Non-intrusive Estimation of QoS Degradation Impact on E-commerce User Satisfaction

Nicolas Poggi[1], David Carrera[1,2], Ricard Gavaldà[1] and Eduard Ayguadé[1,2]
[1] Technical University of Catalonia (UPC)
[2] Barcelona Supercomputing Center (BSC)
Barcelona, Spain

***Abstract*—With the massification of high speed Internet access, recent industry consumer reports show that Web site performance is increasingly becoming a key feature in determining user satisfaction, and finally, a decisive factor in whether a user will purchase on a Web site or even return to it. Traditional Web infrastructure capacity planning has focused on maintaining high throughput and availability on Web sites, optimizing the number of servers to serve peak hours to minimize costs. However, as we will show with our study, the conversion rate —the fraction of users that purchase on a site— is higher at peak hours, where systems are more exposed to suffer overload.**

In this article we propose a methodology to determine the thresholds of user satisfaction as the QoS delivered by an online business degrades, and to estimate its effects on actual sales. The novelty of the presented technique is that it does not involve any intrusive manipulation of production systems, but a learning process over historic sales data that is combined with system performance measurements. The methodology has been applied to Atrapalo.com, a top national Travel and Booking site. For our experiments, we were given access to a 3 year long sales history dataset, as well as actual HTTP and resource consumption logs for several weeks. Obtained results enable autonomic resource managers to set best performance goals and optimize the number of server according to the workload, without surpassing the thresholds of user satisfaction and maximizing revenue for the site.

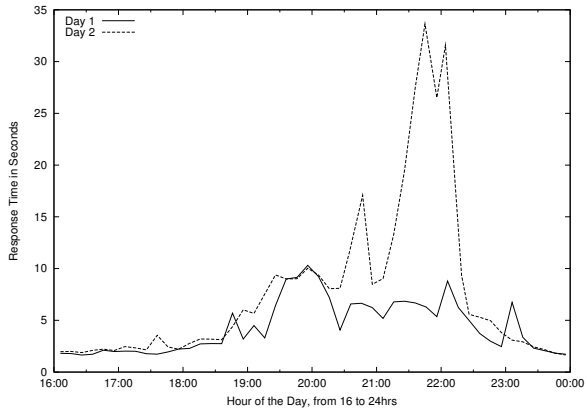
I. Introduction

Over the last years, high-speed Internet access has become commodity both at home and work in many countries, with numbers reaching 91% in the US [1], and similar numbers in Europe [2] at the workplace. High speed Internet access is changing the way users interact with websites, their expectations in terms of performance (response time), and patience levels [3]. A recent consumer report by Forrester Research [1], shows that users expect Web sites to load faster than in previous years; that about 23% of dissatisfied online shoppers attribute their dissatisfaction to slow Web sites, while 17% to crashes or errors. Moreover, at conference series as the *O'Reilly Velocity* conference, Web industry leaders such as Google, Bing, AOL, and Amazon have been releasing results on how performance affects their business: Google reports that by adding half a second to their search results, traffic drops by 20% [4]; AOL reports that the average page views can drop from 8 to 3 in the slower response time decile [5]; Amazon reports that by adding 100ms, sales drop by 1% [6]. Furthermore, Google has announced [7] that it takes into account response time in their page ranking algorithm affecting positioning on search results, a mayor income source for online retailers. Web site performance has become a key feature in determining user satisfaction, and finally a decisive factor in whether a user will purchase on a Web site or even return to it.

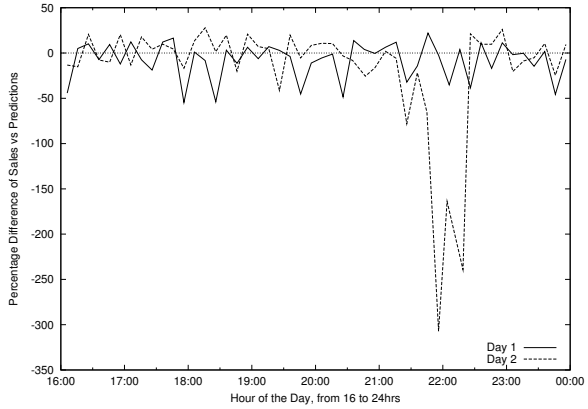
Therefore, capacity planning techniques for online Web sites are living a period of changes: the cloud computing paradigm and the appearance of advanced service management products that dynamically adapt provisioned resources pose new challenges at deployment time. System administrators need to make important decisions about the different parameters that seriously affect the business results of any online Web site such as: what is the best performance goal for each online application, usually presented in the form of a response time objective? What is the highest number of servers that the company may afford on peak hours and other workload surges? What would be the effect, in business terms, of limiting the resources for an application and degrading its performance slightly to reduce the bill of the hosting? In this paper we propose a methodology that provides answers to these questions without need to manipulate the systems in production, as it only requires offline information usually collected by most online Web sites.

Determining the impact of high response time on sales and business volume is something that can be studied injecting delay on pages and using A/B testing methodology to measure, but this approach is not always feasible. Very large companies can intentionally degrade the performance delivered by a fraction of their servers with minimal impact for their business to study the effects of performance degradation on their business balance. The same process may have a strong impact for small companies and thus, they hardly decide to use such approach. Therefore, new methods must be developed to carry on such studies with minimal interference on the systems.

