Déjà vu - Predicting the number of players in online games through normalization of historical data

Jon-Erik Tyvand
and Kyrre Begnum
and Hugo Lewi Hammer
Oslo University College
Pb. 160 St. Olavs Plass

***Abstract*—A key factor to delivering a good online gaming experience is to have sufficient server resources relative to the number of players online. The standard approach of overprovisioning capacity and let large amounts of resources stand idle most of the time does not meet todays expectations of economical and environmental considerations. In this work, we present a simple profiling technique which allows effective prediction of the number of players in the next 24 hour cycle based on both weekend and workday data. The methodology enables planning ahead so that resources can be scaled to the sufficient amount, thereby reducing cost and power consumption.**

I. Introduction

For over a decade, online gaming has become one of the most successfull businesses in the Internet era. Facilitated by an increasing coverage of broadband, virtually all homes with internet access are able to participate. With higher network speeds and better quality of service, more content can be delivered on time, allowing for richer gaming experiences.

In this work, we are concerned with managing the server resources which are responsible for providing a good gaming experience. Online games are dependant not only on good and reliable network speeds, but also on sufficient server resources. The driving factor of a servers resource use is the number of players. However, it is not economically nor environmentally feasable to overprovision resources based on a estimate of the maximum number of players and let them stand idle for most of the time. System administrators face the challenge of being able to adjust the resources based on the actual requirements in order to minimize cost and waste. This cannot be done on-the-fly. Players who want to play will expect the resources to be ready, so starting servers in retrospect will not work. Instead, in order to achieve this goal, one needs the ability to make accurate predictions about how much activity an online game will experience at any time, so that it is indeed possible to make adjustments and plan ahead.

This paper makes use of the knowledge that human behavior follows a 24 hour cycle to create a profile based on past behavior which is able to make fast predictions about how the next 24 hour cycle will look like using just a single observation. We want to test this on real-life data from some of the most popular games played online in order to get a realistic evaluation of our approach. The proposed method is able to use both weekend data and workdays to build the profile, optimizing the learning period.

II. Background

A. Online Gaming as a Service

Online gaming today places a high demand on servers that are running the service. Gamers demands enough resources from the game servers so that they get an expected gaming experience. The industry is forced to follow these demands in order to keep their customers. Online gaming has become such a large service that it takes many data centers with multiple servers in order to manage just one game [23].

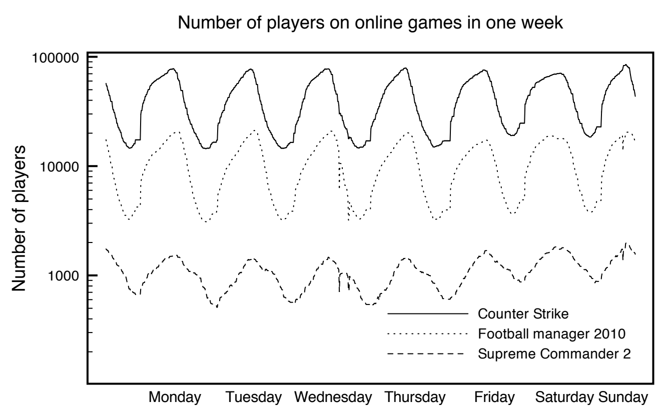

Fig. 1. An example week from three games. Even though the games differ in the number of players, they exhibit a similar profile, when displayed on a logarithmic axis.

The Figure 1 shows how the number of players vary for three different games. Notice, that when using a logarithmic Y-axis, the overall structure of the three games becomes much similar even though the number of players is vastly different. This suggests that the overall periodic behavior is similar accross games and independent from the number of players.

Most research is considered with evaluating the effect of network quality on online gamers. [1] [2] [6] [16] [28] In most of these studies, players have been graded either subjectively or objectively in controlled network environments. Most of these studies have the same conclusion, network quality has a great effect in gaming experience. [29] [35] [36]

In general, perception of a players decision wether to quit or continue playing a game is difficult to study because there are so many factors that affect human decisions. A study in Computing and Network Security Laboratory at National Taiwan University asks the question, how sensitive are online gamers to network quality? [7] This study has shown that game playing time is strongly related to network QoS and is a potential indicator of user satisfaction. This indicates the importance of QoS in online gaming.

