Chapter 12
Clouds and Continuous Analytics Enabling Social Networks for Massively Multiplayer Online Games

Alexandru Iosup and Adrian Lăscăteu

Abstract. Many of the hundreds of millions Massively Multiplayer Online Games (MMOGs) players are also involved in the social networks built around the MMOGs they play. Through these networks, these players exchange game news, advice, and expertise, and expect in return 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. In this chapter we investigate the use of CAMEO, an architecture for Continuous Analytics for Massively multiplayEr Online games on cloud resources, to support the analytics part of MMOG social networks. We present the design and implementation of CAMEO, with a focus on the cloud-related benefits and challenges. We also use CAMEO to do continuous analytics on a real MMOG community of over 5,000,000 players, thus performing the largest study of an online community, to-date.

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

Alexandru Iosup
Parallel and Distributed Systems Group,
Faculty of Electrical Engineering, Mathematics, and Computer Science,
Delft University of Technology,
Mekelweg 4, 2628CD, The Netherlands,
e-mail: `A.Iosup@tudelft.nl`

Adrian Lăscăteu
Computer Science Department,
Faculty of Automatic Control and Computer Science,
Politehnica University of Bucharest,
Spl. Independentei 311, Bucharest, Romania
e-mail: `Adrian_Lascateu@yahoo.com`

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 chapter 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.

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.

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 [2, 8, 23], or analyzing the acquired data using more scalable parallel or distributed algorithms [34, 3]; 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 [9] and Dryad [21, 41] are now supporting large-scale processing both for the general public and for companies such as Google, Facebook, Twitter, and Microsoft. In contrast to this body of previous work, which addresses a wide range of issues, in this chapter we focus on a domain-specific application, MMOG analytics, for which we design and build an integrated solution. Extending our previous work on CAMEO [15], our main contribution is four-fold. First, we introduce a number of MMOG-specific tools to better understand and support the user community needs.
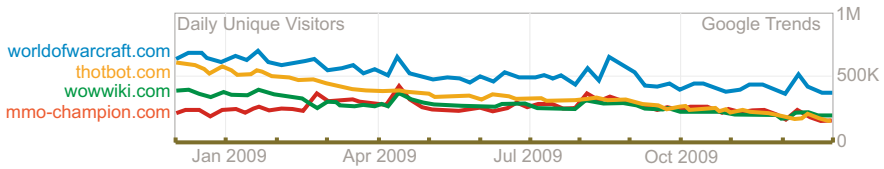
Fig. 1 Size of four online communities built around World of Warcraft. Except for world-ofwarcraft.com, the communities are not maintained by the MMOG operator.

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. Fourth, 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 free-to-use clouds.

The remainder of this chapter is structured as follows. Section 2 formulates the problem of continuous analytics for MMOGs. Section 3 contrasts this work and related research. The CAMEO architecture is introduced in Section 4. The next two sections focus on experiments using CAMEO: Section 5, which introduces the experimental setup, and Section 6, which presents the experimental results. Section 7 discusses future research directions and concludes the chapter.

2   Mission: Continuous Analytics for MMOGs

MMOGs generate data that need to be analyzed at various levels of detail and for various purposes, from high-level analysis of the number of players in a community, to the detailed analysis of the users' mouse-clicking behavior. As in our previous work on CAMEO [15], 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; the relevance of the data is application- and even community-of-users-specific. In this section we present our goal in supporting continuous analytics for MMOGs.

2.1   Goal

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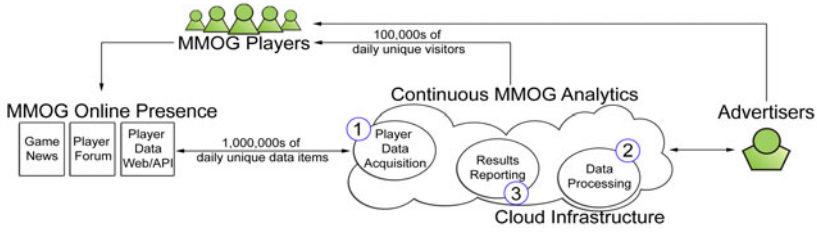
Fig. 2 The MMOG/players/advertisers/third-party communities ecosystem, for a single MMOG.

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 "by the community, for the community" reaches or even exceeds 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 community-built `thottbot.com` has had at the beginning of 2009 a size equal to `worldofwarcraft.com`'s.

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 players/MMOG operator/advertisers ecosystem depicted in Figure 2. To materialize this vision, we introduce in Section 4 CAMEO, our cloud-based continuous MMOG analytics architecture, which provides the services labeled 1 through 3 in the figure.

2.2 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.
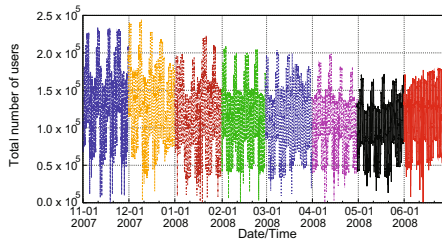
Fig. 3 The number of Active Concurrent Players in RuneScape, between Nov 1, 2007 and Jul 1, 2008. Each month of data is depicted with a different type of line. (Source: our previous work [16].)

2.2.1 Understanding User Community Needs

MMOG communities vary by type and size, and the same community will vary in size over time (see also Section 2.2.2). The communities for casual and constant (hard-core [11]) 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. The consumers of the analysis results may be the users or the community operators, which entails again different analysis needs. For example, a third-party company providing content for the game could base its decisions on the distribution of playing skills and achievements. We conclude that the main challenge for this topic is to **understand the specific and dynamic needs of each community**.