In this paper we introduce a novel methodology for studying what is the total volume of sales lost for an online retailer during performance degradation periods. We use Machine Learning techniques to predict the expected sales volume over time and look for deviations over the expected values during overload periods that introduce performance degradation. Using such technique, we can estimate the total impact of high *response time* in the business activities of an online service in a non-intrusive way. The proposed methodology starts with a study and characterization of the sales log, that leveraging Machine Learning techniques constructs a model of sales. Such model allows for an accurate prediction of expected sales in short time frames. The model is then used to contrast actual sales with expected sales over time, and determine the impact in sales of overload periods that caused degraded QoS



(a) Observed response time



(b) Relative error between actual sales and predicted sales

Fig. 1. Comparison of Response Time and Predictions in Normal and Overload Operation for Two Days

—measured in the response time— by the online applications.

For the sake of clarity, we include here a simple example that shows the use of the methodology presented in this paper to a real Web application. Figure 1(a) shows response time for the same application over a 8h period for two different days, where *Day2* corresponds to a day in which some overload was observed, and as a consequence, performance degradation resulted in a response time surge; on the other hand, *Day1* corresponds to a regular day. A question one may want to answer after observing such performance degradation is: what is the volume of sales that was lost due to response time surge? The result of applying the proposed technique to the previous example can be seen in Figure 1(b), where relative error between real sales and the predicted sales volume is shown. As it can be seen, the expected sales would be higher than what was observed, and it can be even quantified by what margin. In the following sections we will further elaborate on how to systematically build the sales model that helps estimating the loss of sales during the overload period. Where the model needs to capture conversion rate variability of every time and date in small time frames, as QoS on servers changes by the minute and sales are prone to seasonality effects.

Online Travel Agencies (OTAs) are a prominent sector in the online services market: according to the 2008 Nielsen report on Global Online Shopping [8], Airline ticket reservation represented 24% of last 3 month online shopping purchases, Hotel reservation 16%, and Event tickets 15%; combined representing 55% percent of global online sales in number of sales. In our study, we take the case of a Atrapalo.com, a top national Online Travel Agency and Booking Site representative of the OTA industry, that features popular E-Commerce applications found in the Web. We have been given access to a 3-year sales log of the OTA, including time and volume of each operation, as well as several weeks' access and resource usage logs of the company's execution environment. Each dataset, featuring several million HTTP requests, from February to March 2010, some of them comprising overload periods.

The main contributions of this paper are two: firstly, the study of conversion rates for a real OTA, that brings results not previously reported in the literature about peak load periods; and secondly, the use of a sales model built using machine learning technologies with the goal of quantifying sales loses due to overload periods and response time surges. To our knowledge, such application of a sales model has not been previously reported in the literature. We have classified the contributions as a three steps methodology that can be systematically followed to understand the sales characteristics of any online business, and to build and use the sales model mentioned above. In particular, the proposed steps are:

- Step 1: Study the conversion rate variation during the day for the OTA based on sales datasets (Section III).
- Step 2: The construction of a model for sales prediction through Machine Learning techniques and its validation (Section IV).
- Step 3: Characterize response time thresholds of user satisfaction (satisfied, tolerating, frustrated) for the OTAs applications (Section V).

The output of Step 3 is suitable to be used for building autonomic resource managers that dynamically adjust the resources allocated to each different online application in cloud-like environments while observing conversion rates, performance, and energy constraints. Such use of the proposed methodology is our current subject of research. The results presented in this paper are part of a wider study previously presented in [9], where machine learning techniques are leveraged to predict anonymous Web visitors value for the site, and prioritize their sessions on the server to shape its QoS accordingly.

II. The Online Travel Agency

Online Travel Agencies such as Atrapalo.com, present a wide range of *products*: flights, hotels, car, restaurants, activities, vacational packages (or 'trips'), and event booking. For this purpose they rely on a wide range of technologies to support them: dynamic scripting, Javascript, AJAX, XML, SSL, B2B Web services, Caching, Search Algorithms and Affiliation; resulting in a very rich and heterogeneous workload. While these technologies enhance users' experience and privacy, they also increase the demand for CPU and resources on the server side. Furthermore, as Web applications become more resource-intensive and the large number of potential visitors on the Web, system overload incidence is growing along [10]. Moreover, visits to travel sites present a great variability depending on time of the day, season,

