Interactive 3D Services over Windows Azure

Jiri Danihelka
R&D Centre for Mobile Applications (RDC)
Czech Technical University in Prague (CTU)
jiri.danihelka@rdc.cz

Lukas Kencl
R&D Centre for Mobile Applications (RDC)
Czech Technical University in Prague (CTU)
lukas.kencl@rdc.cz

ABSTRACT

Windows Azure and other cloud platforms are ideal for building and deploying rapidly scalable 3D environments, usable for multi-player online games or interactive 3D education and e-commerce. Such 3D environments need to allow cooperation and interaction of multiple users in a single world. This may be difficult to synchronize due to network and cloud operation delays. We present the project on building 3D services in the cloud, an example implementation of such an application, a shared 3D Teapot in Windows Azure, and a measurements-based discussion of performance issues when the teapot is shared and actively manipulated by many simultaneous users.

Keywords

3D environments, cloud services, scalability, delay.

1. INTRODUCTION

Windows Azure and other cloud platforms are ideal for building and deploying rapidly scalable 3D environments, usable for example for multi-player online games or interactive 3D education and e-commerce. Such 3D environments need to allow cooperation and interaction of multiple users in a single world, interacting with each other in real-time and sharing and effecting changes in the same environment. This may be difficult to synchronize due to network and cloud-operation delays. Delays of 10s or 100s of milliseconds may severely hamper the ability of users to share the 3D environments effectively.

Further questions arise when implementing such 3D shared service in the cloud: how to most effectively utilize the resources of the Windows Azure platform when architecting such service, and how to distribute well the efforts of 3D rendering and action monitoring and synchronization between the cloud and the many individual clients with potentially different capabilities.

In this work we present a project on building interactive multi-user 3D services in the Windows Azure cloud. The ultimate goal is to build a prototype platform that would allow creating 3D virtual environments stored in the cloud and accessible by a variety of devices with very different performance and capabilities. Some of the devices (e.g. mobile phones) may not even be able to store the entire scene geometry in their memory and will have to use only partial content downloading. The cloud-side application will automatically and instantaneously adjust and communicate the evolving world content to reflect the actions and changes effected by the many users interacting with objects in the 3D environment.

We present a prototype of such an application, a shared 3D Teapot in Windows Azure, and a measure- ments-based discussion of performance issues when the teapot is shared and actively manipulated by many simultaneous users.

As part of the project work, we have also prototyped a 3D e-commerce platform with a talking-head interface. This platform will allow easy creation of a 3D virtual shop by 3rd parties. In both efforts we utilize brand new Microsoft technologies: the graphics rendering is based on Silverlight and the cloud service is implemented using Windows Azure.

The ultimate goal is to build a generic platform for easy deployment of mobile 3D interactive services based on Microsoft cloud technology. This platform will allow easy creation of 3D virtual shops and games. Shopkeeper-users will be able to upload models of 3D-goods and easily customize the 3D talking-head and environment appearance.

Cloud computing is a perfect platform for introducing more sophisticated services beyond the current WWW paradigm. This research focuses on the topic of dynamic sharing of 3D virtual-world content.. The created 3D worlds will allow cooperation of multiple users in a single world and we investigate methods for providing scalable 3D worlds management in the cloud. However, managing delay and concurrency are vital aspects for usability for highly interactive scenarios. Our practical, measurements-based analysis show the limits of the current cloud-based scenario.

2. RELATED WORK

Maggiorini and Ripamonti [7] showed that game providers can benefit from scalable allocation of resources in the cloud thanks to the reduced cost of ownership and shorter deployment time. Cloud computing allows to avoid overallocating of game server infrastructure that would have meant waste of resources, as well as its underallocating - implying loss of subscribers. They also proposed a tree-tier architecture for massive multiplayer online games in the cloud environment.

Iosup at al. [5] presented an architecture for analytic of cloud massively multilayer games and used it for analyzing popular game RuneScape. They recorded progress of 500,000 players for over a week.

Figure 1: Screenshot of a Client-server 3D talking head based on Windows Mobile 6.5 and OpenGL ES technology (without cloud support), previously developed at RDC [3, 4]

Najaran and Krasic [9] presented an architecture for host first-person-shooting games in a cloud to reduce aggregate bandwidth requirements of the game by using a scalable publish-subscribe subsystem. Players can set preferences for updates they receive, this allows to scale the game up to hundreds of players.

