GamingAnywhere: An Open-Source Cloud Gaming Testbed

Chun-Ying Huang[1], De-Yu Chen[2], Cheng-Hsin Hsu[3], and Kuan-Ta Chen[2]

[1]Department of Computer Science and Engineering, National Taiwan Ocean University
[2]Institute of Information Science, Academia Sinica
[3]Department of Computer Science, National Tsing Hua University

ABSTRACT

While cloud gaming opens new business opportunity, it also poses tremendous challenges as the Internet only provides best-effort service and gamers are hard to please. Although researchers have various ideas to improve cloud gaming systems, existing cloud gaming systems are closed and proprietary, and cannot be used to evaluate these ideas. We present GamingAnywhere, the first open-source cloud gaming system, which is extensible, portable, and configurable. GamingAnywhere may be used by: (i) researchers and engineers to implement and test their new ideas, (ii) service providers to develop cloud gaming services, and (iii) gamers to set up private cloud gaming systems. Details on GamingAnywhere are given in this paper. We firmly believe GamingAnywhere will stimulate future studies on cloud gaming and real-time interactive distributed systems.

Categories and Subject Descriptors: H.5 [Information Systems Applications]: Multimedia Information Systems

Keywords: Remote gaming; performance evaluation

1. MOTIVATION

Computer games have become very popular, e.g., gamers spent 24.75 billion USD on computer games, hardware, and accessories in 2011 [6]. Traditionally, computer games are delivered either in boxes or via Internet downloads. Gamers have to *install* the computer games on physical machines to play them. The installation process becomes extremely tedious because some games are complicated and the computer hardware and system software are fragmented. Take Blizzard's Starcraft II as an example, it may take more than an hour to install it on an Intel i5 PC, and another hour to apply the online patches. Furthermore, gamers may find their computers are not powerful enough to enable all the visual effects yet achieving high frame rates. Hence, gamers have to repeatedly upgrade their computers so as to play the latest computer games.

Cloud gaming is an alternative way to deliver high-quality gaming experience and opens new business opportunity [13].

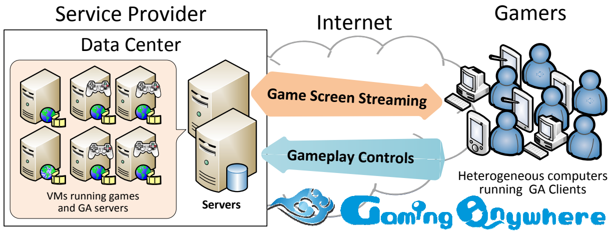

Figure 1: A cloud gaming service based on GamingAnywhere.

In a cloud gaming system, computer games run on powerful cloud servers, while gamers interact with the games via networked thin clients. The thin clients are light-weight and can be ported to resource-constrained platforms, such as mobile devices and TV set-top boxes. Figure 1 illustrates a sample cloud gaming service, in which game servers stream game screens to thin clients, and thin clients send gameplay controls, such as keystrokes and mouse movements, back to game servers. With cloud gaming, gamers can play the latest computer games anywhere and anytime, while game developers can optimize their games for a specific server configuration. The huge potential of cloud gaming has been recognized by the game industry: (i) a market report predicts that cloud gaming market will increase 9 times between 2011 and 2017 [4] and (ii) several cloud gaming startups were recently acquired by leading game developers [3, 5].

Although cloud gaming is a promising direction for the game industry, achieving good gaming experience without excessive hardware investment is a tough challenge. This is because gamers are hard to please, as they concurrently demand for high responsiveness and high video quality, but do not want to pay too much. Therefore, service providers have to not only design such systems to meet gamers' needs but also take efficiency, error resiliency, scalability, and resource allocation into considerations. This renders the design and implementation of cloud gaming systems extremely challenging. Indeed, while real-time video streaming seems to be a mature technology at the first glance, cloud gaming systems have to execute games, handle gamers' inputs, and perform screen rendering, capturing, encoding, packetizing, transmitting, decoding, and displaying in real time, and thus are much more difficult to optimize.