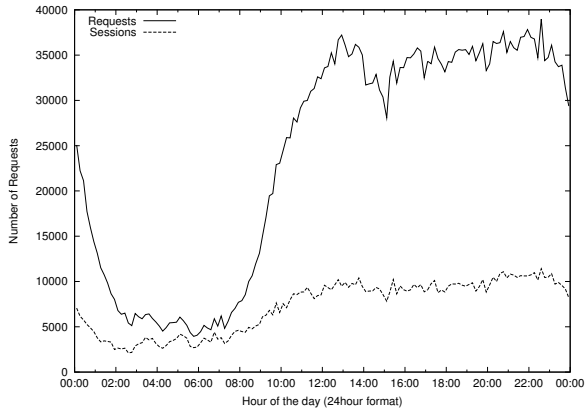


Fig. 2. Number of requests and sessions for 24hr period

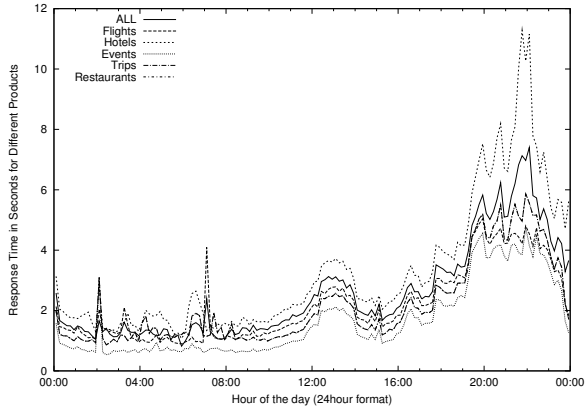


Fig. 3. Average response time by product in 24hr period

promotions, events, and linking; creating bursty traffic, making capacity planning and providing a high QoS a challenge.

A. Products

Some of the above mentioned products inventories are maintained internally (e.g. 'restaurant booking'), some are completely external (e.g. 'flights'), and some other products (e.g. 'hotels') are mix of internal and external providers. Access to the external providers is performed via B2B Web services, which causes QoS to be dependent on external sites for some operations. The company itself is also a B2B provider for some customers and meta-crawlers, acting as their provider via a Web Service API. Although the company's main presence and clientele is in Europe, a small percentage of the visits are from South America, and few more from the rest of the world. It is important to remark that the site is a multi-application Web site. Each *product* has its own independent application code base and differentiated resource requirements, while sharing a common programming framework.

B. Long-term Sales logs and Short-term Performance logs

Sales logs are used for modeling sales through machine learning techniques, as described later in Section IV. For this study, we were given access to sale history datasets for the OTA's different *products*. For each *product* we have the exact date and time for each purchase that was made. However, we did not have given access to sales amounts or margins, just the moment of a user sale. Sales datasets range from 01/01/2007 to 04/01/2010, comprising a period of over 3 years. There is a great variation of sales between products and times of the year due to seasonality effects. Sale volumes are not reported due to confidentiality limitations.

Performance logs are used for quantification of response time effects; they were produced through existing probes in the PHP dynamic application and provided by Atrapalo.com. Which consists of transactions collected over different days of February to April 2010. With a total of 22 full days containing 57,375,787 total dynamic requests, representing 16,393,746 distinct sessions, and 15,488 different pages (non-ambiguous URLs). They contain all kind of system and application metrics from regular HTTP access data, to per-node application resource usage, database, B2B requests and server status. All the information is correlated in time with the sales log.

We also had access to the Apache logs and monitoring system access for the days covered in the dataset and other random weeks of 2009 and 2010. These auxiliary logs have been used to validate and explain obtained results from the workload. In a previous study [11], we performed a complete workload and resource characterization of part of these datasets. Some results of such study are shown in Figure 2, were general traffic volume for a sample 24hr period is shown; and Figure 3 shows the great variation of response time observed over the day for an averaged 24hr period of the different products.

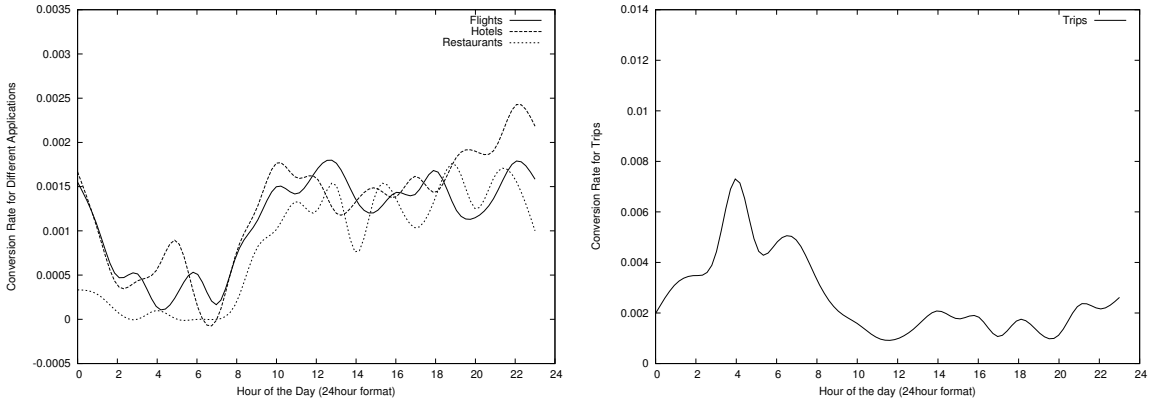
III. Conversion Rates

In Internet marketing, the conversion rate (CR) can be generally defined as the ratio between the number of *'business goal' achievements* and the number of *visits* to a site. In the scope of this paper, a *goal* is achieved when a customer purchases a product offered by the OTA during a browsing session. The CR is one of the most widely used indicators of business efficiency on the Web. A high CR indicates that a high number of visitors purchase on the site on each visit, while a low CR indicates that most visits use server resources without returning value in terms of direct revenue. Many factors can affect the conversion rate, e.g. type of product, content, brand, SEO ranking, affiliation, availability, and QoS measured in response time. Values for CRs are different for each Web site, and are part of their marketing strategy and business nature; however values should remain within the same range over time for similar products on different sites.