Windows Azure has also few drawbacks for gaming; especially it lacks support for unreliable services. Windows Azure currently does not support communications over UDP protocol that is often used in some real-time game for its low latency with the drawback of a decreased reliability on the communications. Azure roles do not allow the creation of UDP endpoints. Games that use unreliable communication must be modified to work on Azure. Many free-to-play games also get along with less reliable services of other providers (e.g. Amazon) that cost less.

Currently only a few games are based on Windows Azure. They are mostly social (multiplayer) turn-based games:

Tankster (www.tankster.net) is a prototype of turn-based shooting game that was written by Microsoft. It is based on Html 5 and also has a client for Windows Phone 7. This game serves as a sample how to use Windows Azure Toolkit for Social Games (https://github.com/WindowsAzure-Toolkits/wa-toolkit-games).

Vampire Legacy (http://apps.facebook.com/vampirelegacy) is a facebook turn-based RPG game written in Adobe Flash that uses servers in Windows Azure. According to its creators company Sneaky Games, new load-balanced instances can be up and running in less than 10 minutes.

Game company Kobojo developed a facebook social robot-fighting game RobotZ (http://apps.facebook.com/robotzarena/). According to their case study [6] Windows Azure provided them a reliable and scalable infrastructure with simplified maintenance.

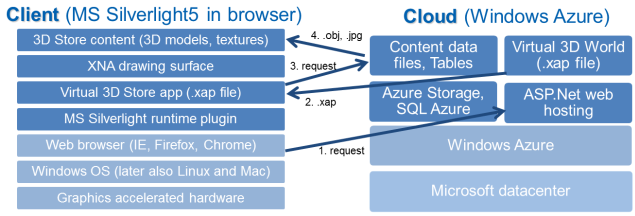

Figure 2: Client and Cloud Functionality distribution

Zombie Pandemic (http://pixelpandemic.net/zombie-pandemic/) is a turn-based MMORPG game based on Windows Azure that is being developed by Pixel Pandemix startup. It takes place after a zombie apocalypse where the player has to explore places and fight with monsters.

The 3D Mobile Internet project at RDC [10] has, over the past few years, created multiple prototypes of platform-independent client-server 3D applications for mobile phones (primary Windows Mobile 6, Android and Symbian – see Figure 2) and desktop computers (Microsoft Windows, Linux, Mac OS). However, only recently we have oriented ourselves towards cloud support of such platforms.

3. ARCHITECTURE

The application architecture consists of multiple layers, as described in figure 2.

Our goal is to create 3D environments that are stored in the cloud and can be accessed by a variety of devices with very different performance and capabilities. We seek to provide such virtual environments both within web browsers and on mobile phones. That is why we use Silverlight 5 technology, because it introduces the ability to use hardware-accelerated 3D graphics using XNA in Silverlight applications. This opens up a whole new set of scenarios for world visualization.

The shop content can be changed at runtime by downloading different model geometries from cloud. We have prepared several scenarios to demonstrate the variety of possibilities – a mobile-phone shop for operators, a plant shop and a furniture shop (see Fig. 4).

4. IMPLEMENTATION

We have focused on new Microsoft technologies. For the mobile client we use mobiles phones with Windows Phone 7 operating system. Rendering of the shop is based on a combination of Silverlight and XNA. The cloud service is implemented using the Windows Azure platform and is accessible via a desktop Windows PC or a device with a different operating system using a browser with Silverlight 5.

To reuse the rendering code between the web browser with Silverlight 5 plugin and Windows Phone 7, we have created a library that encapsulates the differences between those platforms. The most challenging issue of the library was with the graphics shaders. Windows Phone 7 forbids using a custom vertex and pixel shader, because it runs on a device with a graphics chip that does not support shaders and thus

