

CAMEO: Enabling Social Networks for Massively Multiplayer Online Games through Continuous Analytics and Cloud Computing

Alexandru Iosup
Delft University of Technology,
The Netherlands
A.iosup@tudelft.nl

Adrian Lăscăteu
Politehnica U. of Bucharest,
Romania
Adrian.Lascateu@cti.pub.ro

Nicolae Țăpuș
Politehnica U. of Bucharest,
Romania
NTapus@cs.pub.ro

ABSTRACT

Millions of people play Massively Multiplayer Online Games (MMOGs) and participate in the social networks built around MMOGs daily. These players turn into a collaborative community to exchange game news, advice, and expertise, but in return expect support such as player reports and clan statistics. Thus, the MMOG social networks need to collect and analyze MMOG data, in a process of continuous MMOG analytics. With the appearance of cloud computing, it has become attractive to use on-demand resources to run automated MMOG data analytics tools. In this work we present CAMEO, an architecture for Continuous Analytics for Massively multiplayer Online games on cloud resources. Our architecture provides various mechanisms for MMOG data collection and continuous analytics of a pre-determined accuracy in real settings. We implement and deploy CAMEO to perform continuous analytics on data from RuneScape, a popular MMOG. Using resources from various real clouds, including the commercial cloud of Amazon, CAMEO can analyze the characteristics of a community of over 3,000,000 active players, and follow the progress of 500,000 of these players for over a week. Thus, we show evidence that CAMEO can support the social networks built around MMOGs.

1. INTRODUCTION

Massively Multiplayer Online Games (MMOGs) have emerged in the past decade as a novel Internet application with a rapidly growing, world-wide user base of tens of millions of players. There exist now hundreds of MMOGs providers with thousands of MMOGs currently in operation; from these, FarmVille and World of Warcraft number each over 10,000,000 constant players. Around each game or groups of similar games, third-parties such as volunteers and small businesses have built online communities (social networks) that inform and entertain the players. These communities use MMOG analytics to improve visitor experience with player reports, progress charts, clan statistics, etc. While the analysis may differ, the data collection and analysis (collectively, the *game analytics*) can benefit from recent advances in the availability of on-demand resources through cloud computing services such as Amazon's Elastic Compute Cloud (EC2). In this work we present an architecture for MMOG analytics in the cloud.

Cloud computing has emerged in the past few years as a new paradigm for IT, in which the infrastructure, the platform, and even the software are outsourced services. These services have fixed cost and general Service Level Agreements and, most importantly, can be used when and for how long they are needed. There currently exist hundreds of commercial cloud service providers, such as Amazon, Microsoft, FlexiScale, and NewServers.

Online data crawling and analysis has often been employed in the past to determine the stationary and dynamic characteristics of Internet-based communities. However, the focus of the research community has been either in making the crawling process more parallel [3, 9], or analyzing the acquired data using more scalable parallel or distributed algorithms [1, 13]; both these approaches assume that enough resources are available for the task. Recently, data analysis in the cloud has received much attention, and generic programming models such as MapReduce [4] and Dryad [7] are now supporting large-scale processing both for the general public and for companies such as Google, Facebook, and Microsoft. In contrast to this body of previous work, which addresses a wide range of issues, in this work we focus on a domain-specific application, MMOG analytics, for which we design and build an integrated solution that uses on-demand resources. Our main contribution is three-fold:

1. We present a continuous analytics application, third-party MMOG communities, and its challenges (Section 2). This extends our previous work [6] with a specific, more constrained application of continuous MMOG analytics;
2. We present CAMEO, an architecture for continuous MMOG analytics (Section 3). We extend our previous work on CAMEO [6] in three main ways. First, we introduce several new MMOG-specific analysis to support MMOG social networks. Second, we extend the cloud-specific part of CAMEO to support data acquisition even under traffic limitation. Third, we extend the data-specific part of CAMEO to support more efficient data storage and transfer;
3. We show evidence that CAMEO can be used in practice by performing continuous analysis on RuneScape, a popular MMOG, on resources provisioned from either commercial or internally-owned clouds (Section 4). We perform many new types of analytics and the largest RuneScape analytics to date.