In this section we analyze the CR for the different products of the OTA. The objective of such analysis is to understand how selling hotspots are distributed over time for the studied business. Such study is very relevant to measure and quantify the effects of QoS on the sales volume observed for the OTA. Notice that the CR does not necessarily change with oscillations in volume of traffic observed for any Web site (e.g. Figure 2), as CR has to do with the fraction of visitors purchasing products, and not with the overall volume of visitors.

A. Conversion Rates Study

In average for all applications of the site, the conversion rate is higher at the site's peak traffic time in number of requests. The



(a) Conversion Rate by hour of the day for similar apps

(b) Conversion Rate by hour of the day for Trips

Fig. 4. Conversion Rate by hour for different applications

CR of the averaged application follows very closely the daily patterns of the number of sessions in the system (see Figure 2) during a 24 hour period: from 1am to 8am there is a low CR, sleep time; from 8am until 14hrs, lunch time in Spain where most of the traffic comes from, the CR increases progressively during the morning as the number of sessions increases; from 14hrs until 16hrs, the CR lowers during lunch time, then continues to grow in the afternoon along with sessions, until 18hrs where the CR decreases due to end of office hours; it increases again from 20hrs peaking around 22hrs. Figure 4(a), pays special attention to the CR for some products with similar daily patterns: flights, hotels, and restaurants. As it can be observed, they follow similar daily patterns; special mention deserves the case of the *events* application, which is in the magnitude of 100 times compared to applications in Figure 4(a), for this reason not plotted on the figure. This might be due to the fact that certain events, i.e. a concert, can only be purchased in exclusive in this site, making this product more inflexible to loss in sales due to poor performance. Finally, it is remarkable the case of *trips* (vacational packages) plotted in Figure 4(b), which is the only product that does not follow a daily pattern as seen in the other charts and has a higher CR during late night time. Experts from the OTA commented that this might be due to the fact that vacational packages involve more people, normally families, and that decisions are made during the evening and night. More investigation on this is out of the scope of this study, but the *trips* product is included for validation of the results as the application should be more sensible to high response time, as CR is low at the sites peak time.

A remarkable conclusion must be pointed from the observation of these figures: except for the case of the *trips* application, the CR for all the other products available from the OTA follow a pattern that is strongly correlated to the traffic pattern observed in Figure 2. The interpretation of this fact is that the hours at which the traffic volume is the highest, the fraction of customers that are actually purchasing products is also higher, resulting in still higher sales periods. Recall that such a result indicates that the relation between volume and sales is not constant over time, and leads to a very important increase of sales over peak periods were not only traffic volume grows but also the average CR.

Notice that it is a usual case that many infrastructures are not dimensioned for sustaining QoS at peak hours, as they are considered surges in the traffic and static provisioning of resources to manage punctual very high traffic volumes is unaffordable. Of course, such decision results in worse response time and QoS in general during peak periods. Looking at the charts for Atrapalo.com, it can be observed that these are not only surges in traffic, but also the best selling periods of the day. So, although industry and consumer studies (see Section I) reports that a high response time has a direct effect on sales, as conversion rates are higher at peak times, the total loss in sales might not be apparent in most cases.

B. Conversion Rates as a Function of Response Time

It has been show above that, in general terms and for most products, high conversion rate times coincide with the rest of the product's peak hours and worst response times. This can be seen by comparing Figures 2, 3, 4(a), and 5.

Figure 5 explores the *conversion rate* as a function of average response time by combining data fom the long-term sales dataset and data from the short-term performance logs (see Section II) grouped in 10 minutes intervals from the performance dataset. By analyzing the Figure there is no clear indication that a high response time yields less sales. On the contrary, most applications are able to maintain CR and sales even in the periods of higher response times. To study this effect further, the next section presents our methodology for forecasting expected sales in short time frames for each application; in order to measure how a low QoS during peak times affects more sales than previously reported.

IV. Predicting Sales with Machine Learning

Most traditional sales forecasting techniques involves some form of linear or multiple *regression analysis*, in our preliminary work we have tried several methods i.e. *linear*, *quadratic*, and *cubic* regressions to predict expected sales for different weeks in short periods of time bins, e.g. 10 minutes, using the *sales* dataset. We found that predictions were too general and not precise for the short time frames we wanted to measure response time in. To overcome this situation and improve predictions, we

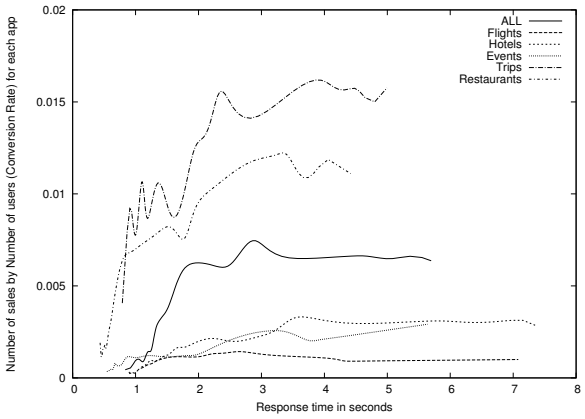


Fig. 5. Conversion Rate as a function of response time

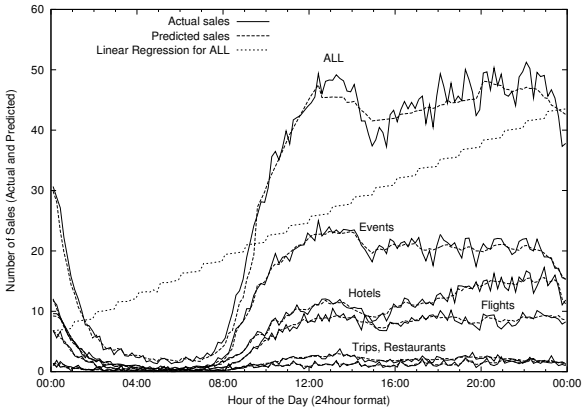


Fig. 6. Classifier Precision by App for an Averaged 24hour Day

have built a sales forecasting prototype implementing Machine Learning numerical algorithms, detailed in the next section.

