Design and evaluation of an overload control system for crisis-related Web server systems

Andersson, Mikael; Höst, Martin; Cao, Jianhua; Nyberg, Christian; Kihl, Maria

Published in:
2006 International Conference on Internet Surveillance and Protection

2006

Design and Evaluation of an Overload Control System for Crisis-Related Web Server Systems

Mikael Andersson, Martin Höst, Jianhua Cao, Christian Nyberg and Maria Kihl
Department of Communication Systems
Lund University
Box 118, SE-221 00 Lund, Sweden
Email: mike|martinh|jcao|cn|maria@telecom.lth.se

Abstract

During recent years we have seen several large-scale crises. The 9/11 terror attacks, tsunamis, storms, floods and bombings have all caused a great deal of damage. A common factor in these crises has been the need for information and one important source of information is usually web sites. In this work we investigate and design an overload control system for web sites that are vital in crises. The overload control system uses content adaption to dynamically control web site performance.

1. Introduction

Recent years have displayed large-scale crises of several kinds. Since the terror attack in New York, September 11, we have seen tsunamis, storms, murders of political leaders and, in 2005, the July bombings in the London underground system. These events have in common that it is crucial to spread information about the crisis. Information has to be updated and reliable. Today many people turn to the Internet when they search for information and a crisis situation is no exception. The problem in a crisis situation is that the web servers behind the web sites get overloaded. For example; the Committee on the Internet Under Crisis Conditions: Learning from September 11 reports overload conditions from web sites during the reporting of the September 11 attack in New York [5].

Crisis-related research in Internet systems is new but nevertheless important. The End-to-End workgroup [2] stated in their 2005 vision for a different world of communications [11] that ”In 10 years, the network itself, and critical applications that run on it, should address the special needs that arise in times of crisis.” A crisis can informally be stated as when a combination of events, e.g. accidents and sabotage, results in a situation that negatively affects society in a way that hinders vital society functions. Examples of crises that are included in this definition might be terror attacks, storms, tsunamis, murders etc. Situations that are not considered crises according to this definition are for example the Olympic games, the Monday lunch rush hour, political elections etc. However, there is no clear line that divides crisis situations from normal situations.

There is a need to protect web servers with vital information from overload in a crisis situation. Research has been carried out for overload control in general for e.g. commercial web shops [7, 8, 10]. However, there are some differences between overload control in a crisis situation and in the general case. A major difference is that rejection of requests is less attractive in a crisis situation than in general. In a crisis situation it is probably better to provide less information to all clients than to reject requests for information.

An overload control mechanism without rejections can be based on so-called content adaptation. Content adaption is used as a technique to degrade web page content that is returned to visitors. The degradation of content can be achieved in numerous ways, depending on the structure of the web site in question. Images for example, tend to account for a high bandwidth in many cases, thus images can be a target of a content adapting system. If a web site is built on dynamic technology such as Java Server Pages or similar, one alternative is to replace dynamic pages with static versions during an overload situation. Another alternative is to reduce the number of web pages, by removing links to them from entry web pages.

Several papers have previously proposed content adaptation mechanisms for web servers. In [6] a content adaptation scheme is presented that uses pregenerated versions of the web site. The scheme switches between the versions depending on server load. In [13] a new framework is presented where the content is adapted to several types of clients. The focus is put on the client's ability to display and download content. In [12] an admission control scheme is presented were the web server supports different classes of clients. Clients can adapt the content by changing to another class of requests. However, there is a need for a content adaptation scheme that focuses on maintaining maximum server utilization and at the same time maximizes the utility of the delivered documents. Since, in a crisis, some documents are more requested, this should be taken into account.

The objective of this paper is to propose and evaluate an overload control mechanism for web server systems that deliver crisis related information. The overload control mechanism uses a dynamic content adaptation scheme that optimizes the information value of the requested pages.

Numerical investigations show the benefits of using content adaptation as a means of controlling overload. Empirical data from a web site delivering crisis-related content is used to evaluate the proposed scheme. The empirical data comes from web server access logs recorded during the bombings in the London underground system in July 2005 at the major Swedish news site Aftonbladet.se.