Figure 1: Size of four online communities built around World of Warcraft. Three of these communities are not maintained by the MMOG operator.

2. CONTINUOUS ANALYTICS FOR MMOG SOCIAL NETWORKS

MMOGs generate data that need to be analyzed at various levels of detail and for various purposes. As in our previous work on CAMEO [6], we define *continuous analytics for MMOGs* as the process through which relevant MMOG data are analyzed in such a way that prevents the loss of important events affecting the data. In this section we present our goal in supporting continuous analytics for MMOGs.

2.1 (Third-Party) MMOG Social Networks

Around each popular MMOG, the dedicated players and even commercial interests have created social networks that support active communities. Through collaborative paradigms such as Wikis, Data Mashups, and Web Services, these *MMOG social networks* aggregate information about the MMOG in the form of encyclopedic reports, tutorials, videos, and even player-customized information. Many of the MMOG social networks are built with the volunteered contributions of common players, who may in return get social rewards such as community recognition.

The MMOG social networks have to support large communities with hundreds of thousands of unique daily visitors. Often, the popularity of the social networks built “for the community, by the community” reaches the popularity of the communities built by the game’s operator. Figure 1 shows the size over time of four communities built around World of Warcraft: *worldofwarcraft.com*, which is built and maintained by the game operators, and the community-built *mmo-champion.com*, *thottbot.com*, and *wowwiki.com*. The four communities total over one million daily unique visitors at the beginning of 2010; the official community is the most important, but not by a wide margin.

2.2 Goal

The business model for MMOG social networks built by third-parties, that is, organizations not sponsored by the MMOG operator, is to obtain revenue from advertisements or from selling virtual goods and services. For this business model to function, the third-party web sites need to retain their visitors, which are only a subset of all the MMOG players; a big step toward visitor retention is to use MMOG analytics to improve visitor experience with player reports, progress charts, clan statistics, etc. These rely, in turn, on continuous MMOG analytics.

Our goal is to design a generic architecture for continuous analytics in support of social networks for MMOGs. We envision such a system operating in the

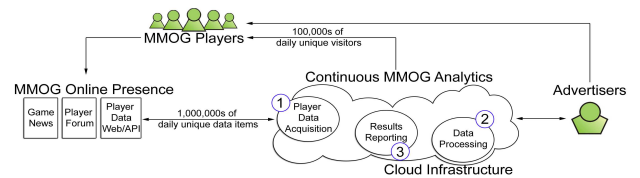


Figure 2: The MMOG/players/advertisers/third-party communities ecosystem, for a single MMOG.

players/MMOG operator/advertisers ecosystem depicted in Figure 2; the advertisers and community volunteers finance the operation. To materialize this vision, we introduce in Section 3 CAMEO, our cloud-based continuous MMOG analytics architecture, which provides the services labeled 1 through 3 in the figure.

2.3 Challenges

We identify four main challenges in achieving our goal: understanding user community needs, enabling and using distributed and collaborative technology, the combined data challenge, and several MMOG-specific challenges relating to system design. We describe each of the challenges, in turn.

- 1. Addressing the specific and dynamic needs of each community** MMOG communities vary by type and size, and the same community will vary in size over time. The communities for casual and constant (hard-core [5]) gamers are very different, with leads to many types of analysis. For example, a hard-core gamer community could be interested in showing each player’s evolution over time and in identifying top-performing players; a casual gamer community could try to identify specific classes of players, for example players with unique combination of skills, and form groups of complementary players.

- 2. Using on-demand technology** The traditional approach to supporting online communities is shared-nothing, as companies build and operate their own infrastructure. This approach is unattractive for MMOG analytics, because the number of potential users varies greatly on both short and long time periods, and the variation is difficult to predict [10]. Instead, MMOG communities could rely on on-demand resources provisioned from a cloud only when and for how long needed.

- 3. Data management and storage** As Web 2.0 became more pervasive, MMOG operators have started to provide access to non-sensitive player data through Web 2.0 interfaces. For example, Jagex’s RuneScape provides access to player scores. However, Web 2.0 is not common among MMOGs, and most games still provide only traditional Web pages with selected player information. The presence of online APIs allows the creation of data mash-ups. For example, most operators provide information for specific player identities, but do not list the identities themselves; instead, the identities can be provided by other web sites. Thus, the main data management challenge is to be able to *use and aggregate both Web (legacy) and Web 2.0 information*. The main challenge related to data storage is to *employ data storage systems that can store MMOG-like amounts of data*.