Network quality is obviously dependent on the entire network infrastructure. From a business perspective, however, the company responsible for hosting an online game may only address the local server and network resources. They have little influence over the entire infrastructure. The effect on players satisfaction when there are insufficient resources has not been studiet in equal depth. However, in other service fields such as web and application servers, service performance and resource management has been studied extensively, especially after the advent of green computing.

III. Green Computing

Green computing has been a hot topic the last few years. Power consumption, waste disposal and environmental effects of production has become more apparent over the years. One of the reasons for environmental change is the growth in IT systems, so one of the main goals of green computing is to use the computer resources as efficiently as possible, while maintaining or eve increasing the performance [15].

Over the last few years, the technologies to reduce power consumption have improved, especially on laptops where the battery time has improved drastically [31]. However, over the same time period the overall power consumption has increased because of the increase in IT systems [4]. The focus on reducing the consumption of power has been on processing power [22]. Processors consume most of the power in the majority of computers or servers [25], by following the well-known Moore's law [27] the power consumption by processors has been reduces over time [12].

A. Power Consumption in Data Centers

With the increasing need of power to run server and the cooling systems, the main focus of green computing lies in methods to reduce the power demands in data centers [22]. The wasteful energy consumption of a data center can easily account for more than half of both the electricity bill and the corporate carbon footprint in the most IT organizations [15].

A recent Internet Data Center report estimated the worldwide cost on enterprise power consumption exceeds $30 billion in year 2008 and is likely to surpass the worldwide spendings on new server hardware in 2008. The rated power consumptions of servers has increased by 10 times over the past ten years [30]. The huge amount of power consumption calls for the need of new energy efficient methods.

In Green Computing the IT energy management is the analysis and management of energy demand within the information technology arena. Global IT energy demand accounts for approximately 2% of global energy demand, this is approximately at the same level as aviation. [13] IT equipment can account for 25% of a modern office buildings energy cost. [26] The main sources of IT energy consumption are PCs and Monitors, accounting for 39% of energy use, followed by data centers and servers, accounting for 23% of energy use. [21]

However, the challenges of today do not necessarily lie in creating new power efficient technologies, but using knowledge and technology that we already have [15]. The tradition in system administration is to overprovision server power and subsequently letting it sit idle for most of its life. With the green computing paradigm, it is expected to only use the *required* amount of power and reduce waste. Resource management through server scaling has become a core compontent of modern data centers and services.

B. Resource Management

Recently, the focus in computer systems has shifted from purely performance to good performance at lowest possible power consumption [18]. A study performed at Intel Research shows power control algorithms that attempt to reduce power consumption of a resource by taking advantage of available low power states. This study compared proactive power control algorithms to reactive power control algorithms. The study showed that proactive algorithms can provide some added benefits at moderate traffic loads.

Researchers try to find effective solutions to make data centers reduce power consumption while keeping the desired quality of service [24]. Researchers at IBM China Research Laboratory, McGill University and University of New Mexico have developed a Green Cloud architecture, which aims to reduce data center power consumption, while guarantee the performance from users perspective. They have verified the efficiency and effectiveness of the Green Cloud architecture by taking an online real-time game, Tremulous, as a VM application. The evaluation results show that they saved up to 27% of the energy when applying Green Cloud architecture to this game.

The need for improved power management in data centers is becoming essential, one of the most promising topics on this is improving Autonomic power management systems [20]. Autonomic power management is defined as a management system that to a certain degree can manage itself given a set of objectives from an administrators [19].

Kandasamy et al[17] propose a control mechanism on the processor to optimize expected behavior using a mathematical model based on a limited look-ahead prediction. This model can be applied to reduce power consumption in processors. Sharma et al [32] have implemented algorithms inside the Linux kernel that scale voltage dynamically in QoS-enabled web-servers, to minimize energy consumption without violating any quality of service constraints.

