The Game Trace Archive

Yong Guo and Alexandru Iosup
Delft University of Technology, The Netherlands
Email: {Yong.Guo, A.Iosup}@tudelft.nl

***Abstract*—Spurred by the rapid development of the gaming industry and the expansion of Online Meta-Gaming Networks (OMGNs), many gaming studies and measurements have been conducted in recent years. However, few or no traces of games and OMGNs are publicly available to game researchers and practitioners. Moreover, the few traces that are available are shared using diverse formats. This situation is an obstacle in exchanging, studying, and using game traces. To address this problem, we design the Game Trace Archive (GTA) to be a virtual meeting space for the game community. We identify three main requirements to build an archive for game traces, and address them with the GTA. We propose a unified format for game traces, and introduce a number of tools associated with the format. With these tools, we collect, process, and analyze 9 traces of both games and OMGNs. We collect in the GTA traces corresponding to more than 8 million real players and more than 200 million information items, spanning over 14 operational years. We also show that the GTA can be extended to include a variety of real-game trace types. Finally, we discuss possible applications of the GTA in gaming area such as game resource management, Quality of Experience for players, and advertisement.**

I. Introduction

Over the last decade, video and computer gaming have become increasingly popular branches of the entertainment industry. Understanding the characteristics of games and OMGNs such as player behavior [1], [2], traffic analysis [3], [4], resource management [5], [6], etc., is essential for the design, operation, and tuning of game systems. However, only a small number of game traces exist; even fewer can be accessed publicly. As a consequence, previous studies used at most a few traces and no comprehensive comparative analysis exists. Moreover, the few available game traces have diverse formats, which makes it difficult to exchange and use these game traces among game researchers. To address this situation, in this work we propose the Game Trace Archive.

There are thousands of successful games in the world, which attract a large number of players. For example, World of Warcraft, one of the most popular Massively Multiplayer Online Games (MMOGs), and CityVille, a widely spread Facebook game, have each tens of millions of players. A number of online communities have been constructed by game operators and third-parties around single games or even entire collections of games. These communities through the relationships between entities such as players and games, form Online Meta-Gaming Networks [7]. Tens of millions of players currently participate in OMGNs, such as Valve's Steam, XFire, and Sony's PlayStation Network, to obtain game-related information (game tutorials, statistic information, etc.) and use none-gaming functionalities (voice chat, user product sharing, etc.).

We design in this work the GTA as a virtual meeting space for the game community, in particular to exchange and use game traces. Game traces, which can be collected at one timepoint, several or series of timepoints, or through a continuous period of observation, contain many types of game-related data about both games and OMGNs. With the GTA, we propose a unified Game Trace Format (GTF) to store game traces, and provide a number of tools to analyze and process game traces in GTF format. Our goal is to make both the game traces and the tools in the GTA publicly available. Mainly because of the diversity and big size of game traces, there are three main challenges in building the GTA. Firstly, game traces can be collected from many sources. They can focus on any of the multiple levels of the operation of gaming systems, from OMGNs to single game traces, from players to player relationships, and to packets transferred between the players and the game servers. The trace content at different levels is significantly different; moreover, even at the same level the traces may be very different. Secondly, the content of each trace can be complex. Traces may include many kinds of relationships between the game entities (players, guilds, etc.) and detailed information of entities (player name, player date of birth, in-game and meta-game information, etc.). Thirdly, it is difficult to process the large-scale and complex game traces, and to choose from tens of or hundreds of game traces depending on specific scenarios.

This work is further motivated by our ongoing project @larGe, which aims at designing new systems that support gaming at large-scale. Trace archives support several other computing areas including the Parallel Workloads Archive (PWA) [8] for the parallel systems, the Grid Workloads Archive (GWA) [9] for the grid systems, etc. However, non of these previous archives can include the complex game traces we target with the GTA. **Our research is the first work in establishing a comprehensive Game Trace Archive to benefit gaming researchers and practitioners.** Towards this end, our main contribution in this work is threefold:

1. We synthesize the requirements for building an archive in the gaming domain (Section II).
2. We design the Game Trace Archive, including a unified format and a toolbox to collect and share game and OMGN traces (Section III).
3. We conduct a comparative analysis using many real game traces (Section IV) and identify several potential applications for the Game Trace Archive (Section V).