2.2.2 Enabling and Using Distributed and Collaborative 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 [29]. For example, consider the evolution over time of Zynga's popular MMOG FarmVille. FarmVille achieved over 1 million daily players after 4 days of operation, and over 10 million after 2 months, exceeding any other MMOG before and after it (until mid-2010). Eventually, FarmVille reached its peak of around 28 million daily players and 75 million monthly players in Feb 2010, nine months after launch, but it stands in July 2010 at only 16 million daily players and 62 million monthly players. For shorter time scales, consider the hourly evolution of the number of concurrent players in RuneScape, another popular MMOG (see Figure 3). The visible daily pattern in the figure indicates a strong variation in the number of players over the course of each day. Thus, for MMOGs such as FarmVille and RuneScape it is not worth provisioning resources in advance for continuous analytics. Instead, the challenge for next generation of platforms is to **provision resources on-demand**, that is, when needed and only for as long as they are needed.

2.2.3   Data Management, Data Growth, and Data Storage

Data Management: Web 2.0, Mash-ups, and Provenance As Web 2.0 has become 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, and so did the now defunct Dungeon Runners by NCsoft. 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. We have used this two-step process to collect data about an online bridge community [33]. Thus, the main challenge related to data management is to be able **to use and aggregate both Web (legacy) and Web 2.0 information**. Finally, the management of the data also involves the ability to trace back the results to their data source and production process, that is, **to record data provenance information** [27, 28].

Data Growth and Storage. The number of players across all MMOGs has increased exponentially over the past decade [38, 29]; the trend also applies to the first operational months of the popular MMOGs. Additionally, many games increase in complexity over time by adding more characteristics and abilities to the player. Thus, the amount of data produced by these games increases quickly, matching the general IT trend of the past decade to produce more data than there is (or will be) storage for them [5]. The main challenge related to data growth is to **filter out irrelevant data**. The main challenge related to data storage is to **employ data storage systems that can store the predicted amount of data**.

2.2.4   Performance, Scalability, and Robustness

Performance and Scale: MMOGs pose unique data scale and rate challenges. The population size of successful commercial MMOGs ranges from a few thousands of players to a tens of millions of players [29]. 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. Other IT projects, such as CERN's Large Hedron 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 game companies 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 (see Figure 3). The main challenge related to performance is that **the system must operate well at MMOG scale**. The main challenge related to scale is that **the system needs to be able to to scale up and down quickly**.

Robustness: MMOGs require robustness to retain and grow their communities. Platform failures are common at large scale [22]; for MMOG analytics, the presence of failures is increased by traffic shaping and failures in the analytics middleware.

However, not responding to failures in the continuous analytics process can quickly lead to community shrinking; we have experienced before [29] *en-masse* departure of gamers, based on rumored or real system problems. Thus, the main challenge related to robustness is to build a system that **is robust**, that is, it minimizes the chances of failure, but also **is able to respond to failures quickly**.

2.3   Other Applications

Building a system for continuous MMOG analytics does not benefit only MMOG communities. Instead, it enables next generation application for the gaming industry and for other domains. We describe in the remainder of this section the most important applications of our research.

Within the gaming industry, the main applications are to audit the group of companies that develop, operate, and distribute the MMOG; to understand the play patterns of users and support future investment decisions; to detect cheating and prevent game exploits; to provide user communities with data for ranking players; to broadcast gaming events; and to produce data for advertisement companies and thus increase the revenue stream for the MMOG owners.

In other domains, the applications may include studying emergent behavior in complex systems (systems theory), understanding the emergence and evolution of the contemporary society [35] (social sciences) and economy [7] (economics), uncovering the use of MMOGs as cures and coping mechanisms [37] (psychology), investigating disease spread models [4] (biology), etc.

3   Related Work

Our goal of designing a generic architecture for continuous analytics in support of social networks for MMOGs places our work at the intersection between large-scale data collection and mining, and data processing using on-demand resources. We survey in this section these two fields, in turn.

3.1   General Data Collection and Mining

Data Collection (crawling) The interest generated by Web search, promoted by companies such as Google and Yahoo, lead to many general web crawling approaches [2, 8, 26, 23]. Garcia-Molina et al. [2, 8] proposed a generic parallel crawling architecture, where the crawler operates via many concurrent crawling processes, either intra-site (parallel) or distributed, with various degrees of coordination between crawling processes (independent, dynamic or static); they also analyzed different crawling modes for static assignment. The topical (focused) crawlers, introduced and studied by Menczer et al. [26], adjust dynamically their crawl according to a rich context provided at start-up, such as topics, queries, user profiles. IRLbot [23] is a generic web crawler designed to scale web crawling to billions of

pages using limited resources. IRLbot was used to crawl over 6 billion valid HTML pages while sustaining an average download rate of over 300 MB/s and almost 2,000 pages/s during an experiment that lasted 41 days. Many of these techniques can be employed for acquiring MMOG data, but need adaptation for domain-specific data collection.

Data Mining. The general data mining community has focused on parallel or distributed analytics since the end of the 1990s [34, 25, 3]. For example, Provost and Kolluri [34] examine many basic techniques for scaling up inductive algorithms. *FacetNet* [25] is a framework for analyzing communities and their evolutions in dynamic temporal networks. Unlike traditional two-stage techniques, which separate community extraction and extraction of community evolution, the FacetNet framework combines these two tasks into a unified extraction process. Many of these techniques can be employed for MMOG analytics, but need adaptation for large-scale data processing and on-demand resource use.

   More recently, the ability to process data in many small tasks has been pioneered by the industry, with companies such as Google, Yahoo, Microsoft, Facebook, and Twitter starting general data processing frameworks. Google's MapReduce [9] and Yahoo's open-source variant Hadoop support generic data processing at large scale through a data flow model comprised of a much simplified programming abstraction and a runtime environment for this abstraction. Using only map and reduce commands, the user can execute complex data mining processes on distributed computing platforms; however, the application has to keep track of its own objects and data. Domain-specific programming languages such as Pig and Facebook's Hive hide the complexity of this programming model, but are not oriented towards processing MMOG data. Microsoft's Dryad [21] and the language DryadLINQ [41] provide together a data flow execution model, similar to MapReduce, that is most suitable for a wide range of problems from large-scale graph analysis and machine learning. Both MapReduce and Dryad scale and are fault-tolerant. These general tools can be used for continuous MMOG analytics, but would need specific adaptation for efficiently combining the data collection part with data processing.