A well-studied technique for increasing data center energy efficiency is dynamic server consolidation. This method migrates application workload onto a minimal number of servers and putting the unused servers into a low-power state.[5] [10] [8] [3] [9]. However, the primary challenge of this technique is the decision-making aspects, which in enterprise data centers can be very complex. Interesting work on this challenge has been done in [14].

IV. Algorithm

The main goal of the methodology is to make type of predictions which are useful for building an understanding of the load profile for an online game. The a property of this prediction should be that it minimizes the risk for predicting fewer players than what turns out to be the reality. It should therefore attempt to be slighly higher rather lower.

A. Training phase

At the heart of the algorithm is the notion of a 24 hour profile which is generated based on a learning period T. A profile p is a series of values of the same size as a the cycle period.

$$p = \{x_1, x_2, ..., x_n\}$$

Where n is the number of values in the series. For example, if the cycle is 24 hours and data is collected every five minutes, n would equal 288. During the training period, the data from all days are collected. After the training period, a single point in time b is selected. For each day in the training period:

$$d_t = \{v_{1t}, v_{2t}, ..., v_{nt}\}, t = [1, T]$$

the value at position b is collected into a set B:

$$B = \{v_{b1}, v_{b2}, ..., v_{bt}\}$$

Next, a percentile value N is calulated from the values in B. a denotes the percentile in question, for example the 85% percentile. For each day d_t in the training set, a normalized version of the day d_t^N is created, where the values are normalized relative to the value N. This is achieved by calculating a factor F_t:

$$F_t = \frac{N}{v_{bt}}$$

All values in d_t are multiplied with their respective F_t in order to create d_t^N.

Finally, p is created by collecting all normalized days and calculating the value x at each position based on the a percentile from all normalized values on that position:

$$x_i = a\ percentile(v_{i1}^N, v_{i2}^N, ..., v_{in}^N)$$

As an example: consider a data set consisting of 30 training days ($T = 30$). Data has been collected every five munites, so there are 288 data items in each day ($n = 288$). The datapoint $b = 108$ corresponds to 9 am in the morning. All data values are collected at 108 and a 85% ($a = 85$) percentile is calculated. This means that from all days we now have a single data value N at position b. The normalization process is about making all days pass through this particular point, either by lifting them or lowering them so that at 108 they all have the value N and the rest of the data items accordingly. This is achieved by calculating a factor F relative to each day. From these normalized days we again pick the 85% percentile at each point, giving us a complete day which would pass through N and represent an above average line.

Through this process, the profile becomes more of a standard representation of an expected curve, and not necesarily a representation of an expected number of players. This becomes more clear when we explain how the algorithm uses the profile in order to make predictions next.

B. Testing phase

Once a profile p is established, we can use the profile on all subsequent days, either in a testing set or live as new data comes in. The actual prediction is a normalized version of p relative to, again, a point b, but this time in the new and observed data. Say, that at 9 am a live value l_{108} is measured. We then calculate the factor F which would make the profile pass through the same value at the same point. The factor F is then applied to the profile p so we get a p_F, which is the predicted continuation of what has been observed.

Notice, that once the training is over, one only needs a single new datapoint to make a prediction. We take the curve, as it was designed by the profile and adjust it so it alligns with the current observation. The rest of that curve is then the prediction.

The training process can be repeated as more data is available in order to update the profile.

V. Results

In order to test the usefullness of the algorithm, it was implemented and tested on real data gathered from Steampowered.com. Steam is a well known platform for distribution of games and social gameplay. They release data on the number of players to the public. Data from the site steampowered.com

[33]was collected over a period of six months. The algorithm was tested on three games, each belonging to a different type of genre. The game ”Counter Strike: Source” is a reowned and popular online game in the First-Person Shooter genre. As can be seen from the Figure X, the game has a consistently high number of players even though it has existed for a long period of time. The second game was ”Football Manager 2010”, a Strategy / Simulation game which allows public ranking of scores. Finally, ”Supreme Commander 2” is a game in the Real-Time Strategy genre. All three games represent a different playing style but also an online component. They also experience signifficantly different number of players, as can be seen from Figure X. Using these three games we wanted to see if the overall amount of players affected the algorithms function and whether the periodic behavior by the players was equally predictable in all three cases.