978-1-4673-4578-1/12/$31.00

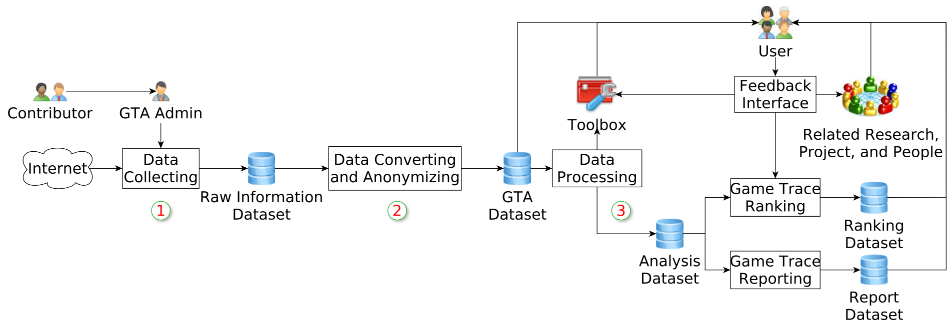

Fig. 1. The design of the Game Trace Archive.

II. Requirements for a Game Trace Archive

Starting from the three main challenges we mentioned in the introduction, in this section we identify three main requirements to build an archive for game traces. Although these requirements are similar to those of archives in other areas, our ambitious goal of building a single format for *all* game data makes it challenging to achieve these requirements.

Requirement 1: Trace collecting. To improve game trace readability and facilitate game trace exchange, a unified game trace format should be provided to collect game traces from diverse sources. Firstly, the game trace format must be able to include many types of game information. Secondly, formats already exist for specific game information; for example, packets sent between game servers and clients. The unified game trace format needs to use these other formats. Finally, due to the rapid evolution of the gaming industry, many new games and game traces may emerge. The format should be extensible to collect traces of future games, while not affecting the usage of old game traces that have already been stored in the archive.

Requirement 2: Trace converting and anonymizing. The raw content of game traces is complex. However, a small part of the content may not be game-related, because of where the game traces are collected. For example, when crawling OMGN traces from their websites, advertisement information in each website may also be collected into the trace. Thus, during the procedure of converting game traces to the unified game trace format, this useless content should be filtered out. Another important issue in converting traces is to provide a privacy guarantee. Many studies [10], [11] have shown that just anonymizing user names is not sufficient to ensure privacy.

Requirement 3: Trace processing. The archive should provide a toolbox to process the converted game traces and generate reports including commonly used characteristics of game traces (e.g., trace size, the number of information items). Furthermore, the toolbox could also be used by archive users to build comprehensive analysis tools.

III. The Game Trace Archive

In this section, we introduce the Game Trace Archive. We first discuss the design of the GTA and then describe the design of the Game Trace Format (GTF).

A. The Design of the GTA

Figure 1 illustrates the overall design of the Game Trace Archive, including the GTA members and how the game traces been processed in the GTA. The circles in Figure 1 represent the three requirements we formulated in Section II.

We envision three main roles for the GTA members. The *contributor*, who is the legal owner of game traces, offers their traces to the GTA and allows public access to the traces. The *GTA Admin* helps contributors to add and convert game traces to the GTA, and manage traces processing and sharing. Our game research team may act as the GTA Admin. The *user* accesses the archived game traces, uses the processing toolbox, and obtains the relevant research information through the GTA. Most users may be game researchers and practitioners, but we believe that people in other areas (e.g., biologists, economists, social network researchers) can also benefit from the GTA.

In the design of the GTA, the *Data Collecting* module is for collecting game traces from multiple sources: contributors, publicly shared game data repositories, game websites, etc. These game traces have their own formats and some of them may include sensitive information. Firstly, we store these traces in a raw information dataset without additional processing. Then, we map the raw content to the unified GTF (for requirement 1).

The *Data Converting and Anonymizing* module is responsible for converting the game traces from their own formats to the unified GTF, while anonymizing the sensitive information (for requirement 2). The anonymization process, which is a topic of research in itself [10], [11], is outside the scope of this work. The map from the original information and the anonymized information will not be distributed, only the corresponding trace contributor has the authority to read it.

In the *Data Processing* module, a toolbox is provided for comprehensive trace analysis: overview information, such as trace size, period, the number of relationships, and the number of players; in-game characteristics, such as active playing time, average session duration, and the number of played games; relationship graph metrics, such as diameter, link diversity, clustering coefficient. The basic tools in the toolbox can be used to build other processing tools and can also be used to process large scale graphs in other areas, such as social network and viral marketing(for requirement 3).