3.2   *Data Collection and Mining Using On-Demand Resources*

With the growth of infrastructure capacity, and the trend of commercialization of computing and storage infrastructure, companies and organizations have started to use on-demand resources instead of the traditional self-owned infrastructure. Two main types of infrastructure are currently available for on-demand resource use, grids and clouds. Grids [10] are collections of resources ranging from clusters to supercomputers, shared by various organizations as a computing and data platform that is ubiquitous, uninterrupted, and has uniform user access–in this sense, similar to the power grid. Desktop grids are a flavor of grid computing in which the resources are provided by volunteers, for example through a resource manager such as BOINC. Cloud computing is an infrastructure similar to grid computing, but with

simplified access and strict service guarantees; access to resources is paid. We survey in the following the use of each of these platforms for data mining.

Grid Computing. Many grid projects, such as CERN's Large Hedron Collider and the Sloan Digital Sky Survey, have focused on data processing on on-demand resources. For example, Natarajan et al. [30] show that large-scale data mining can benefit from using on-demand resources such as grids, when the processing modules (kernels) minimize the data transfer between grid clusters, and between the compute and storage resources within a cluster. However, grids were not designed to support MMOG analytics workloads. As many of the grid data processing workloads were compute-intensive [18], that is, data processing usually took much longer than the data input and output, grids were optimized for long-running, compute-intensive jobs. Furthermore, with few exceptions, grids were not designed to crawl large amounts of small files, a typical scenario in MMOG analytics.

Cloud Computing. The increase in the number of commercial clouds led to an increase in general data processing projects that can use cloud resources. Google has designed many core tools for data storage, processing, and collaboration, such as the Google Files System [12], MapReduce [9] (provided as a service on on-demand cloud resources by Amazon), and the Google Fusion Tables [13], respectively. The Google Fusion Tables enables users to upload and share tabular data and the results of filtering and aggregating them. Thus, it enables collaboration between users, albeit it currently supports only datasets of up to 100MB. The users still have to write their processing code as queries using a subset of SQL. The *Sector storage cloud* and the *Sphere compute cloud* [14] were designed to perform high performance data mining, scaling up to wide area networks, while using a cloud-based infrastructure.

4   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.

4.1   Overview

The core dataset in CAMEO is the *snapshot*, that is, a read-only dataset which has been extracted from original data (often provided by the MMOG operator). The CAMEO architecture is built around the idea of obtaining representative snapshots, which is a classic problem of creating replicas. Similar to other cases of information replicas in distributed systems, creating exact copies of the data for analysis purposes may not be only expensive, but also unnecessary [40]. Instead of ensuring that the replicas are strongly consistent, our goal is to maintain information replicas whose difference is bounded and the bound is under the control of the analyst.
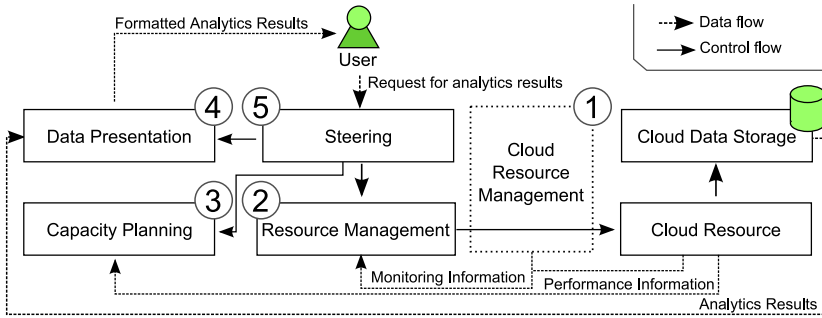
Fig. 4 The CAMEO architecture.

This goal stems from traditional work on quasi-copying [1] and on continuous consistency of information replicas with deviation in the staleness of information [40]. The former model assumes that data writing has a single possible source, which allows for looser consistency guarantees: data in a replica may differ from the original, but the current value of the replica must have been the value of the original at some (earlier) point in time, and the largest delay between an original and the replica can be controlled by the distributed system. The latter model assumes the presence of multiple data writers, and considers inconsistency along multiple axes; the goal here is to bound the absolute inconsistency between replicas.

The CAMEO architecture is designed to collect snapshots of MMOG data and to perform analytics operations on snapshots; all data storage and computation are designed to make use of on-demand resources. The five main components of the CAMEO architecture are depicted in Figure 4. The *Cloud Resource Management* component (component 1 in Figure 4) provides access to the computational and storage resources of the cloud computing environment, and is maintained by the cloud owner. The *Resource Management* component (#2) acquires and releases resources from the cloud and runs the analytics applications. It also uses the monitoring information provided by the cloud resource management and the resources as input for further management actions, such as transparent fault tolerance through application instance replication. The *Capacity Planning* component (#3) is responsible for deciding how many resources must be acquired for the analytics process. The decisions are based on the system's capability to produce results, analyzed during the course of the analytics process, and on the accuracy and cost goals of the process. The *Data Presentation* component (#4) formats and presents the results of the analytics process to the user. The *Steering* component (#5) is responsible for coordinating the analytics process. Towards this end, it takes high-level decisions, expressed through the configuration of each other's component process.

Except for the specific support for MMOGs and for the use of cloud computing resources, our architecture uses a traditional approach. For example, CAMEO's crawling can be classified according to Garcia-Molina et al.'s taxonomy [2, 8] 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 the targeted application. We describe in the remainder of this section three distinctive features of CAMEO.

4.1.1   Resource Management Mechanisms

The triggering of the analytics process depends on the nature of the application and on the system status. On the one hand, the nature of the application may allow the system analyst to design a stable analysis process such as a daily investigation of the whole community of players. On the other hand, special analysis may be required when the system is under unexpectedly heavy load, or when many players are located in the same area of the virtual world. To address this situation, we design the Resource Management component to provide two mechanisms for using cloud resources: one static and one dynamic. The *steady analytics*[1] mechanism allows running a periodic analytics operation on cloud resources. The *dynamic analytics* mechanism allows running a burst of analytics operations on cloud resources. Optimizing the allocation of resources for static analytics or for mixed static-dynamic analytics is a target for this component, but beyond the scope of this work. Similarly, the case when the cost of data transfers is significant, that is, similar or higher to the cost of the computational resources, is left for future work.