Several types of testing and training configurations were tested:

The Test Scenarios

ID	Profile learning days	day(s) tested
Week	1 week	The 3 following weeks
Wednesday	5 Wednesdays	The 20 following Wednesdays
Saturday	5 Saturdays	The 20 following Saturdays
60days	60 days in a row	The following 3 weeks

TABLE I
TABLE OF TEST SCENARIOS.

There are two important properties of interest which have led to this selection of tests. One aspect is wether the algorithm requires ”same-day” data, such as only data from mondays in order to predict the next monday or workday-data in order to predict workdays. Further, we are interested in the algorithms quality based on the length of the learning phase and introduced tests of varying learning length.

In order to determine whether a prediction is good or not, a scoring system based on two factors was developed. The first number, simply called the score, is the number of data points which are above the rest of the observerd day. In other words, how much the prediction stays above the day. This is an important part, as one of the goals of the algorithm should be to make predictions which are slightly higher because we want to avoid predictions which are slightly lower, as this has a potentially detrimental effect on the quality of gaming as a service. On the other hand, a perfect score would be any line which is far above the day, allowing wild predictions to be made. The other factor is therefore the average distance between the prediction and the observed day. The average distance is calculated by summarizing the absolute value of the difference between the profile and the observed day at each data point and dividing by the number of data points. If this value is low, the prediction is close to the observed day. Combined with the score, we see that a good prediction is found where the score is the highest and average distance is as its lowest.

As the value b can be chosen freely, every b was tested and evaluated. An interesting question was wether some positions of b were better than others for making predictions. Since all b were tested, the four scenarios on the three games amounted to a total of 70848 individual predictions. Which where subsequently analyzed.

Result Summary Table: Scores

Game	**ID**	Score			
		Mean	Median	Min	Max
CSS	Week	245.643	273	2	288
CSS	Wednesdays	217.333	250	3	288
CSS	Saturday	252.980	271	29	288
CSS	3Weeks	239.321	258	3	288
FM	Week	216.389	243	0	288
FM	Wednesday	231.952	261	3	288
FM	Saturday	266.949	280	21	288
FM	60days	240.524	266	3	288
SC	Week	222.651	251	2	288
SC	Wednesday	232.420	257	3	288
SC	Saturday	244.991	260	4	288
SC	60days	245.790	274	0	288
Game	**ID**	Average Distance			
		Mean	Median	Min	Max
CSS	Week	11478.676	8542	0	94015
CSS	Wednesdays	8825.054	7141.5	0	84444
CSS	Saturday	16863.063	13767	1512	101236
CSS	60days	9900.050	9280	0	60197
FM	Week	2483.902	1648	0	23903
FM	Wednesday	2934.755	1827.5	0	87520
FM	Saturday	5763.684	4438	532	44015
FM	60days	2688.311	2059.5	0	33805
SC	Week	361.662	218	0	2855
SC	Wednesday	166.501	134	0	3383
SC	Saturday	295.791	243	3	1512
SC	60days	189.124	164	0	1406

TABLE II
THESIS SUMMARY

Table II summarizes the results. The scores show a consistently high score with some examples where the score was low. This was mostly due to anomalies in the data, which we will come back to. From the average distance results, we see that the average distance can be quite high at times, especially in Counter Strike Source, where it may be as high as several thousands of players above the actual day. Most of these cases come from when the b value was from a place where the number of players was increasing, typically in the evenings.