4. Performance, scalability, and robustness While these are traditional system challenges, MMOGs raise specific challenges. The population size of successful commercial MMOGs ranges from a few thousands of players to a tens of millions of players [10]. Popular MMOGs generate massive amounts of information; for example, the database logging user actions for Everquest 2, a popular MMOG, stored over 20 new terabytes (TB) of data per year for each of the peak years of the game. Thus, the main challenges are that *the system must operate well at MMOG scale* and that *the system needs to be able to scale up and down, possibly to thousands of machines, quickly*. Other IT projects, for example the grid computing projects CERN’s Large Hadron Collider or the Sloan Digital Sky Survey, produce data orders of magnitude larger than MMOGs, but these projects are using large and pre-provisioned (expensive) computational and data infrastructure that MMOG communities cannot afford. Furthermore, the data production rate for these other projects is stable over time for spans of days or even weeks, whereas for MMOGs the daily user activity has peaks and may even change hourly.

Platform failures are common at large scale [8]; for MMOG analytics, the presence of failures is increased by traffic shaping and failures in the analytics middleware. However, not responding to failures can quickly lead to community shrinking; we have experienced before [10] *en-masse* departure of gamers, based on rumored or real system problems. The challenge is to *build a robust system*, that is, a system that fails rarely and can respond to failures quickly.

3. THE CAMEO ARCHITECTURE

In this section we present the CAMEO architecture for continuous MMOG analytics. The CAMEO architecture is built around the idea of enabling continuous MMOG analytics while using resources only when needed. To achieve this goal, it acquires and releases dynamically computational and storage resources from clouds such as Amazon Web Services.

3.1 Overview

CAMEO is our architecture for MMOG analytics in the cloud. CAMEO mines information from the Web, collects information using Web 2.0 interfaces provided by various MMOG operators and their collaborators, integrates the information into comprehensive and time-spanning MMOG datasets, analyzes the datasets, and presents application-specific results. The resources used by CAMEO can be provisioned from commercial or enterprise clouds. Through its use of computational intelligence, CAMEO addresses the main challenges faced by system designers addressing the problem of continuous analytics enabling MMOG social networks, including the understanding of the user community needs; the use of distributed and collaborative technologies such as clouds; the data management, data growth, and data storage; and building a system with high performance, scalability, and robustness. Thus, CAMEO can readily be used by a variety of MMOG communities, and may provide a good step forward in supporting other next generation, data-oriented organizations.

We have already introduced the CAMEO architecture in our previous work [6]. Except for the specific support for MMOGs and for the use of cloud computing resources, our

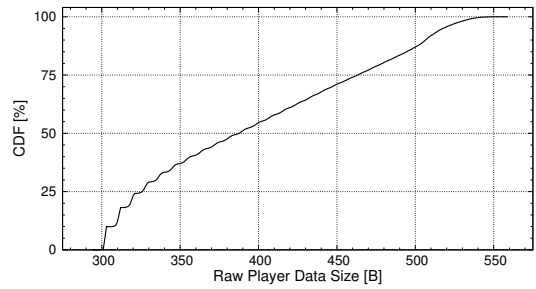


Figure 3: Cumulative distribution function of the size of raw player data. Data collected for the complete RuneScape population on Jul 4, 2010.

architecture uses a traditional approach. For example, our crawling process can be classified according to Garcia-Molina et al.’s taxonomy [3] as intra-site and/or distributed (depending on the location of the machines they run on), using static assignment (each process downloads the pages assigned at start), and operating in firewall mode (each process downloads only the pages within its partition). However, the components have unique features specific to continuous MMOG analytics; we have extended these components from our previous work to address the needs of third-party MMOG communities as explained in Section 1 and in the remainder of this section.

3.2 Addressing the Challenges of Continuous MMOG Analytics

In the remainder of this section we show, in turn, how the CAMEO architecture addresses the four challenges of continuous MMOG analytics introduced in Section 2.3.