4.1.2   Steering through Snapshots of Different Size

The analytics process includes collecting the necessary information from the data source. We further *complete snapshot* a snapshot that includes data for all the players managed by the MMOG, and contrast it to a *partial snapshot*. Taking snapshots complies with the continuous analytics definition introduced in Section 2.1.

Depending on the goal of the analysis, it may be possible to obtain meaningful results through continuous analytics based on partial snapshots; for example, when the goal is to obtain statistical information about the player community it may suffice to continuously analyze a randomly chosen group of players of sufficient size. We design the Steering component to be able to perform a two-step analytics process in which first complete snapshots are taken from the system with low frequency, and partial snapshots are acquired often.

4.1.3   Controlling the Process

Taking a snapshot takes time, which depends on the performance of the cloud resources and also on the limitations set by the owners of the original data; to prevent denial-of-service attacks and to improve scalability with the number of requests, it is common for the data owners to limit the network bandwidth available for an individual resource (IP address).

---

[1] We do not use the term "static" to underline that this is a continuous process.

Assume that a single machine can acquire a new snapshot every T time units (seconds). Then, we can achieve linear scaling (to a certain degree) in the number of acquired snapshots by installing new machines; K machines can acquire K snapshots every T time units. We can then control either how many snapshots we acquire every T time units, or the minimal performance that has to be delivered by each machine to acquire exactly one snapshot every T time units.

4.2 Workflow

In this section we present the flow of data and tasks in CAMEO. Our motivating scenario focuses on a community built around the popular MMOG RuneScape (`http://runescape.com`). The goal of this community is to process player data and post online the results of MMOG analytics.

4.2.1 Data Collection

This data collection process has two parts, identifying players and using player identifiers to obtain data for each player.

There are very few MMOGs or third-party services that offer player identifiers. Instead, these identifiers can be obtained through crawling and then processing web pages made available by either MMOGs or the third-party services, such as high-score lists or forums. RuneScape offers open access to its high-score lists, but not to its forums; several third-party World of Warcraft communities offer open access to high-score lists and forums. RuneScape's high-score lists include the player identifiers of the top 2,000,000 players for over 30 ranking criteria (by each of the 24 skills in the game, by the sum of all skills, by specific in-game achievements, etc.) The crawling system often needs to access hundreds of thousands of web pages to create a comprehensive list of player identifiers; in RuneScape, each list of 2,000,000 players is accessible through web pages that present small chunks of a few tens of players each. Since crawling these pages is not supported through any API, data acquisition of player identifiers can lead to traffic limitation and thus to a serious performance bottleneck. In RuneScape, our crawler was banned for a few hours if more than about 20,000 names were acquired within a few minutes.

Player data can often be obtained individually for each player identifier through a Web 2.0 API. RuneScape and NCsoft's Dungeon Runners offer each an API for collecting player data in this way; for World of Warcraft, several third-party services offer such functionality. Most APIs are explicitly designed to allow collecting data about a single player per request, which leads to millions of requests being necessary to collect data for all players of an MMOG. During this second data collection phase, data is stored, for preservation and to enable later (re-)analysis. For data collected for each player in RuneScape include a triplet (*rank*, *level*, *experience points*) for each skill in the game. While acquiring data, CAMEO's crawlers may fail to retrieve information either because it is not offered anymore by the MMOG, or because the server to which the requests are made is overloaded. The latter errors, distinguished

by socket-level (thus, API-independent) error messages, are recorded and a second attempt is made later to gather their corresponding data.

4.2.2   Data Processing

Data processing is application-specific and applied to the original or derivative datasets. The original datasets have been collected directly by CAMEO's crawlers. The derivative datasets can be obtained (automatically) by extracting specific pieces of information from the original datasets. For example, a derivative dataset may contain only the player overall skill, which is the sum of all individual skills.

CAMEO already supports two generic types of applications to process any of these two types of datasets: single-snapshot and multi-snapshot statistical analysis of players. Single-snapshot analysis is based on a single snapshot acquired by CAMEO. Examples of results that can already be investigated by CAMEO include: ranking players according to one or more skills (which extends the functionality of the current RuneScape web site); extracting the statistical properties of a skill for the whole community, including the extraction of empirical cumulative distribution functions (CDFs); etc.

It has recently become attractive for system analysts to investigate the evolution of large-scale systems such as peer-to-peer file-sharing [36] and social [24] networks. CAMEO enables such analysis for MMOGs through multi-snapshot analysis. For this type of analysis, multiple datasets acquired by CAMEO at different times are analyzed together, and the timestamp of each dataset can influence the results of the analysis. Thus, analyzing single player evolution and the evolution of the complete community are both possible through multi-snapshot analysis. Examples of multi-snapshot analysis already implemented in CAMEO are: extracting the characteristics of players for the players who improved most during a period; computing the average evolution of the Top-k best players during a period, for arbitrary values of k; etc.

4.3   Addressing the Challenges of Continuous MMOG Analytics

In the remainder of this section we show, in turn, how the CAMEO architecture addresses each of the four challenges of continuous MMOG analytics introduced in Section 2.2.

4.3.1   Understanding User Community Needs

CAMEO addresses the main challenge of understanding the user community needs, to understand the specific and dynamic needs of each community, in two ways.

As shown in Section 2.2.2, the number of players in an MMOG is highly dynamic, both over short and long time periods. CAMEO adapts to population dynamics by identifying the number of players dynamically. In the example presented in Section 4.2, CAMEO first extracts the active player identifiers from the high-score lists, and only then collects detailed information about each player. This

mechanism can also be used to support communities with a specific focus, for example by obtaining player identifiers only for the best players featuring a specific skill or achievement and pruning out the other players.

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, as explained in Section 4.2.2, CAMEO can process information from a single or from multiple snapshots, allowing for single time point and evolution analysis. Third, there are many types of specific analysis 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. We leave as future work the task of supporting more types of analysis.