Fortunately, many system researchers have new ideas to improve cloud gaming experience for gamers, and reduce capital expenditure (CAPEX) and operational expenditure (OPEX) for service providers. However, all the existing

MM'13, October 21–25, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502222.

Table 1: Current supported platforms

Program	**Windows**	**Linux**	**Mac OS X**
Server	Windows 7+	Supported	No audio support
Client	Windows XP+	Supported	Supported

cloud gaming platforms [7, 12, 14] are closed and proprietary, which prevent the researchers from testing their ideas on real cloud gaming systems. Therefore, the new ideas were either only tested using simulators/emulators, or, worse, never evaluated and published. Without actual experiments, it is harder for new research ideas on improving cloud gaming (or real-time interactive distributed systems) to be transferred to the industry.

To better bridge the research community and the game/software industry, we initiate GamingAnywhere, which is the first open-source cloud gaming testbed. While in its infancy, we hope GamingAnywhere could gather enough attentions, and quickly grow into a community with critical mass, just like OpenFlow [11], which shares the same motivation with GamingAnywhere in a different research area.

The design philosophy of GamingAnywhere includes:

- *Extensibility*: GamingAnywhere adopts a modularized design. For example, platform-dependent components such as audio and video capturing and platform-independent components such as codecs and network protocols can be easily modified or replaced.
- *Portability*: Gamers may use devices with heterogeneous architectures to access GamingAnywhere, and thus it is important to port GamingAnywhere to as many platforms as possible. Table 1 summarizes the supported platforms as of July 2013. New platforms can be easily supported by replacing the platform-dependent components in GamingAnywhere.
- *Configurability*: System researchers may conduct experiments with diverse system parameters. A large number of built-in audio and video codecs are supported by GamingAnywhere. In addition, GamingAnywhere exposes all available configurations to gamers so that it is possible to try out the best combinations of parameters by simply editing a text-based configuration file for various usage scenarios.
- *Openness*: GamingAnywhere is publicly available at `http://gaminganywhere.org` [8]. Researchers are free to use and modify GamingAnywhere in academic research as long as the open-source license terms are followed.

2. TARGET USERS

GamingAnywhere is designed for three types of users. First, researchers and engineers may use GamingAnywhere to evaluate their new ideas without re-implementing the irrelevant software components. This will allow many new ideas to be tested in real testbeds and deployed in fields. Second, cloud gaming service providers may develop a cloud gaming front-end for gamers to choose the desired games, and integrate their front-end with GamingAnywhere. Compared to developing a brand new cloud gaming system from scratch, adopting GamingAnywhere largely reduces the time to market of new services. Third, gamers may set up GamingAnywhere on their powerful home PCs, and enjoy high-quality gaming experiences using thin clients via a LAN or the Internet. To reach wider audience, GamingAnywhere is written in C/C++ and it leverages several popular open-source libraries such as *ffmpeg* and *libSDL*. Detailed designs and implementations can be found in Huang et al. [10]. We also hope users contributing to this project by installing and playing with the system, providing comments and suggestions, sharing game configurations, and even submitting patches via the forum or public respositories.

3. DIRECTORY STRUCTURE

GamingAnywhere is released with two types of software packs: `all-in-one` and pre-compiled binary packs [8]. The `all-in-one` pack includes GamingAnywhere source codes, third-party library source code, pre-compiled binaries, while a pre-compiled binary pack such as the `bin.win32` pack includes only pre-compiled binaries for running the software on a certain platform. In any case, users can always build their own binaries that match their environments from scratch.

There are six subdirectories upon uncompressing the `all-in-one` pack; their descriptions are given below:

- `bin/`: Contains all the required run-time files, including the executables, shared objects, modules, fonts, and configuration files.
- `ga/`: Source codes of GamingAnywhere, which is divided into four parts: core, module, server, and client.
- `deps.posix/`: Dependencies for POSIX platforms, including libraries and headers. GamingAnywhere supports two POSIX platforms: Linux and Mac OS X. This subdirectory is empty by default, as the files in this subdirectory are supposed to be built from the sources in `deps.src`.
- `deps.win32/`: Dependencies for Win32 platform, including libraries and headers. This subdirectory is also empty by default, and the files in this subdirectory will be installed from the third-party packages in `deps.pkg.win32`.
- `deps.src/`: Source files of the dependent third-party packages. You will need these files to build dependencies, which is mandatory for POSIX platforms.
- `deps.pkg.win32/`: Pre-built dependencies for Win32 platform. You can install all the dependencies for Win32 by running `install.cmd` in this subdirectory. The dependencies are installed into the `deps.win32` subdirectory by default.