1. Understanding User Community Needs CAMEO already supports a wide variety of specific types of analysis. First, it can analyze various pieces of player information, such as skill, experience points, and rank given by the MMOG. Second, it can process information from single or multiple data snapshots, allowing for single time point and evolution analysis. Third, there are many analyses that CAMEO already implements: ranking players according to one or more skills (which extends the functionality of the current RuneScape web site); extracting the statistical properties for the whole community for one or more skills; extracting the characteristics of the k players who improved most during a period (e.g., week); computing the (average) evolution of the Top- k best players during a period; identifying specific players with unique combined skills; etc. CAMEO also adapts to population dynamics by identifying the number of players in the MMOG community automatically (from the Web 2.0 APIs or high-score lists).

2. Using on-demand technology The CAMEO architecture is built around the use of on-demand resources, provisioned mainly from the cloud. In practice, we have used CAMEO on top of the Amazon Web Services cloud and our own, locally-installed, Eucalyptus-based [11] cloud.

One of the fine points of supporting MMOG analytics is the ability to collect and process millions of small web pages; Figure 3 shows that for RuneScape the raw player data size

ranges between 294 and 559B, with a mean of 397B and median of 388B. For this workload, the data collection time is dominated by Internet latencies. The collection time can be greatly reduced if the resource collecting data is located near the data server. There currently exist hundreds of cloud providers, and several of them, including Amazon, have multiple sites spread across the world.

Using on-demand resources can help alleviate the traffic limitations imposed by some web sites, per IP address. For example, RuneScape will limit the amount of player identifiers that a single IP address can access during an interval of a few minutes to about 20,000 players, but will not limit in any way the amount of player data; only the latter is accessed through a Web 2.0 API. We associate this behavior, which is typical for several MMOGs, with poor practice in designing the access APIs. By-passing this limitation by leasing new cloud resources when the old resources are banned can be very costly. Since the typical charging interval is an hour, using a leased resource for only a few minutes before its traffic becomes limited is impractical. Instead, CAMEO uses cloud services that allow the IP address of a leased resource to change in time, such as the Elastic IP Address (EIP) service provided by Amazon EC2. An EIP is a static IP address that is owned by the cloud provider and can be leased for use. An EIP can be attached to any instance that is currently running, replacing the old IP address it exposed outside the cloud. EIP addresses are associated with an account and not a particular instance; thus, they are just another resource to be managed by CAMEO.

3. Data management and storage CAMEO can use and aggregate both Web (legacy) and Web 2.0 information by design. Since there are no standards in MMOG player data presentation, each community will have to write its own data parsers. CAMEO already provides parsers for all the player data offered by the RuneScape operators.

To simplify data management and recording data provenance information, CAMEO stores data centrally, that is, using a single storage administrator (e.g., Amazon S3). For management, CAMEO interacts automatically with the storage administrator. Three main solutions to store data are available to CAMEO: store the data on the same machine that acquires or generates it; store the data outside the cloud; and store the data using the dedicated cloud storage services (like Amazon S3). By default, CAMEO uses the latter solution, which is centralized, reliable, scalable, and provides the highest transfer speed between storage and processing nodes for the continuous analytics workload.

4. Performance, Scalability, and Robustness The scalability and robustness challenges are eased through the use of cloud resources. Scalability-wise, cloud resource allocations are designed to scale up and down quickly; our previous evaluation of Amazon EC2 scaling capability [12] reveals that the resource allocation time is below two minutes when allocating a single resource of the type used throughout this work. Robustness-wise, in our experience with this work clouds are much more robust than grids (see the grid failure data in the Failure Trace Archive [8]). In the experiments presented in Section 4, the reference CAMEO implementation was able to scale and perform without failures the

Table 1: The resource characteristics for the instance types offered by Amazon EC2.

Resource Type	Cores (ECUs)	RAM [GB]	Archi. [bit]	Disk [GB]	Cost [\$/h]
<i>Amazon EC2.</i>					
m1.small	1 (1)	1.7	32	160	0.085
m1.large	2 (4)	7.5	64	850	0.34
m1.xlarge	4 (8)	15.0	64	1,690	0.68
(other instance types omitted)					
<i>UPB Cloud.</i>					
m1.small	1 (2)	0.2	64	4	own
m1.large	1 (2)	0.3	64	4	own
m1.xlarge	1 (4)	0.5	64	4	own
(other instance types omitted)					

largest MMOG analytics experiment to-date.