Two modules are designed for trace sharing. The *Game Trace Reporting* module receives the trace analysis results from Data Processing module and formulate them into more visible reports. The *Game Trace Ranking* module considers both the overview information from Data Processing module and feedback from trace users to rank the game traces. The Game Trace Reporting and Ranking modules help the GTA users to select a game trace based on a quick knowledge of the basic game trace characteristics.

The *Feedback Interface* is for supporting trace sharing and community building. It is the interface for users to submit their feedback after using game traces in the archive. There are four types of feedback: rank of traces, comment on traces, updated or newly designed tools for traces, and research information (e.g., research direction, project, applications for traces). For each game trace in the GTA, we maintain a list of research information derived from feedback. Through these lists, users can know the game community better and users in similar research directions may establish further communication.

B. The Design of the GTF

We propose the Game Trace Format to facilitate the exchange and ease the usage of game traces. It is a unified format to cover many different types of game traces and to include complex and detailed gaming information in each trace. The technical details of the GTF can be found in our technical report [12]. In this subsection, we introduce the main elements of the design of the GTF.

Figure 2 shows the structure of the Game Trace Format. Briefly, the GTF consists of three datasets: the *Relationship Graph Dataset*, the *Node Dataset*, and the *Other Game-related Dataset*. These datasets are responsible for storing different kinds of content in game traces respectively.

From our observation, in many game traces the relationships (e.g., play with, send message to, member of) between many kinds of game entities (e.g., player, group, game in OMGN, genre in OMGN) are significant for the operation of these games and OMGNs. Thus, we model these relationships as *edges* in a *graph* where *nodes* are game entities. We include in the GTF the Relationship Graph Dataset to include the relationships presented in game traces. The Relationship Graph Dataset has three sub-datasets: with *basic edge information*, with *detailed edge information*, and with *meta-information*. The basic edge information includes the essential or must-have elements for relationships (e.g., edge/node type, edge/node identifier). By using different edge types and node types, various relationships in game traces can be presented in our format. To store other diverse edge-related information, we use the detailed edge information sub-dataset, which includes two parts, the *fixed part* and the *extended part*. The fixed part stores typical attributes of edges (e.g., timestamp, edgelifetime). The extended part stores extended edge attributes that are not common for all relationships. The design of extended part makes it possible to cover new types of edge-related information. The meta-information includes the overview of the Relationship Graph Dataset and the definitions of all the edge and node types.

The Node Dataset is designed to include detailed node information. Since the information of different types of nodes can be diverse, to store this information in a unified format, we categorize the information first. We divide the complex node information into *static* (keep constant with time, e.g., player gender, date of birth) and *dynamic information* (change with time, e.g., player rank, level). Each type of node has its own *node sub-dataset*. For each type of node, we further divide their static and dynamic information into *typical static information*, *typical dynamic information*, *extended static information*, and *extended dynamic information*. These four kinds of information are stored in *fixed static part*, *fixed dynamic part*, *extended static part*, and *extended dynamic part*, respectively in the node sub-dataset. Through its design, specifically through its extended static and extended dynamic parts, our Node Dataset is the first to store the information of many kinds of nodes in a unified format.

For more traditional gaming data, such as packets between game clients and servers, match replays, player click streams, etc., we use when possible the de-facto standard formats and store the data in the Other Game-related Dataset. For example, for match replays, we keep their own formats derived from games. The formatted replays, such as StarCraft replays, are used by Weber et al. [13] and Hsieh et al. [14] to study player in-game behavior. For packet traces, we keep raw `tcpdump` and `wireshark` data. Moreover, we provide detailed introduction files to guide the archive users how to process the formats. Meta-information is also provided, to link the game-related information to its corresponding nodes; for example, the links between players and the packets they have sent.

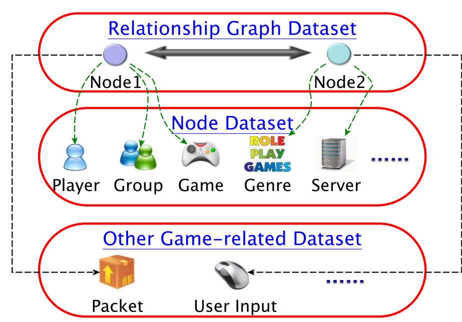

Fig. 2. The structure of the Game Trace Format.