4.3.2 Enabling and Using Distributed and Collaborative 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 [31] cloud.

One of the fine points of supporting MMOG analytics is the ability to collect and process millions of small web pages. 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. While we did not implement a location-aware cloud selection mechanism, we show in our experiments in Section 6.2 that collection time can be reduced greatly through location-aware resource selection. For the same workload, data processing time can be reduced by caching the data, pipelining the use of data (for example through multi-threading), or by grouping data and storing it in larger chunks; the latter technique is employed by Google's File System [12].

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. Bypassing 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

Table 1 Comparison of storage solutions. The aspects are rated through a 7-level Likert scale, from "– – –" to "X" to "+ + +", where the "–", "X", and "+" signs denote negative, neutral, and positive appreciation, respectively.

	Centralization	Traffic Speed	Reliable	Small Load	Medium load	Large load
Cloud machine	–	+ + +	+ + +	+ + +	+ +	–
Local storage	+ +	– – –	– –	+ + +	+ + +	+ +
Cloud storage	+ + +	+ +	+ + +	+ + +	+ + +	+ + +

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.

4.3.3   Data Management, Data Growth, and Data 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. Transferring data in/out of the cloud incurs additional costs and has much lower performance than intra-cloud data transfers. The dedicated cloud storage services are much more reliable than any machine in the cloud. We analyze in the following each solution; Table 1 compares the solutions for three load levels, small, medium, and large. Storing data on the same machine means that the machine gathering the data preserves the data for further local processing, or for being used by another machine; this forces the data processing application to keep track of data location, and effectively breaks the CAMEO assumption that data are stored under a central administrator. This solution also has limited storage capacity (per machine) and limited reliability. Storing data outside the cloud has the downside of low data transfer performance, with data transfers into the cloud being particularly slow. The reliability of the storage solution is another issue for this solution. Storing the data using the dedicated cloud services is a solution that keeps the data inside the cloud as much as possible, and transfers outside the cloud only the results that are needed. Cloud storage is centralized, more reliable than storing data on a cloud machine, and faster than transferring data from outside the cloud to the processing machine; however, cloud storage can be expensive.

CAMEO stores the data using the dedicated cloud storage services, by default. Thus, the ability of CAMEO to store MMOG data matches the ability of the cloud

infrastructure. For example, Amazon S3 is used as the storage solution in the experiments using Amazon EC2 for computation. In S3, users store data in labeled "buckets" (for our purpose, the equivalent of directories, with provenance support). We have already stored several terabytes of data using Amazon S3.

CAMEO can filter out irrelevant data to the extent by which the MMOG API supports this feature, for example by pruning out players (described in Section 4.3.1) or by allowing the acquisition of specific data fields for each player.

4.3.4 Performance, Scalability, and Robustness

The performance of CAMEO is upper-bounded by the performance of the leased cloud resources, and by the location of the data.

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 evaluations of Amazon EC2 scaling capability [32, 20] reveal 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 and with benchmarking four clouds [19], clouds are much more robust than grids (see the grid failure data in the Failure Trace Archive [22]).

In the experiments presented in Section 6.1, the reference CAMEO implementation was able to scale and perform without failures the largest MMOG analytics experiment to-date.

5 Experimental Setup

To test CAMEO in practice we follow the scenario introduced in Section 4.2. 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 the experimental setup we have used.

5.1 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 [6], an integrated interface to services offered by Amazon Web Services such as the Elastic Compute Cloud (EC2) and the Simple Storage Service (S3).

We have also used two additional tools for debugging purposes. Hybridfox is a Firefox extension for managing an Amazon EC2 account. With Hybridfox, launching, stopping, connecting to, and monitoring machines leased from clouds can be done via a graphical interface as opposed to standard command line interface.

```
0    Herccrazy    4306  2321  325506318    2127  99    42880737      24922 99
     15801687    24171 99    19000663      5746  99    34673693      5755  99
     20804786    13596 99    13057112     26763 99    13892161     47528 99
     13270010    54709 99    13162562     13561 99    14118083    178680 82
     2568328     33054 99    13052685      8879  99    13052970     15712 90
     5365614     46742 85     3403281     17862 90     5366501      16882 85
     3499380     10024 99    13167150      7451  99    13155100     11879 93
     7390556       214  99    23932428     57488 82     2421684      7746  90
     5361562      6636  99    13069715    106769 40    37870 14673 1830   -1
     -1     -1     -1    207371            1276  -1     -1    20899 1906 6645
     2424   32885 1138  69240 1544   -1     -1
```

Fig. 5 Player data example.

Table 2 The resource characteristics for the instance types offered by Amazon EC2.

Resource Type	Cores (ECUs)	RAM [GB]	Architecture [bit]	I/O Performance	Disk [GB]	Cost [$/h]
m1.small	1 (1)	1.7	32	Med	160	0.085
m1.large	2 (4)	7.5	64	High	850	0.34
m1.xlarge	4 (8)	15.0	64	High	1,690	0.68
c1.medium	2 (5)	1.7	32	Med	350	0.17
c1.xlarge	8 (20)	7.0	64	High	1,690	0.68
...						

Hybridfox can also be used with the private cloud at UPB. We have also used the S3 Organizer as a visual interface for managing the contents of the S3 bucket.

5.2   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), and number one by number of opened accounts (over 135 million). RuneScape offers detailed player data through a Web 2.0 interface. Figure 5 shows the data gathered for one player. Besides the index (0) and the name ("Herccrazy"), the data includes a triplet $(rank, level, experience points)$ for each skill available to RuneScape players, and other information (see Section 4.2.1).

5.3   Cloud Infrastructure

We have used in our experimental setup two cloud platforms: Amazon EC2 and a Eucalyptus-based private cloud. We describe the two platforms in the following.

Amazon Web Services 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 2.

An ECU 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 giga float operations 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 4.3.2).

Table 3 The resource characteristics for the instance types offered by the UPB cloud.

