $E = 150$, $N = 1000$ and $D = 4$, $K_{min} = 2, 3$, and 4 for G1, G2, and G3 respectively.
- A virtual environment that is more dense may require a larger number of servers
- The results do not show an obvious relationship between t and the different strategies
- m decreases as D increases
  - Furthermore, $m \to 1$ as $D \to E$

Summary

- We investigated the performance and scalability of a two-level architecture for DVE systems

Summary

- We investigated the performance and scalability of a two-level architecture for DVE systems
- We obtained analytic results for the total arrival rate of packets to each server and these results confirmed that the two-level architecture is scalable

Summary

- We investigated the performance and scalability of a two-level architecture for DVE systems
- We obtained analytic results for the total arrival rate of packets to each server and these results confirmed that the two-level architecture is scalable
- We investigated how inconsistencies may arise and be restored

Summary

- We investigated the performance and scalability of a two-level architecture for DVE systems
- We obtained analytic results for the total arrival rate of packets to each server and these results confirmed that the two-level architecture is scalable
- We investigated how inconsistencies may arise and be restored
- We proposed a new technique called Virtual Vision domain

Summary

- We investigated the performance and scalability of a two-level architecture for DVE systems
- We obtained analytic results for the total arrival rate of packets to each server and these results confirmed that the two-level architecture is scalable
- We investigated how inconsistencies may arise and be restored
- We proposed a new technique called Virtual Vision domain
- Simulation results showed that the virtual vision domain technique is effective in reducing inconsistency at the expense of a potential increase in the number of servers required

Discussion Questions

1. Are the standard assumptions that were used to enable the queueing theory analysis valid? For example, user movement was assumed to be modelled by a Markovian chain and the arrival rate of update packets was assumed to follow an exponential distribution.
2. The authors suggest that virtual vision domain technique, which achieves “weak consistency,” may be suitable for role playing games like World of Warcraft, or social sumilation games such as The Sims, but state that it may not be suitable for fast paced games, such as first person shooters. What other techniques could be used for this type of game?