4. EXPERIMENTAL RESULTS

In this section we show evidence that our approach can be used for continuous MMOG analytics. To this end, we present in this section, in turn, the capability and performance results obtained using the reference implementation of CAMEO in practice.

4.1 Experimental Setup

To test CAMEO in practice we have created a realistic scenario, performing continuous RuneScape analytics on cloud resources. In this scenario, CAMEO performs continuous analytics on RuneScape, a popular MMOG, and uses resources leased from one of two clouds, the commercial Amazon Web Services and the Eucalyptus-based cloud installed locally. We describe in the remainder of this section our experimental setup.

CAMEO Implementation The reference CAMEO implementation was written in Python, which makes it portable for a wide range of platforms but with a lower performance than can be achieved in other programming languages, such as C. We have written RuneScape-specific web crawlers for the data collection process. For enabling S3 and EIPs support, we have used boto [2], an integrated interface to services offered by Amazon Web Services such as the Elastic Compute Cloud (EC2) and the Simple Storage Service (S3).

MMOG Case Study: RuneScape RuneScape is a popular MMOG, ranking in Aug 2008 as second by number of players in the US and European markets among MMORPG (about 7 million active players, second to World of Warcraft) [10], and number one by number of opened accounts (over 135 million). RuneScape offers detailed player identifiers through a Web (legacy) interface, and player data through a Web 2.0 interface. The player data includes a triplet (*rank, level, experiencepoints*) for each skill available to RuneScape players, and other information.

Clouds: Amazon Web Services and UPB We have used the commercial cloud Amazon Web Services. Amazon EC2 provides the computational resources to acquire and process Runescape data. The EC2 user can use any of a number of resource (*instance*) types currently available on offer, the characteristics of which are summarized in Table 1. An ECU

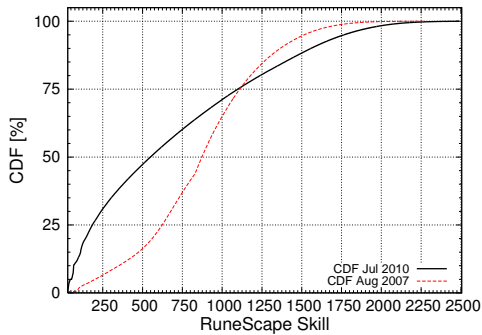


Figure 4: Overall skill level for the RuneScape population on Aug 2007 and Jul 2010.

is the equivalent CPU power of a 1.0-1.2 GHz 2007 Opteron or Xeon processor. The theoretical peak performance can be computed for different instances from the ECU definition: a 1.1 GHz 2007 Opteron can perform 4 flops per cycle at full pipeline, which means at peak performance one ECU equals 4.4 gigaflops per second (GFLOPS). Throughout the experiments conducted for this work we have used the `m1.small` instances; extending the Capacity Planning module with the ability to use multiple instance types is left as future work. We have also used Amazon S3 for storage and the Elastic IP services to alleviate traffic limitations imposed by the MMOG data provider (see Section 3.2).

A private cloud was set up at Politehnica University, Bucharest, Romania (UPB) is based on Ubuntu Enterprise Cloud, which in turns relies on the open-source cloud middleware Euca-lyptus. This cloud offers Amazon EC2-like services allowing us to run the same experiments as on the Amazon cloud. Several instance types are available (see Table 1). The generated data is stored inside the storage attached to the private cloud.