Figure 3: 3D e-Shop with Talking Head on Windows Azure

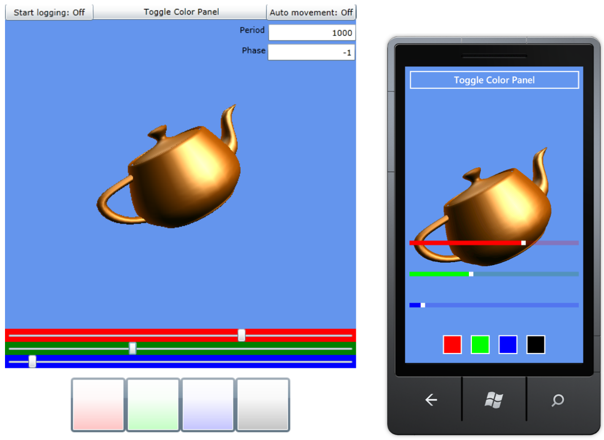

Figure 5: Screenshots of two client versions of the synchronized Windows-Azure-based 3D Teapot, accessible simultaneously by multiple clients: mobile (XNA-based) and desktop (Silverlight 5 in a browser).

only a set of pre-defined shaders can be used. On the other hand, Silverlight 5 3D does not have some high-level rendering functions to save downloading size and achieve higher portability. Those functions have to be implemented using low-level vertex and pixel shaders. Thus Silverlight 5 forces its developers to use shaders.

To overcome the shader problem we had to transfer the predefined Windows Phone 7 shaders to Siverlight 5 and implement Windows Phone 7 high-level rendering functions using low-level Silverlight 5 shaders. In our applications we have to use only those functions available on both platforms, thus we cannot use other shaders than those that are pre-defined. In March 2012 we plan to provide the XNA Interface library to other developers to ease them developing 3D Silverlight projects for multiple platforms.

Our first prototype of a 3D environment is the 3D Teapot application (see figure 4), a 3D object that can be manipulated (i.e. changing orientation or color) and observed in real-time by multiple users simultaneously, using different terminals. The teapot representation is being stored on Windows Azure. A Web role is used as frontend, while the actual 3D Teapot state is being permanently stored in Azure SQL and updated with every manipulation of the teapot.

5. PERFORMANCE MEASUREMENT

In one of our first experiments we let 24 automated client instances attempt to manipulate the 3D teapot in Windows Azure simultaneously (see Fig. 4, 5), to test scalability, reliability and response latency under a shared interaction scenario. The machines are connected at our university using a high-speed, 1 Mbps academic network. The simulation runs for about 1 hour, starting with only one instance of the teapot and incrementally adding other instances, always 2 per PC. We also keep decreasing the period of issuing a turning command to the 3D Teapot, starting with 1000 ms on the first machine and ending with 50 ms frequency on the last.

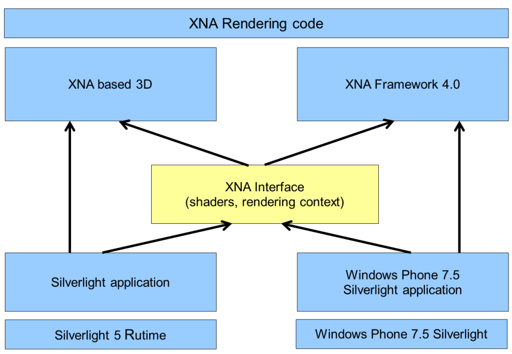

Figure 4: Code reuse, browser and phone versions

It is clearly visible that already with about 6 instances running, the individual 3D Teapots are not properly synchronized, while with 24, there is no trace of synchronization at all.

Measurements of latency of responses from Windows Azure

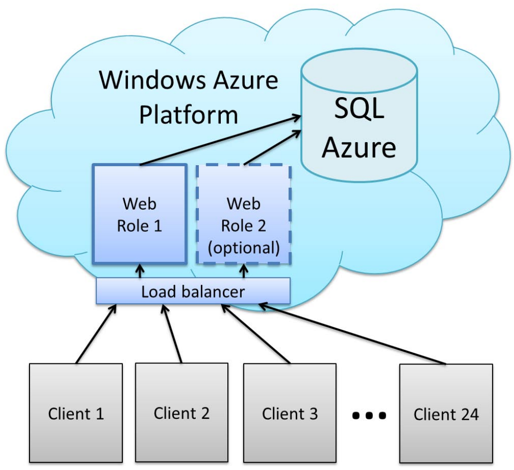

Figure 6: Architecture of the experiment: Up to 24 client instances were simultaneously trying to connect to the web service to synchronize state of a single object in the database every 100 ms.