Figures 2,3 and 4 show examples of the profile being applied to a day during the testing periods. In Figure 1, we see a prediction which is close, but sometimes below the percieved day. A score of 262 out of 288 possible illustrates this. Also, at the end of the day, the prediction rises above the percieved day, contributing to the somewhat high average distance. In Figure 2, the same phenomenon is exaggerated. It also shows how a short learning period is susceptible to producing extreme predictions when b is in an area where there is normally high degree of variation in the data. Figure 3 showcases where the predictions work best. The score is a perfect 288 and the prediction remains close, but above the perceived day. More in-depth coverage of the results along with variations of these experiments can be found in [34]

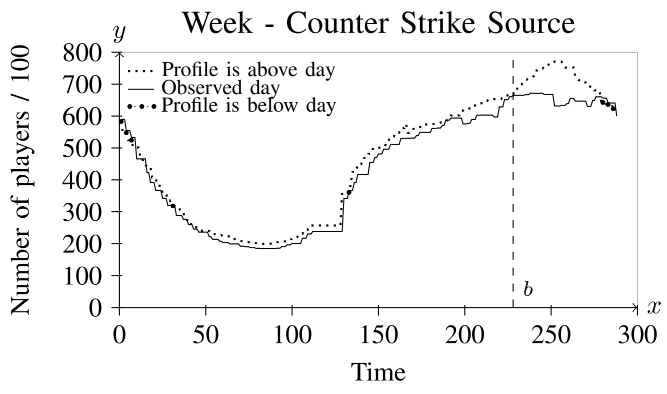

Fig. 2. Results from the game Counter Strike Source of scenario: Week, using *beta* value 228 on day 19. Score: 262 (90.97%) Avg dist: 3209

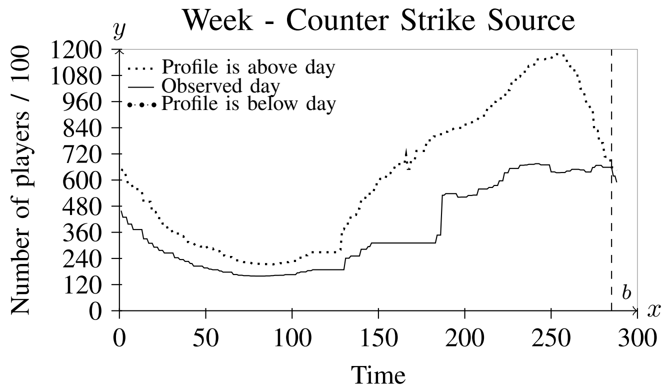

Fig. 3. Results from the game Counter Strike Source of scenario: Week, using *beta* value 285 on day 18. Score: 288 (100%) Avg dist: 23303

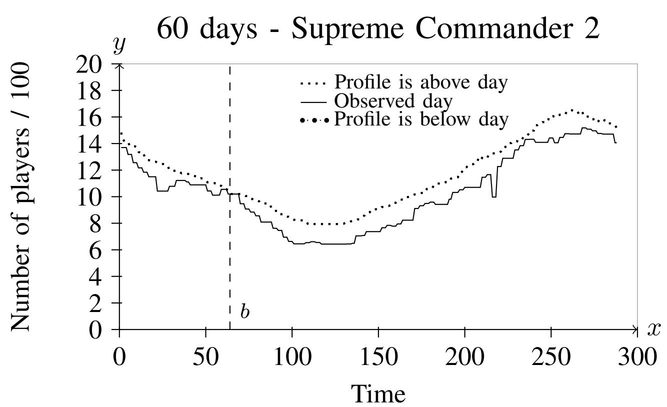

Fig. 4. Results from the game Supreme Commander 2 of scenario: 60 Days, using *beta* value 64 on day 7. Score: 288 (100%) Avg dist: 125

VI. Discussion and Conclusion

One of the interesting things about this methodology, is that one only needs a single point in order to make the prediction. However convenient, this is also a potential drawback. Our observation was that once the data gets noisy, or there is an anomaly, the prediction will end up to be maladjusted to the rest of the day. One way to address this is to use a sliding average which will smooth noisy data. Using the data from Steampowered.com, we did not need to reduce the noise, which was very low. However, we would observe some anomalies, specifically drops in the player count.