4.2 Observing 3,5 Million RuneScape Players

Using CAMEO we were able to perform the largest RuneScape data collection and analysis, to-date. We have gathered in a dataset of 1,817,211 players with level 30 or higher, based on a set of 2,899,407 player identifiers obtained manually through a painstaking gathering process, in 2007. We have repeated the process in 2010 using the automated tools provided by CAMEO. This time, a total of 3,531,478 unique player names were gathered from the highscore system and detailed data was successfully retrieved for 3,239,089 of them. Figure 4 shows the cumulative distribution function (CDF) of the overall skill level for the players in both datasets. In 2007, the maximum overall skill level was 2280, and the empirical distribution was well characterized by a (skewed) normal-like distribution. The 2010, the maximum overall skill has reached 2488, and the distribution is very different. About 90% of the players have a skill level below 1750; while few players have a level above 2280, the empirical distribution is no longer normal-like. The median level has decreased significantly, which means that more players have enough room to improve skill and thus remain longer in the system.

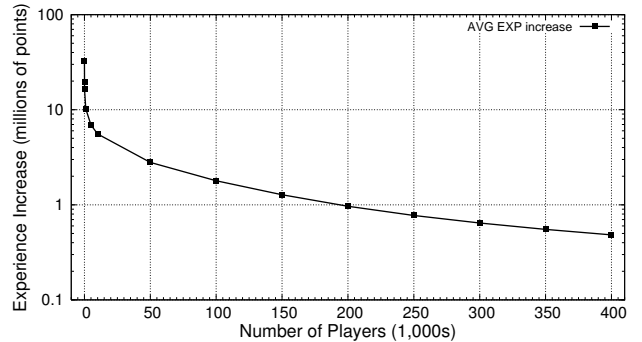


Figure 5: Average experience increase over a period of 11 days for the Top-k RuneScape players.

Table 2: The Top-15 players by increase of total experience points, over an 11 days period.

Top-15 Rank	RuneScape Overall Rank	Experience Points gained
3	101,092	37,635,063
11	364,344	23,710,650
12	357,856	23,002,277
14	127,681	20,308,110

(other Top-15 ranks omitted)

4.3 CAMEO Capability Results

Often, MMOG designers cannot account for real (emerging) gameplay, which leads to playing difficulty that is too high or too low in comparison with design expectations. Several large player communities asked and obtained from the designers of RuneScape to revert particular design changes [10]. The average experience increase is a type of analysis that can show, for a community of players, if their advancement rate is similar to the design expectation. Figure 5 shows the average experience increase for the Top-k players for various values of k. The decrease in the average is abrupt, with a Pareto-like shape; a few top players dominate the others in the amount of experience obtained. Depending on what the community wishes, the game designers may be asked to “even out” the difficulty of the game.

Advancing only one skill or a very reduced set of skills is called *skilling*, and can lead to good rewards in MMOGs, because players are often rewarded by overall skill instead of their best skill. However, players who are *skilling* (*skillers*) cannot perform the *skilling* activity alone for very long periods of time unless they find a group of players that needs the skill, because *skilling* is a highly repetitive (boring) activity. CAMEO can find *skillers* by analyzing the characteristics of individual players, and can find *active* *skillers* by observing the evolution of a player’s characteristics (stats) over time. Table 2 depicts the Top-15 players, ranked by the number of experience points gained over the 11 day period of our observation. Four out of the top fifteen players are ranked 100,000 or lower in RuneScape’s overall skill classification, and two of these four are even ranked lower than rank 350,000 by the same criterion. This gives evidence that even lowly-ranked players can quickly advance in experience, using their best skill or group of skills; these players are active *skillers* that

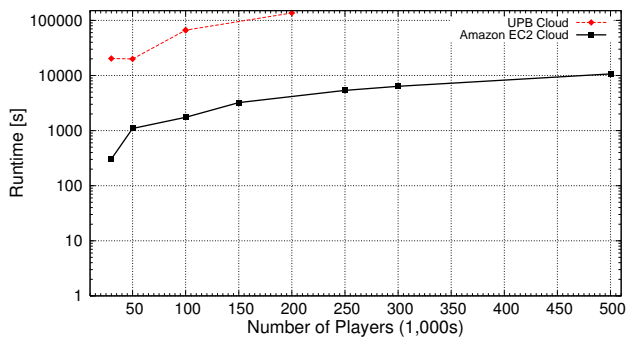


Figure 6: Performance of collecting player identifiers using the two clouds, Amazon EC2 and UPB. The vertical axis is logarithmic. Data: Jul 2010.

would help most both themselves and the group of players they will join. We leave a recommending system for group formation to future work.