IV. Game Traces and Trace Analysis

In this section, we present and analyze the game traces collected by the GTA. Using our GTA toolbox, we have conducted comprehensive analysis on workload characteristics, winning, and long-term player behavior of match-based games in our previous study [15] and on the change of daily active users and player departure in our technical report [12].

Table I summarizes the nine game traces currently formatted in the GTF. These traces have been collected from five types

978-1-4673-4578-1/12/$31.00 ©2012 IEEE

TABLE I
SUMMARY OF GAME TRACES IN THE GTA.

Trace	Period	Size (GB)	# Nodes (K)	# Links (M)	Genre
KGS[1]	2000/02-2009/03	2	832	27.4	board
FICS[2]	1997/11-2011/09	62	362	142.6	board
BBO[3]	2009/11-2009/12	2	206	13.9	card
XFire[4]	2008/05-2011/12	58	7,734	34.7	OMGN
Dota-League[5]	2006/07-2011/03	23	61	3.7	RTS
DotAlicious[6]	2010/04-2012/02	1	64	0.6	RTS
Dota Garena[7]	2009/09-2010/05	5.2	310	0.1	RTS
WoWAH [16]	2006/01-2009/10	1	91	N/A	MMORPG
RS [17]	2007/08-2008/02	1	0.1	N/A	MMORPG

of games or OMGNs, including board games, card games, RTS games, MMORPG games, and OMGNs. Our GTA can cover game traces collected by ourselves. The KGS and FICS traces include a large number of matches in two popular board games, Go, and chess. The BBO trace is collected from one of the largest bridge sites in the world, where people can play bridge online for free. The dataset of the OMGN XFire contains the information of thousands of games and millions of players, as well as complex relationships between those games and players. Defense of the Ancients (DotA) is mod build on the RTS game Warcraft III. A match of DotA is played by two competing teams, allowing at most 5 players in one team. The main content in the Dota-League, DotAlicious, and DotA Garena traces is plenty of DotA matches played on three different DotA playing platforms. We are also able to include in the GTA existing game traces. WoWAH is a public dataset consists of online session records and attributes of World of Warcraft avatars collected from game servers in Taiwan. RS collects the online player counts of more than 100 servers of the MMORPG game RuneScape. The size in Table I is the raw data size of each trace. The node may be player, avatar, or game server, corresponding to each trace. In XFire, we use the link as friendship. For the other traces, we define the link as playing in a same match. With these game traces, we can conduct a comparative analysis of games, inter- and intra-genre.

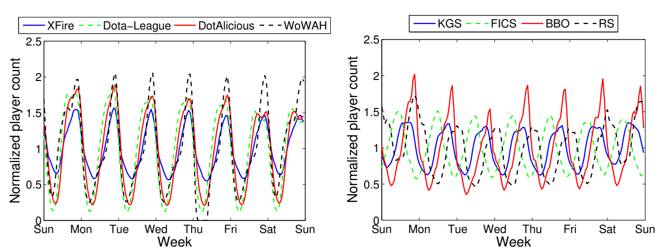

Fig. 3. Weekly pattern of normalized online player count.

[1]http://www.gokgs.com/
[2]http://www.freechess.org/
[3]http://www.bridgebase.com/
[4]http://beta.xfire.com/
[5]http://www.dota-league.com/
[6]http://www.dotalicious-gaming.com/
[7]http://replay.garena.com

A. Weekly pattern of online player count

The online player count is an important metric of the usage and workload of game severs. In this subsection, we study how the online player count fluctuates during the week.

We define the *normalized online player count* at any moment of time, as the ratio between the player count at that moment and the trace-long average player count. Figure 3 illustrates the normalized online player count for eight game traces. The ninth trace (Dota Garena) currently included in the GTA does not have player count information over time. The figure shows that, for each game trace, the online player counts have obvious diurnal patterns; the peak and bottom counts occur at nearly the same time for each day. However, the occurring time of both peak and bottom differ per game. For example, for chess the peak occurs during the day, whereas midnight is the busiest period for all DotA platforms. Unexpectedly and unlike the workloads of web servers, for our traces the player counts do not differ significantly between week days and week ends. Moreover, for DotAlicious the player presence is even lower during week ends. Due to the scheduled weekly maintenance, there is an outage period on Thursday morning in WoWAH; the normalized player count drop to 0, see Figure 3 (left).