To forecast sales, we have built a prototype based on the WEKA [12] open-source Machine Learning package, which contains several ready to use classifier algorithms. As we wanted to predict expected sales for short time bins, representative of the *response time* (QoS) in the system at that moment; while also having precise prediction results, which is dependent on the volume of sales per time bin in the *sales* dataset, we have tested predictions for: 30, 15, 10 and 1 minute bin intervals. 1 minute bins turned out to be too low to have accurate prediction results, and as Web sessions for buying visits are longer [13], it only partially reflected the response time for the session. 30 minutes was too high, as server status might have changed drastically in that time frame. After some experimentation, we have decided to use 10 minutes as the time frame for the rest of the experiments; it is low enough to represent the current QoS on the system and high enough to cover the most of the Web session with accurate sales predictions.

In WEKA, a *predictor* model is trained using a specially formatted *training dataset*, that should contain the most relevant available variables to predict sales. After training the *predictor* with the training dataset, a *test dataset* with the same format is used to perform predictions. The predictor reads the test dataset, ignoring the *class* —the unknown variable— in our case the number of sales, and according to the training and the algorithm used, outputs a prediction of sales —the class— for each of the time bins.

A. Prediction Methodology

For the training dataset, we have preprocessed the 3 year+ long sales history dataset (see II-B) into 10 minute bins, each day containing 144 bins; creating a *training* and *test* datasets for each application of the OTA, and one for *ALL* the applications combined. The resulting dataset contains the following attributes: Number of sales for the 10 minute bin; Year of the sale; Month of the sale; Day of the year; Day of the week; Hour of the day; and the 10 minute bin for the day.

The goal is that each attribute adds valuable information when building the *predictor* model. For example, the month of the year should add information on whether we are in low or high season. Each day of the week has differentiated sales volume: Mondays have more sales and decreases through the weekend for most applications, except for *events* which is higher on Fridays. The same goes for the time of the day, that has different conversion rates for each application (Figure 4(a)). More attributes could be added to improve predictions, especially to cover seasonality effects e.g. of holidays, where the numbers of days to a holyday could have been added. However the purpose of the prototype is to provide representative predictions and a proof-of-concept of the method.

It is important to remark that the *training* dataset only contains data previous to the *test* dataset. We use the 3 year long sale history to create the training dataset, we cut the dataset the day before we want to apply the predictions —the 31^{st} of January 2010— so no information of the future known when building the model. The prototype outputs the predictions for the next two months, until the 1^{st} of April. As resulting predictions should be as-close-as-possible to the actual number, but not necessarily predicting the exact number, for this purpose we have implemented in the prototype several *numerical classifiers* available in WEKA, that predict values within an error percentage range. The next sub-section presents results for the classifier algorithms.

B. Prediction Results

The training dataset used by the prototype to build the *predictor* contains 162,438 instances, while the test dataset contains 8,151 instances. We present results for the following numerical classifiers found in WEKA: LinearRegression, REPTree, Bagging(M5P), and Bagging(REPTree). More complex classifiers could have been used e.g. *neural networks*, but the processing time required for training seems too high to consider them in a real-time applications and were discarded for the time being. Table I presents accuracy for the different selected classifiers: *LinearRegresion* is the least performing classifier with a Relative Absolute Error of 66.7%, while *M5P* and *REPTree* have 24.0% and 21.6% Relative Absolute Errors respectively. The *Bagging* meta-classifier was also tested implementing both *M5P* and *REPTree*, improving precision for both algorithms to 19.4% in the case of *Bagging(REPTree)*. Meta-classifiers split the training dataset in instances, testing different parameters of the selected classifier

	LinearRegression	M5P	REPTree	Bagging(M5P)	Bagging(REPTree)
Correlation coefficient	0.7136	0.9499	0.9589	0.9527	0.9674
Mean absolute error	13.0187	4.61	4.1584	4.4595	3.7328
Root mean squared error	15.575	6.4784	5.8763	6.297	5.2484
Relative absolute error	67.7335%	23.985%	21.635%	23.2019%	19.4211%
Root relative squared error	72.4566%	30.1381%	27.3373%	29.2945%	24.4163%

TABLE I
CLASSIFIER EVALUATION

and selecting the most precise ones, they perform several iterations of the regular classifiers with different attributes and selecting the best for each case, but take longer to train. As a note, all experiments were performed using default values in WEKA. Linear regression has essentially no parameters. RepTree and M5P, like all decision tree methods, have a parameter controlling the size of the tree; we verified that, in our case, optimizing over that parameter gave only negligible advantage over the default value.