The rest of this paper is organized as follows; Section 2 describes the architecture of the overload control system, Section 3 contains an analysis of the optimization problem used in the system. Section 4 shows the numerical investigations of the system. Section 5 discusses the results, and conclusions can be found in Section 6.

2. Overload control system

In this paper, the target system is a web site that delivers crisis related information. This is normally a non-session based web site in the sense that a client does not log into the system, as is the case in e.g. web shops. It is assumed that the capacity bottleneck is the bandwidth, since mostly static pages will be delivered to the clients.

The overload control system includes three main components: The Monitor, the Optimizer and the Gate, see Figure 1. The basic principle of the control mechanism is to have pre-generated adapted versions of the web pages. The versions range from complete web pages to web pages with the highest possible rate of degradation. A degraded (adapted) web page should be less "expensive" to deliver than the complete version. During runtime the request for a web page will be intercepted and replaced with a more lightweight version. The adapted versions have to be pre-generated, because it would take too much time to generate them during runtime.

The Monitor constantly monitors the web server system. It measures the used bandwidth and the arrival traffic statistics.

The Optimizer decides at what service level requests for documents should be processed, by optimizing (when needed) according to the procedure described below.

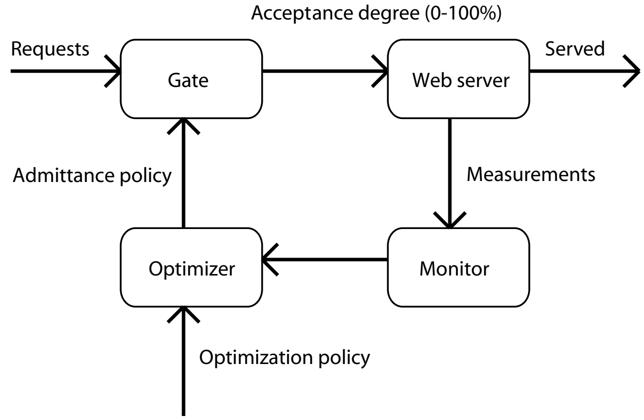

Figure 1: The overload control system

The Gate is the actuator in the overload control system that decides which document should be retrieved. The main idea is that when a certain document is requested, there are more than one version to choose from when the document is returned. The Gate intercepts the file request, and replaces it with a request for an appropriate file version.

3. Optimization procedure

The choice of which web page version to return at a certain time depends on the traffic at the web server. An optimization problem can be formulated that maximizes the utility of the returned files. Utility can be expressed in a number of ways, for example number of bytes or relevance to the crisis. In this section we describe a general optimization formulation and a specific problem, where a solution is proposed.

3.1. Documents and representations

Let $\mathcal{D}$ denote the set of documents at the web site. Let d_i, $1 \leq i \leq N$ denote the i:th document in $\mathcal{D}$, where N is the total number of documents at the web site. Documents are requested by the clients at a total rate of λ_{TOT} requests per second.

The probability that document d_i is requested is given by the document *popularity function*, $p(j)$. A document's popularity is the number of requests for the document divided by the number of requests to all documents in $\mathcal{D}$. Given λ_{TOT} and $p(j)$, the arrival rate λ_i for document d_i is given by

$$\lambda_i = \lambda_{TOT} \cdot p(j). \tag{1}$$

Each document consists of a set of objects (images, stylesheets etc). To enable content adaptation, several (degraded) *representations* are created for each document. Let r_i^j, $1 \leq j \leq N_i$ be the j:th representation of document d_i.

The representations of document i range from a minimal representation to the complete document representation.

Let $c(i,j)$ represent the *cost function* for representation r_i^j, which gives the *total cost*, C, during one second

$$C = \lambda_{\text{TOT}} \cdot \sum_{\forall i} p(i) \cdot c(i,j) \quad (2)$$

where the index j is the index for the selected representation for document d_i. The *budget*, denoted B, is the highest cost the system can "afford".

Representations should not be chosen arbitrarily. Instead, the representations should be chosen so that the amount of "values" is maximized. Let $u(i,j)$ represent the *utility function* for representation r_i^j. This means that the *total utility*, U, during one second is

$$U = \lambda_{\text{TOT}} \cdot \sum_{\forall i} p(i) \cdot u(i,j) \quad (3)$$

where the index j is the index for the selected representation for document d_i.