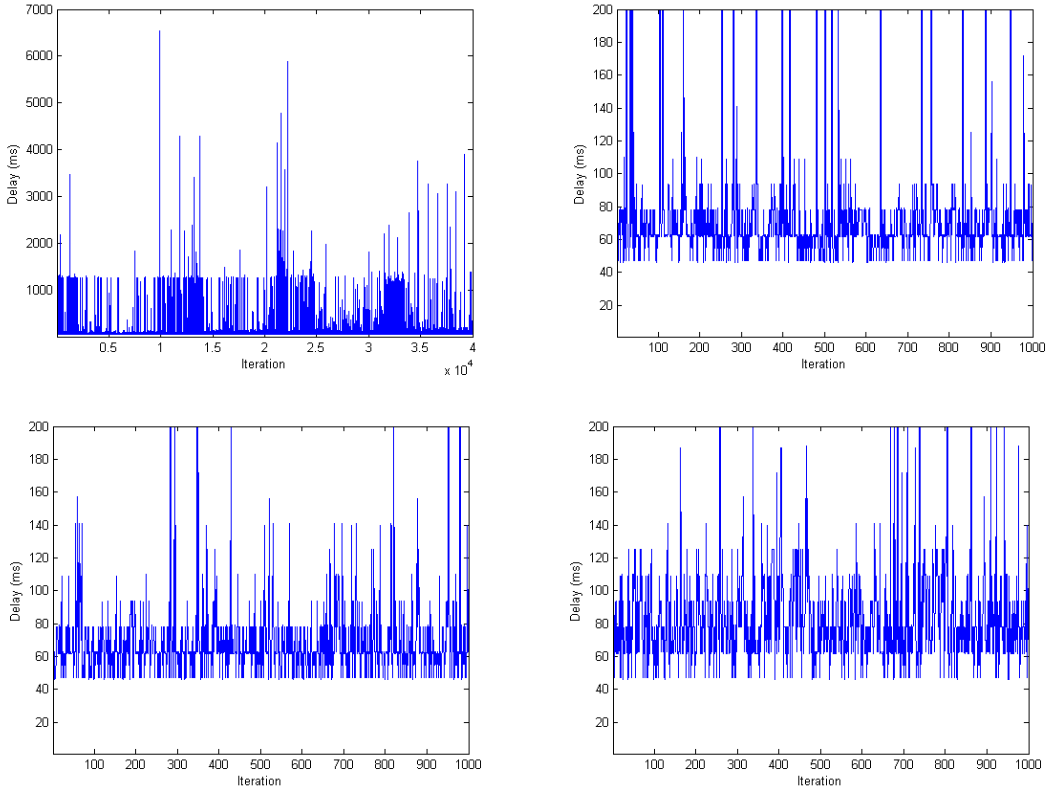

Figure 7: 3D Teapot in windows Azure, response latency in milliseconds, the entire simulation (top left), requests 1000-2000 (top right) 20000-21000 (bottom left) and 40000-41000 (bottom right).

is presented in figure 5. Clearly, there is quite some unpredictability, with most responses around 70 ms, some around 1200 ms (packet retransmits?) and some outliers above 5000 ms. Clearly with the growing number of teapot instances, the latency becomes less predictable and higher, but still within a manageable interval of below approximately 120 ms. We have only used a single instance on the side of the cloud – it is likely that scaling there would yield significantly better results.

Per the entire experiment the mean latency is 91.40 ms with standard deviation 165.86, minimum latency of 46 ms and maximum of 8078 ms.

Another experiment we have carried out is to run the teapot shared between two instances on a single machine, who both tried to manipulate periodically on the order of 500ms, over laptop WiFi in a busy, centrally located *Starbucks Cafe* (see Fig. 5. Here, the network effects are clearly much stronger than on the university high-speed connection network. The mean delay value is 204.38 ms, with standard deviation of 100.953.

6. CONCLUSION AND FUTURE WORK

The early measured results show some question marks as to whether Windows Azure can provide enough reliability and short-enough latency to support true multi-user interaction in 3D environments in the cloud. In particular the relatively high variability in the latency (standard deviation) may be a factor; perhaps the reliance on TCP may be hurtful in this respect. Much more experiments, analysis and measurements are needed though, which we intend to carry out in the very near future.