4.4 CAMEO Performance Results

We have evaluated the performance of various CAMEO components alone and together in various scenarios; for brevity we present here one such scenario. In each of our performance evaluation experiments we report average results obtained after 10 repetitions of the same experiment.

Obtaining player identifiers has been in our earlier CAMEO experience the slowest part of the data analytics process. Even after automating the data collection, IP bans and the latency of crawler-server HTTP accesses have made this a multi-day operation. We have compared the use of resources leased from our local UPB Cloud and from Amazon EC2 with Elastic IPs (see Section 3.2). Figure 6 shows the time spent by each approach in acquiring identifiers for a number of players ranging from 10,000 to 500,000. We used a single node leased from each cloud. For UPB, the runtime of the collection process quickly exceeds one day, to over 140,000 seconds required to collect 200,000 player identities. In contrast, the Amazon EC2 machine can collect 500,000 player identities in just above three hours (10,690 seconds).

5. CONCLUSION AND FUTURE WORK

The growth of data produced by Massively Multiplayer Online Games (MMOGs) lead to interesting new challenges in (distributed) analytics. In this work we have presented CAMEO, an architecture for MMOG analytics. Using resources provisioned on-demand from clouds, CAMEO mines MMOG data from the Web, collects information using the Web 2.0 interfaces provided by various MMOG operators and their collaborators, integrates the information into comprehensive and time-spanning MMOG datasets, analyzes the datasets, and presents the results. CAMEO already implements a variety of MMOG-related analyses, such as ranking players according to one or more skills, extracting the characteristics of the players who improved most during a period (e.g., week), and identifying players that improved or possess a unique skill or combination of skills.

We have implemented and deployed CAMEO in practice,

and were able to perform various large-scale analysis processes on data provided by the popular MMOG RuneScape over a period of over two years. Using resources provisioned on-demand from Amazon Web Services, a commercial cloud, we have analyzed the characteristics of almost 3,000,000 players, and followed closely the progress of 500,000 players for over a week. We have also compared the performance of Amazon and local resources. Our results give evidence that cloud computing resources can be used for continuous MMOG data acquisition and analysis.

For the future, we intend to extend CAMEO towards identifying cheaters and making recommendations for players in search of an in-game group. We will also study various resource allocation policies, exploiting the cost-performance trade-off present in cloud-local resource use.

6. REFERENCES

- [1] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *WWW*, pages 131–140, 2007.
- [2] Boto. Boto: A Python interface to Amazon Web Services. <http://code.google.com/p/boto/>.
- [3] J. Cho and H. Garcia-Molina. Parallel crawlers. In *WWW*, pages 124–135, 2002.
- [4] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.
- [5] T. Fritsch, B. Voigt, and J. H. Schiller. Distribution of online hardcore player behavior: (how hardcore are you?). In *NETGAMES*, page 16, 2006.
- [6] A. Iosup. CAMEO: Continuous analytics for massively multiplayer online games on cloud resources. In *Euro-Par Workshops*, volume 6043 of *LNCS*, pages 289–299. Springer, 2009.
- [7] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *EuroSys*, pages 59–72. ACM, 2007.
- [8] D. Kondo, B. Javadi, A. Iosup, and D. Epema. The Failure Trace Archive: Enabling comparative analysis of failures in diverse distributed systems. In *CCGRID*, pages 1–10, 2010. [Online] <http://fta.inria.fr>.
- [9] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov. Irlbot: Scaling to 6 billion pages and beyond. *TWEB*, 3(3), 2009.
- [10] V. Nae, A. Iosup, S. Podlipnig, R. Prodan, D. H. J. Epema, and T. Fahringer. Efficient management of data center resources for massively multiplayer online games. In *SC*. IEEE/ACM, 2008.
- [11] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *CCGRID*, pages 124–131, 2009.
- [12] S. Ostermann, A. Iosup, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. An early performance analysis of cloud computing services for scientific computing. In *CloudComp*, volume 34 of *LNICST*, pages 1–10, 2009.
- [13] F. J. Provost and V. Kolluri. A survey of methods for scaling up inductive algorithms. *Data Min. Knowl. Discov.*, 3(2):131–169, 1999.