Figure 6 presents a graphical comparison of actual sales by product of the best performing classifier —Bagging(REPTree)— and the linear regression for *ALL*. As it can be seen, *LinearRegression* does not follow closely sales patterns, while *Bagging(REPTree)* and the rest of the classifiers detects shifts by hour hour of the day, especially day and night patterns with great accuracy for the different products.

Apart from linear regression, the rest of the classifiers have less than $\pm$ 2% error difference compared to actual sales when average response time for the 10 minute time bin falls within 2 seconds. Between 2 and 4 seconds, the classifiers are less precise and underperform up to -4%, this might indicate that the site sells more than expected at this QoS. After 4 seconds, classifier error rate starts increasing positively, over predicting results, with milestones at 6, 8, and 10 seconds. From 10 seconds, classifier error starts to grows steeply from an average of +3% errors to 40% at 14 seconds. When response time is above 10 seconds, classifiers over predict indicating that sales should have been higher, and deviate from actual sales results.

In this section we have presented our methodology for predicting future sales for short, 10 minutes, time bins by implementing and testing different Machine Learning classifiers on the sales dataset. Tree classifiers —M5P and REPTree— as well *Bagging* meta-classifier implementing both algorithms, offer high accuracy predicting sales for normal Web site operation. In the next section we perform an evaluation of a high response time effect on predictions and user satisfaction.

V. RESPONSE TIME EFFECT ON USERS

In this Section we will analyze how perceived user satisfaction, measured in observed sales, is affected by abnormally high response times. Such effect is measured by comparing actual sales with predicted sales (modeled according to the description provided in Section IV).

Notice that, as it was shown in Section III-B, peak load times coincide with high conversion rate periods for most applications of the OTA, where we have shown that 4 out of the 5 applications analyzed have a corresponding high CR

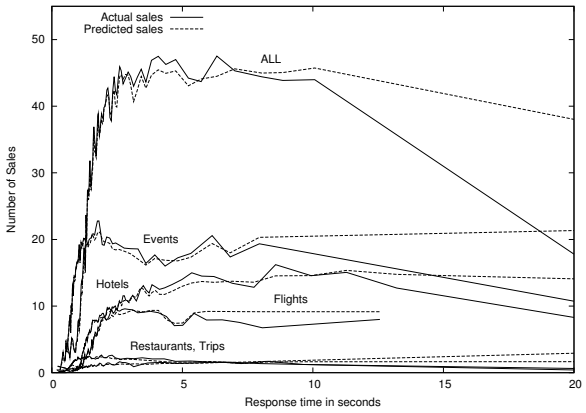


Fig. 7. Deviation from the prediction as a function of actual response time

when there are more users on the system and the QoS is worst. Therefore, a loss in sales due to high response time may not be apparent as the fraction of buyers in the workload is also higher at these times. Figure 5, exemplifies this situation where a high response time seems to maintain or even improve sales for most applications and the loss in sales might be undetected by system administrators, and most importantly, by management on high level reports. By comparing the actual sales with predicted sales (based on mostly non-overloaded system state) will highlight the net loss of sales due to surges in response time.

A. Determining response time thresholds and their effects

Figure 7 shows the observed and predicted sales volume for each application (Y-axis) as a function of the average response time (X-axis), using requests grouped in 10-min bins. The chart can be read as follows: for instance, for application *events*, request bins that showed an average response time of 5s, contained in average about 20 requests that were actual sales.

To produce the predicted data for this experiment, we have selected the *M5P* classifier, as it has less performance requirements than *bagging* meta-classifiers and the output tree model for predictions is more complete than REPTree. The output model is used for validation and to understand what features in the training dataset are the most relevant for predicting sales. Recall that each application has differentiated response time averages and also different conversion rates for each time bin.

For *ALL* applications, actual sales start to deviate from expected values at about 7 seconds, and from 10 to 20 seconds have a huge drop compared to the expected sales. Next line on the graph in number of sales is the *events*, which as stated

previously, due to exclusivity of event tickets the application is less flexible to changes, however at around 8 seconds sales have an important inflection point compared to predictions, but less steeply. Next is *Hotels*, which has the worst performing response times, as the application is heavily dependent on the database and external request QoS, however the huge drop in sales starts at 13 seconds and sales and decrease by 42% at 20 seconds. Next is the *Flights* application, in which sales start to drop at 6 seconds, however they don't drop completely, just under performs until the maximum response time of 13 seconds recorded by this application. *Trips* and *restaurants* applications have lower sales volumes and the inflection points cannot be seen correctly in the graph: *Trips* seems to be less tolerant to high response time, with drops starting at 3 seconds slowly until 10 seconds then a prominent drop; *Restaurants* in the other hand, the drop starts at 5 seconds and a pronunciated drop at 12 seconds.

Table II summarizes inflection times for each application, where we have separated inflection points in two thresholds in accordance to the APDEX standard [14]: *tolerating* and *frustration*. At the *tolerating* threshold, some sales are lost, but most users remain on the site; while at the *frustration* threshold, a high percentage of sales is lost and more users abandon the site. Also, an average of sales loss is presented for each range. The rest of the users with response time before the thresholds are considered to be *satisfied*. For the analyzed OTA, there is a *tolerating* response time threshold between 7 to 10 seconds, where some sales are lost. In average the *frustration* threshold for the analyzed OTA is at 10 seconds, where each increase in 1 second interval increases total sale loss by 6%. Even within the same Web site, each application has differentiated thresholds, and as products exhibit different flexibility, they also show different *tolerating* and *frustration* times and each should be treated separately from one another.