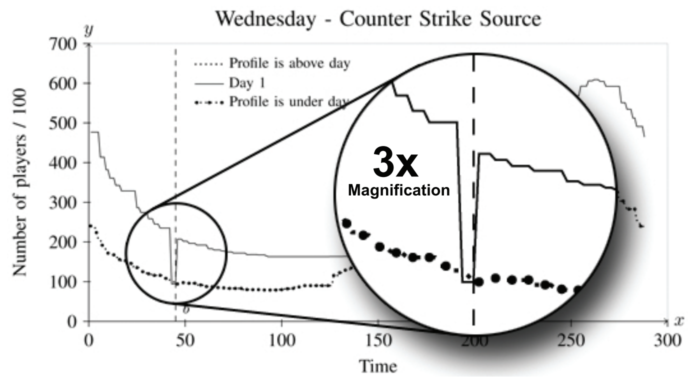

Fig. 5. 4x example

Another interesting question which arises from this work is, if one only needs one data measurement to make a prediction, what is the best - or subsequently most predicting - point in the day? Our observations indicate that areas with the least variation in the normalized days contribute to the best predictions. Those areas are when the number of players decrease, typically during the late night and early morning and during peaks, when the number of players is at its highest.

When the final profile is calculated, using a high percentile will lead to a more optimistic prediction. Ultimately this actually leads to a better result along the lines that a prediction which stays close to the actual day, but slightly above it is better than an observation which is close but sometimes actually below the observed day. Using a percentlie of 95% gave us a higher score albeit somewhat larger average distance as well. Our experience is that as one uses a higher percentlie, it becomes even more important to have sufficient training days so that the distribution at each datapoint has sufficient data.

There is no direct restriction on the neccesary number of learning days for the profile to become optimal. Our experience achieved a good result with 30 days but even with a week's worth, the algorithm was able to make good predictions. The only problem is that with fewer days, the impact of anomalies, such as small downtimes or peaks, is much greater, which may lead to faulty predictions if b coincides with the anomaly.

One interesting find, was that weekends and workdays could be used in the same learning set without degradation of the predictions. Due to the normalization process, differences are minimized and they can be used together. We percieve this as a benefit, as all days in the week can be used leading to a stronger profile. This is especially important for predicting weekends. For example, if one would want a profile from 30 weekend days, one would to wait 105 days (15 weeks) in order to gather sufficient data, which is impractical, to say the least. Using only the last 30 days, one has cut the waiting time down by over 70%. We also found that the profile remained accurate over the entire testing period, suggesting that it is not neccesary to re-calculate the profile often.

Even though the game servers for games such as Counter Strike Source stem from an individual company and datacenter. We believe that the type of behavior observed in our data is appliccable to other games as well. From inspecting date for over 30 games from Steam, the same periodic structure was identified. We argue that the results correspond with those from MMORPGs such as EVE Online [11].

A. Conclusions

This paper has introduced a new method for the prediction of the number of players on sepcific games. The algorithm has a resource management viewpoint and can be adjusted in order to provide more optimistic predictions. Our results show that the algorithm has the ability to make good predictions if allowed enough time for learning. Recommendations have been made as to how anomalies in the data can be circumvented so that they do not have a detrimental effect. In our future work, we will attempt to use this algorithm in a real-life scenario where the number of servers is scaled to match the prediciton.

References

[1] G. Armitage. An experimental estimation of latency sensitivity in multiplayer Quake 3. In *Networks, 2003. ICON2003. The 11th IEEE International Conference on*, pages 137–141. IEEE, 2003.

[2] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in unreal tournament 2003®. In *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, pages 144–151. ACM, 2004.

[3] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing sla violations. In *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*, pages 119–128. IEEE, 2007.

[4] M. Bray. Review of computer energy consumption and potential savings. *Dragon Systems Software Limited (DssW), December*, 2006.

[5] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle. Managing energy and server resources in hosting centers. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, pages 103–116. ACM, 2001.

[6] K.T. Chen, P. Huang, G.S. Wang, C.Y. Huang, and C.L. Lei. On the sensitivity of online game playing time to network QoS. In *IEEE Infocom, April*. Citeseer, 2006.

[7] Kuan-Ta Chen, Polly Huang, and Chin-Laung Lei. How sensitive are online gamers to network quality? *Commun. ACM*, 49:34–38, November 2006.

[8] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. In *ACM SIGMETRICS Performance Evaluation Review*, volume 33, pages 303–314. ACM, 2005.