Resource Type	Compute Units	RAM [GB]	Architecture [bit]	Disk [GB]
m1.small	1	192	64	4
m1.large	1	256	64	4
m1.xlarge	2	512	64	4
c1.medium	2	1024	64	4
c1.xlarge	4	2048	64	4

UPB 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 Eucalyptus. This cloud offers Amazon EC2-like services allowing us to run the same experiments as on the Amazon cloud. Five instance types are available with characteristics detailed in Table 3. One compute unit in this case is the equivalent of an Intel Xeon processor at 2 Ghz. Although the private cloud is able to support elastic IP assignment we did not use this service because we wanted to conduct an experiment without EIP support for comparison purposes. Generated data is stored inside the storage attached to the private cloud.

6 Experimental Results

Using CAMEO, we have taken and analyzed several complete snapshots of the state of RuneScape over a period of one and a half years. We have also also taken partial snapshots of the state of RuneScape in quick succession, which enabled us to study the short-term dynamics of the RuneScape community. In this section we show evidence of CAMEO being able to fulfill in practice the challenges of continuous MMOG analytics. To this end, we present sample results rather than perform an in-depth performance evaluation or a detailed analysis of the RuneScape community. For an in-depth performance evaluation we refer to our previous work [42].
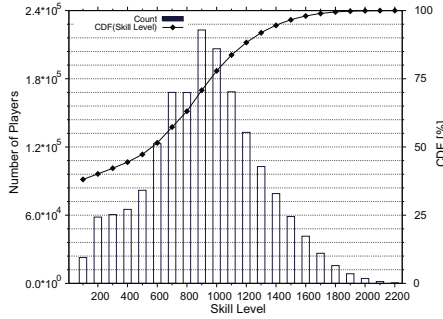
Fig. 6 Pareto graph, that is, combined PDF (left vertical axis) and CDF (right vertical axis) depiction of the skill level of the RuneScape player population. Each bar represents a range of 100 levels. CDF stands for cumulative distribution function; $CDF(x)$ is the total number of players with skill level up to and including x. Note that the left vertical axis is not linear. See text for why the CDF of the skill level does not start at 0%.

6.1   Understanding User Community Needs

We exemplify in this section the ability of the reference CAMEO implementation to adapt to the specific and dynamic user community needs.

Using CAMEO, we analyzed the skill level of millions of RuneScape players, which shows evidence that CAMEO can be used for measurements several orders of magnitude larger than the previous state-of-the-art [39]. CAMEO collected in August 2008 official skill level data for 2,899,407 players. We have repeated the experiment in 2009 and 2010; for these measurements, the population size has increased to over 3,000,000 active players, despite a high rate of player abandonment (more than 25% of the players identified in 2008 are not present in the 2010 experiments). The success of this experiment across multiple years also demonstrates the ability of CAMEO to adapt to long-term changes in the population size.

CAMEO automatically analyzes the player skill distribution; from the 2008 dataset, 1,817,211 (over 60%) players had an overall skill level above 100; the maximum overall skill level was 2280 in 2008. The values for players with skill level below 100 include application-specific noise (mostly starting players) and are therefore polluted. Thus, we present here only data for all players with skill above 100, and for a single measurement. Figure 6 depicts the overall skill level of RuneScape players, with bins of 100 levels. The number of players per bin is well characterized by a skewed normal-like distribution; the majority of the players are of average skill or below, the most populated skill level bins are those corresponding to the middle skill level, and the number of high-level players is significant. We have explored the implications of the overall skill level distribution in our previous work on automatic content generation [16, 17].
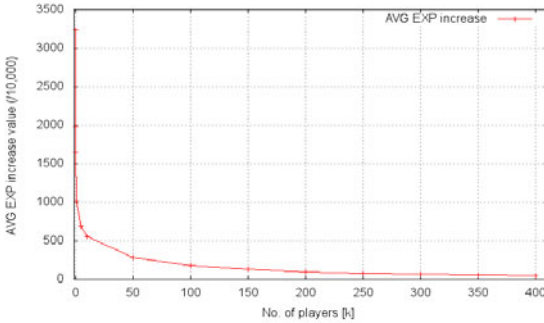
Fig. 7 Average experience increase over a period of 11 days for the Top-k RuneScape players. The values of k are in in 1,000, e.g., $k = 400$ refers to 400,000 players.

Table 4 The Top-15 players by increase of total experience points, over an 11-day period.

Top-15 Rank	RuneScape Overall Rank	Experience Points gained
1	272	49,880,094
2	28	39,233,769
3	101,092	37,635,063
...		
11	364,344	23,710,650
12	357,856	23,002,277
13	65	21,061,414
14	127,681	20,308,110
15	53,922	20,181,985

We show in the following two types of analysis enabled by CAMEO; both use multiple datasets and follow the evolution RuneScape players over a period of 11 days. We exemplify with a use scenario each of the two types.

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 [29]. 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 7 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 activity. Thus, identifying active skillers can be used to form groups of highly-effective, yet lowly-ranked, players and at the same time prevent skillers from becoming bored and leaving the game. 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 4 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 would help most both themselves and the group of players they will join.

6.2   Enabling and Using Distributed and Collaborative Technology

To demonstrate the capability of CAMEO to perform both dynamic and steady analytics, and to monitor the process, we show in Figure 8 the evolution of the cumulative number of consumed CPU hours over time. The dynamic analytics are based on uneven bursts of activity, of which the burst during March 10 is the most prominent. The steady analytics part of the experiments reveals an even use of resources over time, with the steps indicating a new work cycle.

To give a first estimation on the cost of continuous MMOG analytics, we use a simple analytics process that acquires partial snapshots and only browses the data in memory during the processing phase. Figure 9 shows the total cost incurred by the continuous analytics process over the course of one month. For this simple analysis process the cost is below $500 per month. It is not our intention to argue that the cost of continuous analytics for an MMOG can be this low; much more complex
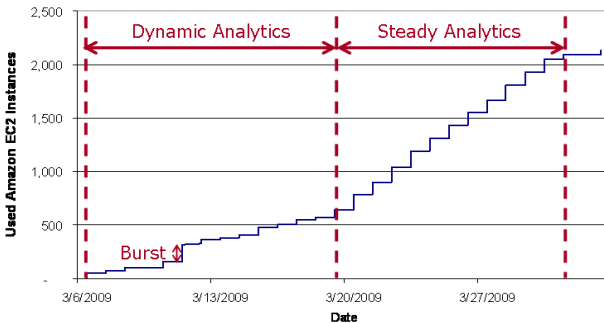

Fig. 8 Resource consumption in the two analytics modes: dynamic and static.