Lee et al. [16] investigate the daily, weekly, and monthly patterns of avatar counts in World of Warcraft. Chambers et al. [5] study the online player count over a 4-week period in FPS, casual, and MMORPG games. Both of their results show a similar weekly pattern as our result, but for fewer games.

B. Gaming graph

The relationships between entities in game traces can form gaming graphs. As we will argue in Section V, the structure and evolution of gaming graphs may become essential for game operation. The formatted Relationship Graph Dataset in the GTA makes it easier to investigate the graph characteristics of various games. In this subsection, we analyze the gaming graph for seven traces, except WoWAH and RS, which do not have the graph information. We form the graphs as follow: for all traces, each node represents a player. Then, for XFire each edge expresses a player-to-player friendship; for all the other traces, each edge represents a played game match.

Table II shows, for each trace, typical graph metrics of the largest connected component of the gaming graph. A connected component is a sub-graph in which all pairs of nodes are reachable. The largest connected component contains almost all the nodes of the whole graph in each game trace, which means that nearly every player can reach any other player in the trace. From the link density, all these largest connected components are *not* dense. For XFire, the average friend count per user ($\bar{\mathbf{D}}$ in Table II) is much lower than for Facebook: 8 vs. 130. For the other traces, the relationship counts range from 19 to 1663; we leave for future work a study of the strength and meaning of such game relationships.

TABLE II
GRAPH METRICS OF THE LARGEST CONNECTED COMPONENT.

Trace	# Comps	# Nodes	# Edges	d ($\times 10^{-4}$)	$\bar{D}$
KGS	6,099	819,249	17,884,783	0.53	44
BBO	11	206,333	13,654,906	6.41	132
FICS	2,574	356,244	50,460,185	7.95	283
XFire	198	7,733,276	29,257,283	0.01	8
Dota-League	1	61,171	50,870,316	272	1663
DotAlicious	1	64,083	20,006,143	97.4	624
Dota Garena	1,544	291,706	2,767,594	0.65	19

 # Comps is the number of connected components in the graph.
 # Nodes, **# Edges**, **d**, and $\bar{\mathbf{D}}$ are the number of nodes, the number of edges, the link density, and the average degree of the largest connected component, respectively.

V. APPLICATIONS OF THE GAME TRACE ARCHIVE

Many research directions may benefit from the comprehensive game information stored in the GTA. In this section, we discuss the potential applications of the GTA. We look at three main application categories, game resource management, Quality of Experience (QoE) for players, and advertisement.

A. Game resource management

Resource management is critical for game and OMGN operators and players. Inadequate management can lead to service shut-down, player departure, etc. We identify four main uses of GTA traces for game resource management.

1) *Provisioning the overall resources needed for game and OMGN operation* [7]. Knowledge about the evolution of the player graph (e.g., a graph formed as in Section IV-B), from the simple player count to the complex guild information, can enable accurate prediction of the needed resources.
2) *Deploying resources*. The GTA can help in understanding the change of game workloads with different time patterns (diurnal, weekly, etc.). Idle game resources of one game may be used to support other games or applications during a specific period in a day or week [5].
3) *Reducing resource consumption*. The use of positional update messages, which account for a large portion of network consumption, can be tuned to play characteristics [18]. We believe that the GTA can further help with this tuning process. Firstly, we may identify groups and core group players from player graphs. Then, using the core players as proxies for their groups, the game may transfer only the core players' characters positional updates and significantly reduce the network consumption.
4) *Assigning relevant resources to different types of players*. The types or importance of players in a group can be identified from their playing behavior. For example, when supporting a group voice-communication [19], the group leaders may need more upload bandwidth than other group members for sending their commands, because they have multiple voice channels, one for each other group member.

B. Quality of Experience for players

The Quality of Experience for players covers a wide range of aspects. In the following, we list four applications that could use the GTA to improve the QoE.