[9] Ian Whalley James E, Jeffrey O. Kephart, and Hanson Malgorzata Steinder. Intelligent workload consolidation with power and performance tradeoffs.

[10] E. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. *Power-Aware Computer Systems*, pages 179–197, 2003.

[11] W. Feng, D. Brandt, and D. Saha. A long-term study of a popular mmorpg. In *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, pages 19–24. ACM, 2007.

[12] W.C. Feng. Making a case for efficient supercomputing. *Queue*, 1(7):54–64, 2003.

[13] Gartner. Gartner estimates ict industry accounts for 2 percent of global co2 emissions, 2007. .

[14] J.E. Hanson, I. Whalley, M. Steinder, and J.O. Kephart. Multi-aspect hardware management in enterprise server consolidation. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 543–550. IEEE, 2010.

[15] R.R. Harmon and N. Auseklis. Sustainable it services: Assessing the impact of green computing practices. In *Management of Engineering Technology, 2009. PICMET 2009. Portland International Conference on*, pages 1707 –1717, aug. 2009.

[16] T. Henderson and S. Bhatti. Networked games: a QoS-sensitive application for QoS-insensitive users? In *Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?*, pages 141–147. ACM, 2003.

[17] N. Kandasamy, S. Abdelwahed, and J.P. Hayes. Self-optimization in computer systems via on-line control: Application to power management. 2004.

[18] K. Kant and J. Alexander. Proactive vs. reactive idle power control. *Proc. of DTTC*, 2008.

[19] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[20] B. Khargharia, S. Hariri, and M.S. Yousif. Autonomic power and performance management for computing systems. *Cluster Computing*, 11(2):167–181, 2008.

[21] Steve Kleynhans. the green pc environment, 2007.

[22] P. Kurp. Green computing. *Communications of the ACM*, 51(10):11–13, 2008.

[23] D. Kushner. Engineering everquest: online gaming demands heavyweight data centers. *Spectrum, IEEE*, 42(7):34 – 39, 2005.

[24] Liang Liu, Hao Wang, Xue Liu, Xing Jin, Wen Bo He, Qing Bo Wang, and Ying Chen. Greencloud: a new architecture for green data center. In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ICAC-INDST '09, pages 29–38, New York, NY, USA, 2009. ACM.

[25] J.R. Lorch and A.J. Smith. Reducing processor power consumption by improving processor time management in a single-user operating system. In *Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 143–154. ACM, 1996.

[26] Simon Mingay. IT Vendors, Service Providers and Users Can Lighten IT's Environmental Footprint. 2007.

[27] G.E. Moore et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[28] J. Nichols and M. Claypool. The effects of latency on online madden NFL football. In *Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*, pages 146–151. ACM, 2004.

[29] M. Oliveira and T. Henderson. What online gamers really think of the Internet? In *Proceedings of the 2nd workshop on Network and system support for games*, pages 185–193. ACM, 2003.

[30] Ramya Raghavendra, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. No power struggles: coordinated multi-level power management for the data center. *SIGARCH Comput. Archit. News*, 36:48–59, March 2008.

[31] J.A. Roberson, G.K. Homan, A. Mahajan, B. Nordman, C.A. Webber, R.E. Brown, M. McWhinney, and J.G. Koomey. Energy use and power levels in new monitors and personal computers. *Lawrence Berkeley National Laboratory, Berkeley, CA*, 2002.

[32] V. Sharma, A. Thomas, T. Abdelzaher, K. Skadron, and Z. Lu. Power-aware qos management in web servers. In *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*, pages 63–72. IEEE, 2003.

[33] Steam. About steam. http://store.steampowered.com/about/.

[34] J. Tyvand. On the predictability of server resources in online games, an investigative approach. Master's thesis, University of Oslo, 2011.

[35] T. Yasui, Y. Ishibashi, and T. Ikedo. Influences of network latency and packet loss on consistency in networked racing games. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–8. ACM, 2005.

[36] S. Zander and G. Armitage. Empirically measuring the QoS sensitivity of interactive online game players. In *ATNAC04: Australian Telecommunications Networks and Applications Conference*, 2004.