Billing Statement: April 1, 2009		
Billing Cycle for this Report: March 1 - March 31, 2009		
		Expand All | Collapse All
Rate	**Usage**	**Totals**
⊟ **Amazon Elastic Compute Cloud**		
View/Edit Service		
Amazon EC2 running Linux/UNIX		
$0.10 per Small Instance (m1.small) instance-hour (or partial hour)	2,097 Hrs	209.70
Amazon EC2 Bandwidth		
$0.100 per GB Internet Data Transfer - all data transfer into Amazon EC2	611.005 GB	61.10
$0.170 per GB Internet Data Transfer - first 10 TB / month data transfer out of Amazon EC2	507.121 GB	86.21
Taxes		67.83
Charges due on April 1, 2009+		**424.85**

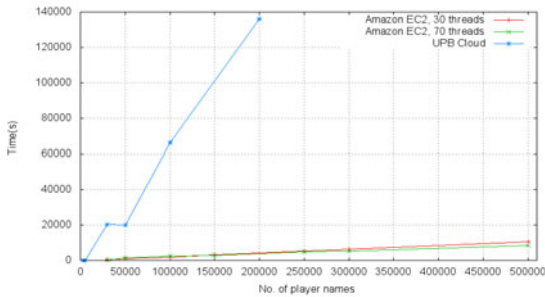**Fig. 9** Putting a cost on continuous analytics for MMOGs.

Fig. 10 Performance of collecting player identifiers for three configurations, Amazon EC2 with 30 and 70 collection threads, and UPB cloud with 30 collection threads.

analytics taking many more computational hours are performed for any of the applications presented in Sections 2.3 and 6.1.

To demonstrate the benefits of using EIPs, we collect player identifiers from the RuneScape servers located in the US (West Coast) using a single cloud machine. We use, in turn, the closely located resources of Amazon and the remote resources of UPB; besides a different location, UPB does not use EIPs. We also vary the number of threads for the faster Amazon location, which allows us to estimate the benefit of using EIPs in comparison with using traditional parallelism through multithreading. Figure 10 shows the time spent by each approach in acquiring player identifiers for a number of players ranging from 10,000 to 500,000. As expected, for UPB the time needed to collect the same amount of player names without EIP support increases quickly with the number of player identifiers, because of the repeated bans. In contrast, the Amazon EC2 uses the EIP mechanism to alleviate traffic limitations imposed by the MMOG data provider (see Section 4.3.2). This way, a single machine can be used effectively to collect the player identities in a reasonable amount of time. The number of threads does not affect the performance similarly to the number of EIPs, and may lead to data collection loss as many concurrent requests can lead to the requesting IP address being banned (not shown here).

7 Conclusion and Future Work

The expanding world of Massively Multiplayer Online Games (MMOGs) fosters important derivative online applications and raises interesting new challenges to the distributed computing community. In this work we present CAMEO, an architecture for MMOG analytics in the cloud that 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 hides the complexity of detailed resource allocation and use from the user, and operates on top of clouds to provision resources on-demand, that is, only when and for long they are needed.

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. Our results give evidence that cloud computing resources can be used for continuous MMOG data acquisition and analysis. Finally, we have provided a first cost estimation for the continuous MMOG analytics process.

The analysis capabilities already implemented in CAMEO already cover a wide range of community uses. However, we identify as future research directions the use of these results to find and reward good players, to single out potential cheaters, to identify good matches between players and to make grouping recommendations based on them, etc. Each of these uses of MMOG analytics comes with distinct design, implementation, and testing challenges, but will potentially impact the lives of a large number of people.

Acknowledgments

References

1. Alonso, R., Barbará, D., Garcia-Molina, H.: Data caching issues in an information retrieval system. ACM Trans. Database Syst. 15(3), 359–384 (1990)
2. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Internet Technol. 1(1), 2–43 (2001),
   http://doi.acm.org/10.1145/383034.383035