3.2. General problem

Given the definitions above, a general optimization problem can be formulated. The objective is to maximize the total utility U, subject to the cost C being smaller than the budget B.

maximize

$$U = \lambda_{\text{TOT}} \cdot \sum_{\forall i} p(i) \cdot u(i,j) \quad (4)$$

subject to

$$C = \lambda_{\text{TOT}} \cdot \sum_{\forall i} p(i) \cdot c(i,j) \leq B \quad (5)$$

$$1 \leq j \leq N_i$$

3.3. Optimization problem formulation

To be able to solve the general problem, the functions u and c needs to be defined. The utility function can be chosen in many ways, however, in a crisis, the perceived information value is more important than for example monetary value. In this paper, the utility function is defined by the information theoretical model for self information [15], which is given by

$$u(i,j) = log_2(b_i^j). \quad (6)$$

The model states that the amount of so called self information is the logarithm of the length of the message, which in this case is the byte count of the document representation.

In this paper, the cost function c is defined by the *byte count*. A representation consists of several objects, where the sum of the number of bytes of the objects is the representation's byte count. Documents cannot be degraded to any byte count for practical reasons, instead only certain byte counts are allowed. Let b_i^j, $1 \leq j \leq N_i$ be the byte count of representation r_i^j so that $b_i^1 \leq b_i^2 \leq \ldots \leq b_i^{N_i}$ and $c(i,j) = b_i^j$. The budget B is represented by the bandwidth of the system. Other possible choices for the cost function are for example the number of required CPU cycles to process the request or the number of required database accesses.

Given the definitions above it is now possible to formulate a specific optimization problem. Let $\hat{b}_i$ be the optimal byte count of the adapted document i. The optimization problem can be formulated as:

maximize

$$U = \lambda_{\text{TOT}} \cdot \sum_{\forall i} p(i) \cdot log_2(\hat{b}_i) \quad (7)$$

subject to ($\forall i$)

$$C = \lambda_{\text{TOT}} \cdot \sum_{\forall i} p(i) \cdot \hat{b}_i \leq B \quad (8)$$

$$\hat{b}_i \in \left\{ b_i^1, b_i^2, \ldots b_i^{N_i} \right\} \quad (9)$$

$$1 \leq i \leq N$$

3.4. Solutions

3.4.1. Benchmark solution

Assume that byte counts can be chosen arbitrarily as long as they are non-negative, that is

$$\hat{b}_i \geq 0 \quad (10)$$

instead of the constraint in (9). One of the solved problems in [14] resembles the optimization problem in this paper, given the constraint in (10) instead of (9). After rewriting *Task 1* in [14], problem (7) can be solved analytically as

$$\hat{b}_i^{\text{BENCH}} = \frac{\lambda_i \cdot p(i) \cdot B}{\lambda_i \cdot d(i) \sum_i \lambda_i \cdot d(i)} \quad (11)$$

$$U(\hat{b}^{\text{BENCH}}) = \sum_{\forall i} \lambda_i \cdot d(i) log_2(\hat{b}_i^{\text{BENCH}}) \quad (12)$$

This solution is optimal and yields a theoretical upper limit of total utility U. In the rest of the paper, the solution in Eq. (11) and Eq. (12) will be called *BENCH* and the solution will be used as a benchmarking comparison to a more practical solution. However, since any positive byte count is allowed the *BENCH* solution can assign illegal byte counts, i.e. higher than the maximum byte count $b_i^{N_i}$.

3.4.2. Bounded solutions

Since the *BENCH* solution can assign illegal byte counts, another solution is needed, where the byte counts are bounded by their minimum and maximum values.

Let instead

$$b_i^1 \leq \hat{b}_i^{\text{BOUND}} \leq b_i^{N_i} \tag{13}$$

replace the constraints in Eq. (9). This problem is a non-linear and convex optimization problem. It can be solved in reasonable time by using a solver; there are several solvers available. In Section 4 the SNOPT solver [4] was used to optimize the byte counts. This solution is referred to as *BOUND* and the optimal byte counts are called $\hat{b}_i^{BOUND}$. The *BOUND* solution is still not suitable for use in an implemented system because, even if byte counts are assigned within their bounds, they are assigned in a continuous space. In practice, byte counts need to be set to a discrete set of choices, as defined in (9). A sub-optimal solution is therefore introduced, called *BOUNDDISC*, where the optimal byte counts are discretized onto allowed byte counts.