Proper understanding of distributed 3D worlds in the cloud will allow adding new kind of services hosted on the Windows Azure and Windows Phone platforms. An open plat-

Figure 8: Tests of multiple parallel clients simultaneously attempting to manipulate an identical 3D Teapot stored in Windows Azure cloud.

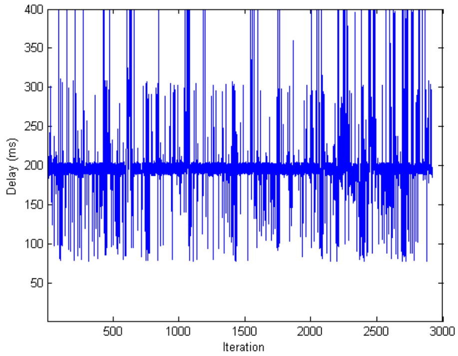

Figure 9: 3D Teapot in Windows Azure, response latency in milliseconds, when used for sharing and active manipulation of two users in a city cafe.

form for interactive 3D graphics in the cloud will result in strong social and commercial impact worldwide – new assistive services may be deployed or users may enjoy the experience of 3D internet shopping or other interactive services [10]. Such distributed 3D environments may improve productivity – e.g. by reducing travel – and have positive impact on the environment by the same token as well as by optimizing its own energy consumption.

As a major novelty, we investigate and prototype algorithms, methods and techniques for providing scalable, state-of-the-art multi-user interactive 3D services in the mobile cloud. Furthermore, the goal is to provide an open platform for other researchers to build on and a tool for easy deployment of such services, having potential impact in many interactive domains (e-commerce, assistive technologies, real-time interaction, gaming, . . .) and improving productivity or even contributing towards the environment. It will foster rapid growth of visually attractive 3D applications appearing in the Windows Phone 7 Marketplace.

A further research topic we would like to focus on is dynamic component-code migration between client/phone and server/cloud at runtime. This concept is beginning to attract attention among computer science researchers – first prototype approaches have been described by Cuervo et al [2] or Chun and Maniatis [1]. In the future we would also like to add support for Xbox360 or its successor.

Ultimately the outcome of the project will be made available as part of the Microsoft Research Hawaii Service Store [8].

Acknowledgment

This project is partially sponsored by Microsoft Czech Republic and the CTU FEE Interoperability Lab. Many thanks also belong to researchers in the MSR Redmond Connections and Networking Research Group, especially Dr. Judith Bishop, Dr. Victor Bahl and Dr. Stefan Saroiu, for initial project consultations.

7. REFERENCES

[1] B. Chun and P. Maniatis. Augmented smartphone applications through clone cloud execution. In *Proceedings of the 12th conference on Hot topics in operating systems*, pages 8–8. USENIX Association, 2009.

[2] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: making smartphones last longer with code offload. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 49–62. ACM, 2010.

[3] J. Danihelka, R. Hak, L. Kencl, and J. Zara. 3d talking-head interface to voice-interactive services on mobile phones. *International Journal of Mobile Human Computer Interaction IJMHCI*, 3(2), 2010.

[4] J. Danihelka, L. Kencl, and J. Zara. Reduction of animated models for embedded devices. In *International Conference on Computer Graphics, Visualization and Computer Vision*, 2010.

[5] A. Iosup, A. Lăscăteu, and N. Ţăpuş. Cameo: enabling social networks for massively multiplayer online games through continuous analytics and cloud computing. In *Proceedings of the 9th Annual Workshop on Network and Systems Support for Games*, page 7. IEEE Press, 2010.

[6] Kobojo. Game developer meets demand and prepares for success with scalable cloud solution. http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?CaseStudyID=4000008216.

[7] D. Maggiorini and L. Ripamonti. Cloud computing to support the evolution of massive multiplayer online games. *ENTERprise Information Systems*, pages 101–110, 2011.

[8] Microsoft Research Redmond. Mobile cloud project hawaii. http://research.microsoft.com/en-us/um/redmond/projects/hawaii/.

[9] M. Najaran and C. Krasic. Scaling online games with adaptive interest management in the cloud. In *Proceedings of the 9th Annual Workshop on Network and Systems Support for Games*, page 9. IEEE Press, 2010.

[10] RDC. 3d mobile internet project. http://www.rdc.cz/en/projects/3DMobileInternet/.