4. SETUP

GamingAnywhere can be installed by simply uncompressing the software packs, with or without the source code. If the `all-in-one` software pack is downloaded, the complete GamingAnywhere system can be compiled from scratch. One exception is the dependent Windows libraries: because building them can be tricky, we include the pre-built Win32 libraries in the software packs.

We briefly describe how to build GamingAnywhere binaries below. GamingAnywhere consists of three binaries: (i) `ga-client`, which is the thin client, (ii) `ga-server-periodic`, which periodically captures the desktop or a window as the game screen, and (iii) `ga-server-event-driven`, which directly hooks into the game executables to capture game screens.

POSIX. To build GamingAnywhere on POSIX platforms (i.e., Linux and Mac OS X), please make sure

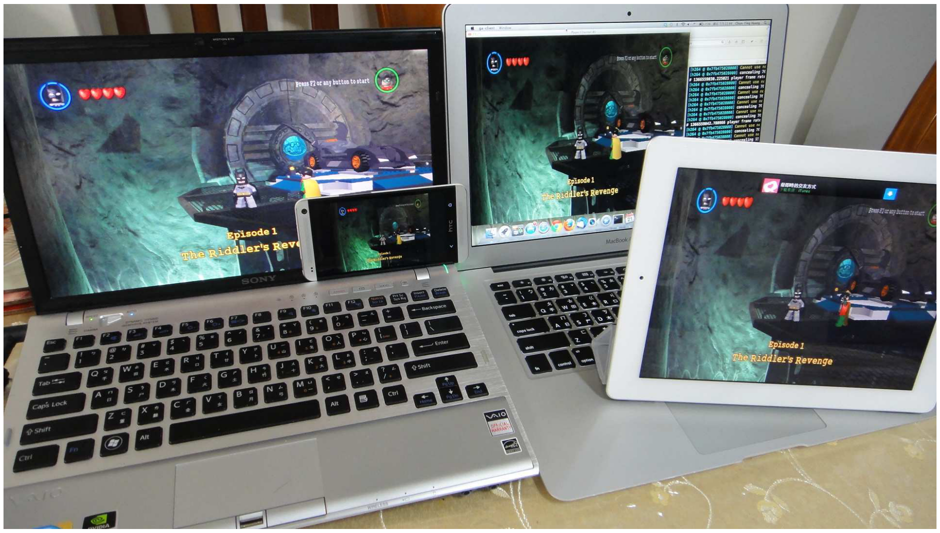

Figure 2: A demonstration of GamingAnywhere system. There are four devices in the photo. One game server (the left-hand side laptop) and three game clients (a MacBook, an Android phone, and an iPad 2).
† LEGO Batman software is copyrighted by Traveller's Tales Games Publishing Ltd.

`g++`, `pkg-config`, `libX11`, `libXext`, `libXtst`, `libfreetype6`, and `libasound2` are installed. In addition, `libgl1-mesa`, `libglu1-mesa`, and `libpulse` are required on Linux. You must install both binaries and development files for the above packages. Next, the following instructions are recommended:

1. Edit the `env-setup` script and point `GADEPS` to the absolute path of `gaminganywhere/deps.posix/`.
2. Load environment variables from `env-setup` by using the `.` or `source` commands.
3. Build dependencies using `make` under `deps.src/`.
4. Build GamingAnywhere using `make all` under `ga/`.
5. Install GamingAnywhere using `make install` command under `ga/`. All the generated binaries will be copied into `bin/`.