The *trips* application exhibits unique characteristics due to a distinctive conversion rate pattern, which is actually lower at high peak time, as it was shown in Figure 4(b) discussed in Section III. This makes the *trips* tolerating thresholds also lower than the rest of the applications, at 3 seconds of response time sales start to get lost. This effect is also seen in Figure 5, while the rest of the applications are not affected. The next sub-section continues with the discussion of results.

B. Discussion of Results

Results from this section show that the user's tolerating *response time* thresholds are higher for most applications of the OTA from previous literature, especially insdustry reports. Where our response time numbers are in line with Miller's [15] work on "Threshold Levels of Human Interaction and Attention With Computers" (see Section VI). We believe that most of industry reports take static pages as a base measurement, which might not be representing the reality of many E-Commerce sites. Some E-Commerce sites such as the OTA presented here, have pages that usually take a long time to be generated, e.g. *flight availability search*, that have a especial waiting page from which users assume a high complexity of the search, and thus are more patient on the results. The same can be observed for pages that involve external transactions, such as booking a restaurant, where many checks need to be performed i.e. credit card fraud, availability, re-check rates, Web Service calls, before the booking is made.

Appl	1^{st}thresh.	1^{st}drop	2^{nd}thresh.	2^{nd}drop
ALL	7-10s	5%	10-20s	53%
Events	7-8	10%	8-20s	48%
Hotels	10-13	13%	13-20s	42%
Flights	-	-	6-13s	22%
Trips	3-10s	20%	10-20s	50%
Restaurants	6-12s	25%	12-20s	50%

TABLE II
PERCENTAGE OF SALE LOSS BY INCREASING RESPONSE TIME: TWO INCREMENTAL RESPONSE TIME THRESHOLDS, AND CORRESPONDING DROP OF SALES IN PERCENTAGE

Furthermore, *tolerating* and *frustration* times are different for each application. For example the *events* application has exclusive content that cannot be purchased in online competitors, making it more inflexible than other applications such as *flights*, that has multiple competitors. Having different conversion rates and thresholds per application poses new challenges for differentiated and dynamic per-application QoS management during the day. Considering the current trend in Web eco-system to observe lower conversion rates due to different factors (e.g. rising competition, affiliation, meta-crawling, and changes in user habits such as multi-tabs[16]), on-line retailers will progressively support more traffic for less direct revenue by visit, increasing the importance of optimizing the number of servers without sacrificing sales.

The presented methodology enables online retailers to determine inflection points where sales start to be affected by the current application's *response time*. Where resulting values can be applied on autonomic resource managers to optimize the number of servers and reduce infrastructure costs in cloud computing environments. Most importantly, optimizations should not be made to accommodate all load, but to provide the best QoS when conversion rates are higher, generally at peak loads. Our model could have benefited from more overload periods in the dataset to improve precision, however, even at the low number of samples of high response time for the less popular products, main inflection points and loss of sale tendencies can be obtained from it. As an additional contribution, results from this study had led the presented OTA to make important changes in their infrastructure to avoid high response times, especially at peak times with positive results.

VI. Related Work

Response time effect on user behavior has been studied as long as 1968, where Miller [15] describes the "Threshold Levels of Human Interaction and Attention with Computers". In 1989, Nielsen [17] revalidated Miller's guidelines and stated that the thresholds are not likely to change with future technology. These thresholds being: 0.1 to 0.2 seconds, instantaneous; 1-5 seconds the user notices the delay but system is working; and 10 seconds as the threshold for user attention. Results from our experiments re-validate these results in the context of E-Commerce Web site for the average, *ALL*, application. However, threshold limits are different for different applications. Other authors [18], [14]

adhere to what they call the 8 seconds rule, where no page should take longer than 8 seconds to reply. The are several industry reports stating that users expect faster response times than previous years, specially the younger generation [3], [1]

To prevent loss in sales due to overloads several techniques have been presented such as *session-based admission* control systems [19], [20] used to keep a high throughput in terms of properly finished sessions and QoS for limited number of sessions. However, by denying access to excess users, the Web site loses potential revenue from customers. Later works include service differentiation to prioritize classes of customers, in [21], [22] authors propose the use of utility functions to optimize SLAs for gold, silver and bronze clients. In [9] authors propose the use of machine learning to identify most valuable customers and prioritize their sessions. Nowadays, Cloud computing enables enable sites to be configured at a minimum number of server resources and provision according to the incoming load, with a *de facto* unlimited number of servers, enabling autonomic resource managers to auto-configure server numbers [23], [6]. In [6], Mazzucco proposes the use of utility functions to optimize auto-provisioning of web servers. None of this works however, explore the effects of higher conversion rates a peak loads, and by ignoring this fact, QoS of service is no optimized and potential sales are lost.

VII. Conclusions

We have argued that the effect of response time degradation can be hidden by the fact that peak load times can coincide with high conversion rates, i.e. when higher fraction of visitors have intention to purchase. To overcome this effect we have introduced a novel methodology for studying what is the volume of sales lost in an online retailer due to performance degradation without modifying its application. We use machine learning techniques to predict the expected sales volume over time and look for deviations over the expected values during overload periods that may introduce performance degradation. Using such technique, we can quantify the impact of response time in the business activities of an online service.