Because there might not be a byte count that exactly fits the optimal byte count, the idea is to assign a proportion α of the document requests to the nearest lower byte count, and a proportion β of the document requests to the nearest higher byte count. The proportions should be distributed so that the average byte count is the optimal byte count. Let b_i^{low} and b_i^{high} be the nearest surrounding byte counts;

$$\hat{b}_i^{\text{BOUND}} = \alpha \cdot b_i^{\text{low}} + \beta \cdot b_i^{\text{high}}; 1 = \alpha + \beta$$

which gives

$$\alpha = \frac{b_i^{\text{high}} - \hat{b}_i^{\text{BOUND}}}{b_i^{\text{high}} - b_i^{\text{low}}}; \beta = \frac{\hat{b}_i^{\text{BOUND}} - b_i^{\text{low}}}{b_i^{\text{high}} - b_i^{\text{low}}}.$$

4. Quantitative performance evaluations

The numerical investigations were based on the empirical data described below. The objective of the investigations was to show the advantage of content adaptation over pure admission control, to show the impact of the number of possible byte counts per document and validate that the constraints are kept at overload.

In all investigations, the bandwidth constraint was set to 1 Gbits per second, as was the case for the site from which the web server access logs were taken.

4.1. Empirical data

Empirical data was collected from Aftonbladet.se [1], which is the largest news site in Sweden with several millions visits per week [3]. The logs from Aftonbladet.se are more extensively analyzed in [9]. The data was used to get representative input to the optimization problem. The web site consists of several identical web caches in front of ordinary web servers. The logs in this work come from one of the web caches. The web pages consist of mostly static content. The content is continuosly updated in the web caches.

The web server access logs from the reporting of the London underground bombings were recorded during 26 hours, starting 2.15 PM July 7th, 2005, and showed 38 million file requests for, in total, 200 gigabytes of data.

In this paper, document requests are defined as those requests that start with an HTML request. However, some commercial advertisment parts that are included in article documents also appear in pure HTML format. These commercial advertisement requests were not included in the definition of documents. With this definition, documents can be seen as requests for articles on the news site. Document popularity corresponds to the relative number of requests a certain document receives at the web site. The access logs were split into 15 minute intervals where document popularity was collected for each document and interval.

Document popularities and document sizes were calculated from the first 15 minutes of the recorded logs. Minimum document sizes were set to 1460 bytes, representing the maximum HTTP data length in a standard Ethernet frame. Returning web pages that fit into a single IP packet was, for example, used by CNN during the 9/11 crisis [5].

The number of representation byte counts in the *BOUNDDISC* solution were set to five for each document. The minimum document size, the maximum document size were used, and three additional byte counts were distributed in an equi-distant manner (with respect to byte counts) between minimum and maximum document sizes.

4.2. Investigations

The *BENCH* and *BOUNDDISC* solutions were compared to a pure admission control mechanism (called *ADM*) without content adaptation. The admission control mechanism throttles the document requests so that some documents get rejected. All requests are rejected with the same probability, and the objective of the admission control mechansism is to maintain the maximum bandwidth. Therefore, the rate of accepted requests, $\lambda_{ACCEPTED}$, is given by

$$\lambda_{\text{ACCEPTED}} = \frac{B}{\sum_{\forall i} p(i) \cdot b_i^{N_i}} \tag{14}$$

All solutions were first of all evaluated with respect to throughput, measured in bytes per second. For a solution to work properly, the throughput at any arrival rate must be below or equal to the bandwidth. Because the goal of the content adaptation is to maximize the utility of the returned documents, the total utility was measured for each