Windows. To build GamingAnywhere on Microsoft Windows, please install Visual Studio (C++) 2010, Windows SDK, and DirectX SDK. Then, follow the instructions:

1. Install dependencies by running `install.cmd` under `deps.pkg.win32\`.
2. Install DirectX SDK into `C:\Microsoft DirectX SDK`. If you install it at a different location, please modify the SDK paths in `ga\module\vsource-desktop\NMakefile.d3d` and `ga\server\event-driven\NMakefile`.
3. Launch Visual C++ command line prompt.
4. Build GamingAnywhere by running `nmake /f NMakefile all` command under `ga\`.
5. Install GamingAnywhere by running `nmake /f NMakefile install` command under `ga\`. All the generated binaries will be copied into `bin\`.

5. EXECUTION

This section explains how to run GamingAnywhere servers and client. Before running the binaries, please check the following prerequisites. On POSIX, please add `deps.posix/lib` to the system-wide `ld.so` search path. Moreover, modify `bin/config/common/server-common.conf` to ensure that `display` is set to to an X-Window desktop with the XTEST extension enabled. The X-Window desktop can be a `VNC` or `Xvfb` desktop. On Windows, copy `deps.win32\bin\*.dll` to the same directory of GamingAnywhere binaries (i.e., `bin\`) or a directory in the system-wide DLL search path.

We give examples on how to run GamingAnywhere servers and client using DirectX SDK samples on Windows.

- **Run the periodic capture server.** To stream a game window to the client, launch the game first and then use the following command: `ga-server-periodic config\server.d3dex.conf`. The server configuration file sets either the `find-window-name` or `find-window-class` option to specify the title and the class name, respectively, of the game window to be streamed. Then, `ga-server-periodic` moves the specified window to the upper-left corner of the desktop, and starts capturing the images of the specified window periodically. Gamers should avoid moving the window manually as it may affect the mouse cursor functionality. If you do not know the window title or the class name, you can find the information for a window by using the `Spy++` tool and the `xwininfo` tool on Windows and Linux, respectively.
- **Run the event-driven server.** Rather than capturing a game's window periodically, GamingAnywhere provides another sample server which is able to capture the game window each time right after the game updates its screen. Please use the following command: `ga-server-event-driven config\server.d3dex.conf`. `ga-server-event-driven` launches the game executable specified by the `game-dir` and `game-exe` options in the configuration file; that is why each game has its own configuration file.
- **Run the client.** The GamingAnywhere thin client can be run by `ga-client config\client.abs.conf rtsp://192.168.1.1:8554/desktop`, where the server IP address is 192.168.1.1; by default GamingAnywhere uses TCP port 8554 for RTSP streams and TCP/UDP port 8555 for control messages. The `client.abs.conf` is a sample configuration which enables the absolute mouse cursor posi-

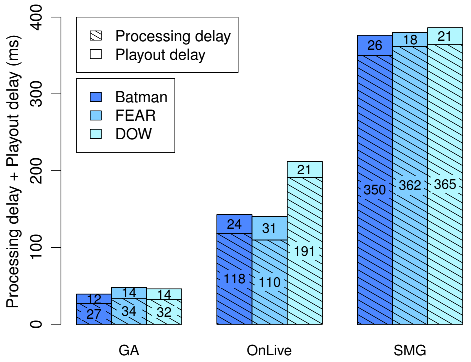

Figure 3: Response delays of GamingAnywhere, compared to two commercial systems, OnLive and StreamMyGame.

tioning. In contrast, `client.rel.conf` employs the relative mouse cursor positioning. The choice of either configuration files is game-dependent, as different games use different mechanisms for mouse position sensing and management.

6. DEMO

Figure 2 shows a GamingAnywhere testbed in our lab. In this figure, the laptop on the left-hand side runs the LEGO Batman game with GamingAnywhere event-driven server (i.e., `ga-server-event-driven`). The MacBook laptop on the right-hand side runs the GamingAnywhere client. Two additional clients also connect to the server at the same time. One client runs on an Android phone and another client runs on an iPad 2. Moreover, as GamingAnywhere server delivers audio and video frames using the standard RTSP/RTP protocol, any compatible RTSP/RTP players can be used to observe game plays.