We have tested the approach on logs from a top Online Travel Agency, using 3+ year long sales dataset, HTTP access log, and resource consumption logs for several weeks of 2010. From the obtained results we are able to identify inflection points where sales start to drop for different applications when response time is high. For the OTA, there is a *tolerating* response time threshold from 7 to 10 seconds, where some sales are lost, and a *frustration* threshold at 10 seconds, where each increase in 1 second interval increases total sale loss by 6%.

We are currently generalizing the model to be integrated in an autonomic resource manager for cloud-like environments, which takes into account conversion rates, while observing performance and energy constraints. The goal for such resource manager is to feature dynamic server provisioning, optimizing the number of servers according to the incoming workload, without surpassing the thresholds of user satisfaction and maximizing revenue for each of the hosted applications of a site.

Acknowledgements

We would like to thank Atrapalo.com, who provided the experiment datasets and domain knowledge for this study. This work is partially supported by the Ministry of Science and Technology of Spain and the EU contracts TIN2007-60625, TIN2008-06582-C03-01, and by the Generalitat de Catalunya (2009-SGR-980, 2009-SGR-1428).

References

[1] "E-commerce web site performance today. an updated look at consumer reaction to a poor online shopping experience," August 17, 2009.

[2] "Oecd report on broadband penetration. available at: http://www.oecd.org/sti/ict/broadband," Dec, 2009.

[3] C. Rheem, "Consumer response to travel site performance," April 2010.

[4] M. Mayer, "In search of a better, faster, stronger web," June 2009.

[5] D. Artz, "The secret weapons of the aol optimization team," June 2009.

[6] M. Mazzucco, "Towards autonomic service provisioning systems," *Cluster Computing and the Grid, IEEE International Symposium on*, vol. 0, pp. 273–282, 2010.

[7] "Using site speed in web search ranking. webpage: http://googlewebmastercentral.blogspot.com/2010/04/using-site-speed-in-web-search-ranking.html," April 2010.

[8] "Trends in online shopping, a Nielsen Consumer report," tech. rep., Nielsen, Feb. 2008.

[9] N. Poggi, T. Moreno, J. L. Berral, R. Gavald, and J. Torres, "Self-adaptive utility-based web session management," *Computer Networks Journal*, vol. 53, no. 10, pp. 1712–1721, 2009.

[10] S. Power, "Metrics 101: What to measure on your website," June 2010.

[11] N. Poggi, D. Carrera, R. Gavald, J. Torres, and E. Ayguadé, "Characterization of workload and resource consumption for an online travel and booking site," *IISWC - 2010 IEEE International Symposium on Workload Characterization*, December 2 -4, 2010.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[13] N. Poggi, T. Moreno, J. L. Berral, R. Gavald, and J. Torres, "Web customer modeling for automated session prioritization on high traffic sites," *Proceedings of the 11th International Conference on User Modeling*, pp. 450–454, June 25-29, 2007.

[14] P. J. Sevcik, "Understanding how users view application performance," *Business Communications Review*, vol. 32, no. 7, pp. 8–9, 2002.

[15] R. B. Miller, "Response time in man-computer conversational transactions," in *AFIPS '68 (Fall, part I): Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, (New York, NY, USA), pp. 267–277, ACM, 1968.

[16] N. Poggi, T. Moreno, J. L. Berral, R. Gavald, and J. Torres, "Automatic detection and banning of content stealing bots for e-commerce," *NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security*, December 8, 2007.

[17] J. Nielsen, "Usability engineering at a discount," in *Proceedings of the third international conference on human-computer interaction on Designing and using human-computer interfaces and knowledge based systems (2nd ed.)*, (New York, NY, USA), pp. 394–401, Elsevier, 1989.

[18] D. F. Galletta, R. Henry, S. Mccoy, and P. Polak, "Web site delays: How tolerant are users," *Journal of the Association for Information Systems*, vol. 5, pp. 1–28, 2004.

[19] L. Cherkasova and P. Phaal, "Session-based admission control: A mechanism for peak load management of commercial web sites," *IEEE Transactions on Computers*, vol. 51, pp. 669–685, 2002.

[20] J. Guitart, D. Carrera, V. Beltran, J. Torres, and E. Ayguad, "Session-based adaptive overload control for secure dynamic web applications," *Parallel Processing, International Conference on*, vol. 0, pp. 341–349, 2005.

[21] D. F. Garcia, J. Garcia, J. Entrialgo, M. Garcia, P. Valledor, R. Garcia, and A. M. Campos, "A qos control mechanism to provide service differentiation and overload protection to internet scalable servers," *IEEE Transactions on Services Computing*, vol. 2, pp. 3–16, 2009.

[22] J. M. Ewing and D. A. Menascé, "Business-oriented autonomic load balancing for multitiered web sites," in *MASCOTS*, pp. 1–10, 2009.

[23] M. Al-Ghamdi, A. Chester, and S. Jarvis, "Predictive and dynamic resource allocation for enterprise applications," *Computer and Information Technology, International Conference on*, vol. 0, pp. 2776–2783, 2010.