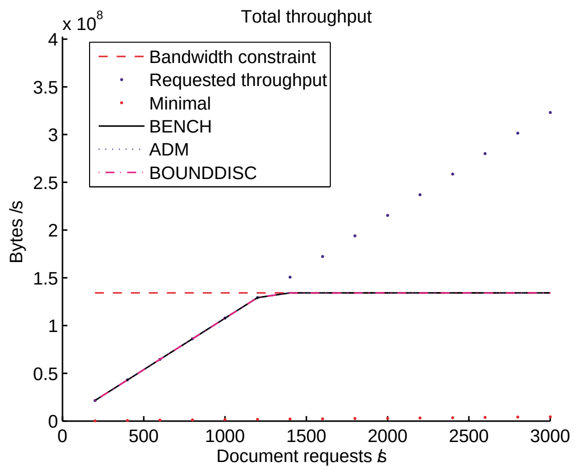

Figure 2: Throughput of the different solutions.

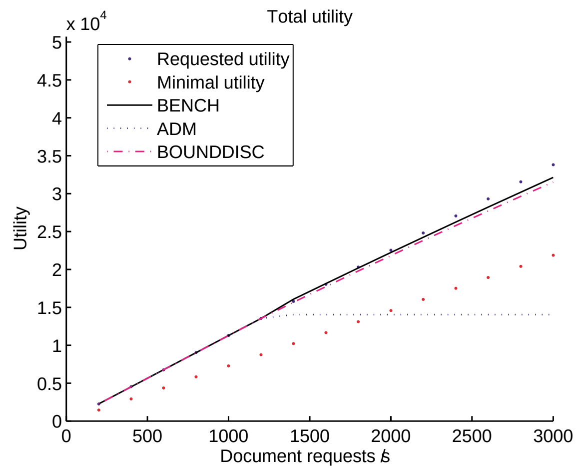

Figure 3: Utility of the different solutions.

solution and investigated. Also, the utility for varying numbers of possible document byte counts was investigated. The *BOUNDDISC* solution gives sub-optimal results depending on the number of possible byte counts. It is therefore valuable to investigate the impact of adding more or less document byte count possibilities.

5. Results

5.1. Throughput

First, the throughput is considered. Figure 5.1 shows the throughput for the different solutions. Also, the bandwidth constraint of 1 Gbits per second is shown. The figure shows the point where the web server gets overloaded, at approximately 1250 document requests per second. The dotted curves for requested throughput and minimal throughput represents the cases where maximum and minimum byte counts are returned. Since the *BENCH* and *ADM* solutions are constrained, they follow the bandwidth constraint after the point of overload. *BOUNDDISC* instead degrade the throughput since the solution is adjusted to represent feasible byte counts.

5.2. Utility

Figure 5.2 shows the solutions' total utility, U. The *BENCH* solution actually yields more utility than requested, because byte counts may be larger than feasible. The figure also demonstrates the advantage of content adaptation over admission control; the *ADM* solution does not improve total utility after the point of overload. The *BOUNDDISC* solution on the other hand continues to deliver increasing utilities, not much less than the *BENCH* solution.

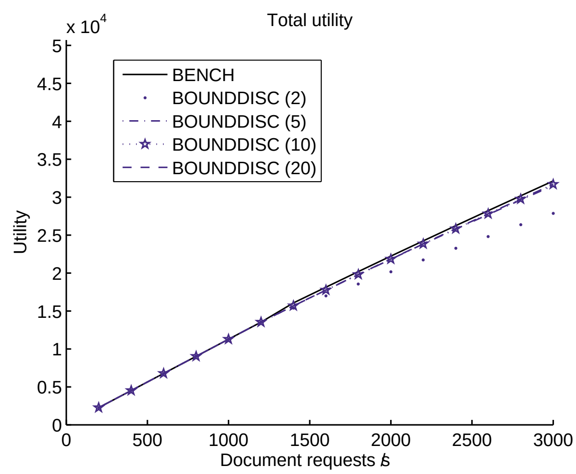

Figure 4: Obtained utility using solution given 2, 5, 10 and 20 possible byte counts (BOUNDDISC solution), compared to the BENCH solution.

5.3. Impact of number of representations per document

Figure 5.3 shows the impact of the number of possible byte counts for the documents. As can be seen, using only two document byte count possibilities (minimum and maximum) renders significantly lower utilities. However, the difference between using 5, 10 and 20 possibilities is quite small. This means that a system will be able to use content adaptation with a small number of representations for each document.