3. Bayardo, R.J., Ma, Y., Srikant, R.: Scaling up all pairs similarity search. In: International Conference on the World Wide Web (WWW), pp. 131–140 (2007)
4. BBC NEws, Virtual game is a 'disease model'. News Item (2009),
   http://news.bbc.co.uk/2/hi/6951918.stm
5. Berman, F.: Got data?: a guide to data preservation in the information age. Commun. ACM 51(12), 50–56 (2008)
6. Boto: Boto - a python interface to amazon web services,
   http://code.google.com/p/boto/
7. Castronova, E.: On virtual economies. Game Studies 3(2) (2003)
8. Cho, J., Garcia-Molina, H.: Parallel crawlers. In: International Conference on the World Wide Web (WWW), pp. 124–135 (2002)
9. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008)
10. Foster, I.T., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. Int'l. J. of High Performance Computing Applications 15(3), 200–222 (2001)
11. Fritsch, T., Voigt, B., Schiller, J.H.: Distribution of online hardcore player behavior (how hardcore are you?). In: Workshop on Network and System Support for Games (NETGAMES), p. 16 (2006)
12. Ghemawat, S., Gobioff, H., Leung, S.T.: The Google File System. SIGOPS Oper. Syst. Rev. 37(5), 29–43 (2003)
13. Gonzalez, H., Halevy, A.Y., Jensen, C.S., Langen, A., Madhavan, J., Shapley, R., Shen, W., Goldberg-Kidon, J.: Google Fusion Tables: web-centered data management and collaboration. In: SIGMOD 2010: Proceedings of the 2010 International Conference on Management of Data, pp. 1061–1066. ACM, New York (2010)
14. Grossman, R.L., Gu, Y.: Data mining using high performance data clouds: Experimental studies using sector and sphere. CoRR (2008),
    http://arxiv.org/abs/0808.3019
15. Iosup, A.: CAMEO: Continuous analytics for massively multiplayer online games on cloud resources. In: Lin, H.-X., Alexander, M., Forsell, M., Knüpfer, A., Prodan, R., Sousa, L., Streit, A. (eds.) Euro-Par 2009. LNCS, vol. 6043, pp. 289–299. Springer, Heidelberg (2010)
16. Iosup, A.: POGGI: Puzzle-based Online Games on Grid Infrastructures. In: Sips, H., Epema, D., Lin, H.-X. (eds.) Euro-Par 2009. LNCS, vol. 5704, pp. 390–403. Springer, Heidelberg (2009)
17. Iosup, A.: POGGI: generating puzzle instances for online games on grid infrastructures. Concurrency and Computation: Practice and Experience (2010) (accepted April 2010, in print), doi: 10.1002/cpe.1638
18. Iosup, A., Dumitrescu, C., Epema, D.H.J., Li, H., Wolters, L.: How are real grids used? the analysis of four grid traces and its implications. In: IEEE/ACM International Conference on Grid Computing (GRID), pp. 262–269. IEEE, Los Alamitos (2006)
19. Iosup, A., Ostermann, S., Yigitbasi, N., Prodan, R., Fahringer, T., Epema, D.: Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Trans. on Parallel and Distrib. Sys. (2010) (accepted September 2010, in print)
20. Iosup, A., Yigitbasi, N., Epema, D.: On the performance variability of production cloud services. Tech.Report, TU Delft (2010),
    pds.twi.tudelft.nl/reports/2010/PDS-2010-002.pdf

21. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: distributed data-parallel programs from sequential building blocks. In: EuroSys, pp. 59–72. ACM, New York (2007)

22. Kondo, D., Javadi, B., Iosup, A., Epema, D.: The Failure Trace Archive: Enabling comparative analysis of failures in diverse distributed systems. In: IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID), pp. 398–407 (2010), Archive data available: `http://fta.inria.fr`

23. Lee, H.T., Leonard, D., Wang, X., Loguinov, D.: Irlbot: Scaling to 6 billion pages and beyond. ACM Transactions on the Web (TWEB) 3(3) (2009)

24. Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic evolution of social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 462–470. ACM, New York (2008)

25. Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: Huai, J., Chen, R., Hon, H.W., Liu, Y., Ma, W.Y., Tomkins, A., Zhang, X. (eds.) International Conference on the World Wide Web (WWW), pp. 685–694. ACM, New York (2008)

26. Menczer, F., Pant, G., Srinivasan, P.: Topical web crawlers: Evaluating adaptive algorithms. ACM Transactions on Internet Technology (TOIT) 4(4), 378–419 (2004)

27. Miles, S., Groth, P.T., Deelman, E., Vahi, K., Mehta, G., Moreau, L.: Provenance: The bridge between experiments and data. Computing in Science and Engineering 10(3), 38–46 (2008)

28. Muniswamy-Reddy, K.K., Macko, P., Seltzer, M.I.: Making a cloud provenance-aware. In: Workshop on the Theory and Practice of Provenance. USENIX (2009)

29. Nae, V., Iosup, A., Podlipnig, S., Prodan, R., Epema, D.H.J., Fahringer, T.: Efficient management of data center resources for massively multiplayer online games. In: ACM/IEEE Conference on High Performance Networking and Computing (SC). IEEE/ACM (2008)

30. Natarajan, R., Sion, R., Phan, T.: A grid-based approach for enterprise-scale data mining. Future Generation Comp. Syst. 23(1), 48–54 (2007)

31. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID), pp. 124–131 (2009)

32. Ostermann, S., Iosup, A., Yigitbasi, M.N., Prodan, R., Fahringer, T., Epema, D.: An early performance analysis of cloud computing services for scientific computing. In: CloudComp. LNICST, vol. 34, pp. 1–10. Springer, Heidelberg (2009)

33. Posea, V., Balint, M., Dimitriu, A., Iosup, A.: An analysis of the BBO Fans online social gaming community. In: RoEduNet International Conference (RoEduNet), pp. 218–223. IEEE, Los Alamitos (2010)

34. Provost, F.J., Kolluri, V.: A survey of methods for scaling up inductive algorithms. Data Min. Knowl. Discov. 3(2), 131–169 (1999)

35. Steinkuehler, C., Williams, D.: Where everybody knows your (screen) name: Online games as "third places". In: DIGRA Conf. (2005)

36. Stutzbach, D., Rejaie, R., Sen, S.: Characterizing unstructured overlay topologies in modern p2p file-sharing systems. IEEE/ACM Trans. Netw. 16(2), 267–280 (2008)

37. Williams, D., Yee, N., Caplan, S.: Who plays, how much, and why? debunking the stereotypical gamer profile. Journal of Computer-Mediated Communication 13(4), 993–1018 (2008)

38. Woodcock, B.S.: An analysis of mmog subscription growth. Online Report (2006), `http://www.mmogchart.com` (November 2008)

39. Yee, N.: The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. Presence 15(3), 309–329 (2006)
40. Yu, H., Vahdat, A.: Efficient numerical error bounding for replicated network services. In: VLDB 2000: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 123–133. Morgan Kaufmann Publishers Inc., San Francisco (2000)
41. Yu, Y., Isard, M., Fetterly, D., Budiu, M., Erlingsson, Ú., Gunda, P.K., Currey, J.: Dryadlinq: A system for general-purpose distributed data-parallel computing using a high-level language. In: OSDI, pp. 1–14. USENIX (2008)
42. Iosup, A., Lascateu, A., Tapus, N.: CAMEO: Enabling Social Networks for Massively Multiplayer Online Games through Continuous Analytics and Cloud Computing. In: Workshop on Network and System Support for Games (NETGAMES), vol. 7, pp. 1–6. IEEE Press, Piscataway (2010)