We provide configuration files for several public-domain games [8], such as Armagetron Advanced, Assault Cube, and Neverball, so that gamers can quickly experience GamingAnywhere. Notice that the server configuration files are game-dependent, and therefore we provide a sample server configuration for each game. To get the server configuration working, please modify the `game-dir` and `game-exe` options in the configuration files to point to the game executables.

7. PERFORMANCE EVALUATION

Responsiveness is one of the most important factors affecting gaming experience [2, 9]. We define response delay to be the time difference between a user issuing a command and the in-game action appearing on the client. Cloud gaming systems have stringent response delay requirements, as gamers may demand for less than 100 ms response delay for certain games [2]. We adopt the measurement techniques proposed in [1] in order to compare the performance of GamingAnywhere against two commercial cloud gaming systems: OnLive [12] and StreamMyGame (SMG) [14].

The response delay consists of two components: (i) processing delay, which is the time required for the server to receive and process a gamer's command, and to encode and transmit the corresponding frame to the client and (ii) playout delay, which is the time required by the client to receive, decode, and render a frame on the display. We do not report network delay in our measurement studies as it is not controllable by the cloud gaming systems. Figure 3 summarizes the processing delay (at server) and the playout delay (at client) measured under 3 games and 3 cloud gaming systems. We note that the playout delay is small, ≤ 31 ms for all games and systems. Moreover, GamingAnywhere achieves much shorter processing delay (≤ 34 ms), while OnLive and SMG suffer from up to 191 and 364 ms processing delay. Considering the 100 ms requirement suggested by Claypool and Claypool [2], GamingAnywhere works well when the network delay is < 50 ms, which is fairly achievable, e.g., a GamingAnywhere server running at Los Angeles can serve a client at as far as Oklahoma City. Due to the space limitations, detailed evaluation results on GamingAnywhere can also be found in Huang et al. [10].

8. CONCLUSION

In this paper, we have presented GamingAnywhere, the first open-source cloud gaming system. Different from closed and proprietary cloud gaming systems, GamingAnywhere allows researchers and engineers to quickly explore their new ideas on how to improve gamers' experience and increase service providers' revenue. Furthermore, GamingAnywhere is also designed to be extensible, portable, and configurable, and thus can be adopted in various usage scenarios, e.g., service providers may use GamingAnywhere to build up cloud gaming services, and gamers may use GamingAnywhere to set up private cloud gaming systems. We hope GamingAnywhere can become the cornerstone for multimedia research community to build and evaluate many cloud gaming and real-time interactive distributed systems in the future.

References

[1] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei. Measuring the latency of cloud gaming systems. In *Proc. of ACM Multimedia 2011*.

[2] M. Claypool and K. Claypool. Latency and player actions in online games. *Communications of the ACM*, 49(11):40–45, 2006.

[3] Cloud gaming adoption is accelerating ... and fast!, 2012. `http://www.nttcom.tv/2012/07/09/cloud-gaming-adoption-is-acceleratingand-fast/`.

[4] Distribution and monetization strategies to increase revenues from cloud gaming, 2012. `http://www.cgconfusa.com/report/documents/Content-5minCloudGamingReportHighlights.pdf`.

[5] Electronic Arts buys online gaming studio ESN, the developers behind battlefield's battlelog online social network, 2012. `http://techcrunch.com/2012/09/26/electronic-arts`.

[6] Entertainment Software Association: Industry Facts, 2013. `http://www.theesa.com/facts`.

[7] Gaikai web page, 2012. `http://www.gaikai.com/`.

[8] GamingAnywhere: An open source cloud gaming project, 2013. `http://gaminganywhere.org`.

[9] T. Henderson. *The Effects of Relative Delay in Networked Games*. PhD thesis, Department of Computer Science, University of London, 2003.

[10] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen. GamingAnywhere: An open cloud gaming system. In *Proc. of ACM MMSys 2013*.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar. 2008.

[12] OnLive web page, 2012. `http://www.onlive.com/`.

[13] P. Ross. Cloud computing's killer app: Gaming. *IEEE Spectrum*, 46(3):14, 2009.

[14] StreamMyGame web page, 2012. `http://streammygame.com/`.