1) *Improving match-making systems* [20]. Considering more player information stored in the GTA such as skill, rank, win ratio, and the friendship graph may improve the quality of match-making systems and player gaming experience. For example, considering friendships and assigning friends in a same team may allow better match-making.
2) *Detecting cheaters* [21]. The behavior of cheaters may be different from that of normal players, especially different from that of the cheaters' graph neighbors. Identifying graph neighbors and extracting their behavior pattern may be helpful for distinguishing and confirming cheaters.
3) *Building reputation systems* [22]. The reputation of a player (ratee) is calculated from ratings given by the other players (raters). During the calculation, each rating should have its own weight, which may be assigned according to many metrics derived from the GTA data, such as the level and online time of rater, the relationship of rater and ratee in the gaming graph, etc.
4) *Recommending in OMGNs: games, friends, videos, and screenshots*. Better recommendation should be made based on players' interests and their relationship graph.

C. Advertisement

Advertisement is an important source of revenue for game operators. Successful game advertisements should meet at least the following requirement: for advertisers, advertisements should attract more users than their expectation; for game and OMGN operators, they should obtain more income from advertisement than the revenue lose due to the effect of user experience. To achieve this requirement, the content of advertisements should be well designed and the placement of advertisements should be well selected. We focus in this work on potential application of the GTA data to select the location and moment to integrate advertisement.

1) *Integrating advertisements into games with less influence to gaming experience* [23]. If an advertisement interrupts player immersion, especially during important and time-limited tasks, the advertisement may fail. What is worse, this may results in player departure. Analyzing the type and frequency of player in-game operations may help advertisers and game operators to integrate seamless advertisements.
2) *Finding places for advertising in OMGNs*. Delivering advertisements in homepages of OMGNs is a common way to prompt potential users. However, there may be other good places for displaying advertisements, for example, popular discussion topics. These popular topics can be identified from the GTA data, such as the number of responses, the influence of participants, and the links between participants in OMGNs.

VI. Related Work

Most previous research in gaming is based on individual game traces [1], [2], [3], [4]. In contrast, a recent study [5] uses four game traces in three different game genres; there is insufficient intra-genre data.

A number of archives have been build in the other computer science areas. The Internet Traffic Archive (ITA) [24] archives internet network traffic traces for the research of network characteristics. The Parallel Workloads Archive (PWA) [8] collects workloads from parallel environments. The Grid Workloads Archive (GWA) [9] is a grid trace repository and a community center for the grid area. For the peer-to-peer community and wireless network community, the Peer-to-Peer Trace Archive (P2PTA) [25] and CRAWDAD [26] are designed, respectively. These archives are limited to their specific areas and cannot cover the game traces.

Social network and large-scale graph analysis are popular research topic in recent years. Large-scale graphs also exist in the gaming area, forming by the game entities and their relationships. We already store more complex information, such as more types of nodes and timed relationships, in our gaming graphs. Many formats have been designed to enable the exchange of social networks and graphs [27], [28], but none of them is able to cover the gaming graphs.

VII. Conclusion and Future Work

We have designed the Game Trace Archive to address the lack of game traces for the game community. We have proposed a unified Game Trace Format that underlies a generic game and OMGN data repository. We have also provided a toolbox and mechanisms to facilitate trace sharing and community building. The currently archived traces and our example trace analysis have shown that the GTA can already store different types of game traces, and that the GTA supports comparative analysis of gaming traces. We have discussed potential applications for using the GTA in three main aspects.

For the future, we plan to improve the Game Trace Archive in three ways: find more contributors and game traces to enrich the dataset, process game traces in more depth to generate more comprehensive reports, extend the functionality of GTA by building game trace generators.

Data Availability and Contributions

The Game Trace Archive is available online at: *http://gta.st.ewi.tudelft.nl*. We are in possession of hundreds of game traces, of various length, collected independently or with the help of gaming companies. We plan to upload one trace per month until the GTA stores at least 20 online traces. We are looking for contributors who would like to share their game traces through the GTA.

Acknowledgment

The authors would like to thank Marcin Biczak, Boxun Zhang, and Siqi Shen. This work is supported by the NSFC No.61272483 and by the STW/NOW Veni grant 11881.

References

[1] M. Suznjevic, I. Stupar, and M. Matijasevic, “MMORPG player behavior model based on player action categories,” in *NetGames*, 2011.

[2] N. Ducheneaut, N. Yee, E. Nickell, and R. Moore, “The life and death of online gaming communities: a look at guilds in world of warcraft,” in *ACM SIGCHI*, pp. 839–848, 2007.

[3] K. Chen, P. Huang, and C. Lei, “Game traffic analysis: an MMORPG perspective,” *Computer Networks*, vol. 50, no. 16, pp. 3002–3023, 2006.