5.4. Implementation issues

The optimization problem must be solved in real-time by the Optimizer in order to find the optimal representation of each document. As mentioned before, the system makes use of pre-generated document representations, because it would be too demanding to create on-the-fly representations. This paper uses byte counts that are distributed in an equi-distant manner but that is not a necessity. In order not to increase the tasks for the web site administrator, some tool should be added in the content management system behind the web site. The tool could automatically create adapted versions of every document that is uploaded to the web server. Methods that can be used for reducing document sizes are for discussed in e.g [6].

The Monitor periodically monitors the web server for changing input traffic. The Optimizer re-optimizes the byte counts whenever needed. The optimization can obviously not be performed too often since it requires computational efforts in the web server. However, it needs to be updated whenever a new document is uploaded or removed in the server. The Gate, the Optimizer and the Monitor are suitable for inclusion in an Apache module. Since an Apache module works "inside" the Apache source code, the extra processing needed for the content adaptation system is minimized. Also, within a module, because of the closeness to the Apache kernel, it is easy to monitor throughput and input traffic patterns.

6. Conclusions

This paper has presented and evaluated an overload control system to be used in web servers that delivers crisis-related information. The overload control system uses a dynamic content adaptation scheme that optimizes the information value of the returned web pages by adjusting the document byte counts, in order to keep the throughput below the bandwidth.

The numerical investigations were based on empirical data from Aftonbladet.se, the major news site in Sweden, collected during the reporting of the July underground system bombings in London. The findings show that the proposed *BOUNDDISC* solution, even though truncated, performs well under overload and is superior to common admission control. Also, the numerical investigations showed that a large variety of document representations is not needed, as shown above, five different representations per document did not show significantly lower results than ten or twenty.

Finally, implementation issues were discussed where a number of techniques were presented to use in the content adaptation system.

Acknowledgment

This work has been funded by the Swedish Emergency Management Agency (SEMA). The authors also wish to express their gratitude to Aftonbladet.se, especially Jesper Wahlberg, for contributing the access logs from Aftonbladet.se. Finally, the authors thank Pål Nilsson for discussions on optimization.

References

[1] Aftonbladet. `http://www.aftonbladet.se`.

[2] Irtf. `http://www.irtf.org`.

[3] Kiaindex - insight xe, 2005 v36, unika webbläsare. `http://www.annons.se/?get=content&action=view&id=127-98`.

[4] Snopt. `http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm`.

[5] Internet under crisis conditions: Learning from september 11. *National Research Council*, 2003.

[6] T. Abdelzaher and N. Bhatti. Web content adaptation to improve server overload behavior. *Computer Networks*, 31(11-16):1563–1577, 1999.

[7] T. Abdelzaher, K. Shin, and N. Bhatti. Performance guarantees for web server end-systems: a control theoretic approach. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):80–96, January 2002.

[8] M. Andersson. Performance modeling and control of web servers. Technical Report 160, Department of Communication Systems, Lund Institute of Technology, 2004. Lic. Thesis.

[9] M. Andersson, A. Bengtsson, M. Höst, and C. Nyberg. Web server traffic in crisis conditions. In *In Proceedings of the Swedish National Computer Networking Workshop, SNCNW 2005*, 2005.

[10] L. Cherkasova and P. Phaal. Session-based admission control: A mechanism for peak load management of commercial web sites. *IEEE Transactions on computers*, 51(6):669–685, June 2002.

[11] D. D. C. et al. Making the world (of communications) a different place. Technical report, End-to-End Research Group, 2005.

[12] S. C. Lee, J. C. Lui, and D. K. Yau. A proportional-delay diffserv-enabled web server: Admission control and dynamic adaptation. *IEEE Transactions on Parallel and Distributed Systems*, 15(5), 2004.

[13] R. Mohan, J. Smith, and L. Chung-Sheng. Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia*, 1(1):104–114, 1999.

[14] P. Nilsson and M. Pióro. Solving dimensioning tasks for proportionally fair networks carrying elastic traffic. *Performance Evaluation*, 49:371–386, 2002.

[15] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 1948.