[4] J. Kinicki and M. Claypool, “Traffic analysis of avatars in Second Life,” in *NOSSDAV*, 2008.

[5] C. Chambers, W. Feng, S. Sahu, D. Saha, and D. Brandt, “Characterizing online games,” *IEEE TON*, vol. 18, no. 3, pp. 899–910, 2010.

[6] A. Denault, C. Canas, J. Kienzle, and B. Kemme, “Triangle-based obstacle-aware load balancing for massively multiplayer games,” in *NetGames*, 2011.

[7] S. Shen and A. Iosup, “The XFire online meta-gaming network: observation and high-level analysis,” in *MMVE*, 2011.

[8] The Parallel Workloads Archive Team, “Parallel Workloads Archive,” July 2007. http://www.cs.huji.ac.il/labs/parallel/workload/.

[9] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. Epema, “The Grid Workloads Archive,” *FGCS*, 2008.

[10] A. Narayanan and V. Shmatikov, “De-anonymizing social networks,” in *Security and Privacy*, pp. 173–187, 2009.

[11] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Zhao, “Sharing graphs using differentially private graph models,” in *IMC*, pp. 81–98, 2011.

[12] Y. Guo and A. Iosup, “The Game Trace Archive: A Technical Report,” Tech. Rep. PDS-2012-005, Delft University of Technology, 2012. http://www.pds.ewi.tudelft.nl/fileadmin/pds/reports/2012/PDS-2012-005.pdf.

[13] B. Weber and M. Mateas, “A data mining approach to strategy prediction,” in *CIG*, 2009.

[14] J. Hsieh and C. Sun, “Building a player strategy model by analyzing replays of real-time strategy games,” in *Neural Networks*, 2008.

[15] Y. Guo, S. Shen, O. Visser, and A. Iosup, “An Analysis of Online Match-Based Games,” in *MMVE*, 2012.

[16] Y.-T. Lee, K.-T. Chen, Y.-M. Cheng, and C.-L. Lei, “World of Warcraft Avatar History Dataset,” in *ACM Multimedia Systems*, 2011.

[17] V. Nae, A. Iosup, and R. Prodan, “Dynamic Resource Provisioning in Massively Multiplayer Online Games,” *IEEE TPDS*, 2011.

[18] D. Ahmed and S. Shirmohammadi, “Improving online gaming experience using location awareness and interaction details,” *Multimedia Tools and Applications*, pp. 1–18, 2011.

[19] T. Hildebrandt, S. Bergsträßer, C. Rensing, and R. Steinmetz, “Dynamic voice communication support for multiplayer online games,” in *NetGames*, 2008.

[20] J. Fritsch, B. Voigt, and J. Schiller, “The next generation of competitive online game organization,” in *NetGames*, 2008.

[21] H.-K. Pao, K.-T. Chen, and H.-C. Chang, “Game Bot Detection via Avatar Trajectory Analysis,” *IEEE TCIAIG*, 2010.

[22] E. Kaiser and W. Feng, “PlayerRating: a reputation system for multiplayer online games,” in *NetGames*, 2009.

[23] B. Lewis and L. Porter, “In-game advertising effects: Examining player perceptions of advertising schema congruity in a massively multiplayer online role-playing game,” *Journal of Interactive Advertising*, 2010.

[24] P. Danzig, J. Mogul, V. Paxson, and M. Schwartz, “The Internet Traffic Archive,” April 2008. http://ita.ee.lbl.gov/.

[25] B. Zhang, A. Iosup, and D. Epema, “The Peer-to-Peer Trace Archive: Design and Comparative Trace Analysis,” Tech. Rep. PDS-2010-003, Delft University of Technology, 2010. http://pds.twi.tudelft.nl/reports/2010/PDS-2010-003.pdf.

[26] J. Yeo, D. Kotz, and T. Henderson, “CRAWDAD: a community resource for archiving wireless data at Dartmouth,” *ACM SIGCOMM CCR*, 2006.

[27] J. Leskovec, “Stanford Network Analysis Project (SNAP),” July 2009. http://snap.stanford.edu.

[28] M. Tsvetovat, J. Reminga, and K. Carley, “DyNetML: Interchange format for rich social network data,” Tech. Rep. CMU-ISRI-04-105, Carnegie Mellon University, 2004. http://www.casos.cs.cmu.edu/publications/papers/CMU-ISRI-04-105.pdf.