Performance Tuning and Cost Discovery of Mobile Web-Based Applications

Matthias Book, Volker Gruhn, Malte Hülder, André Köhler
Chair of Applied Telematics/e-Business, University of Leipzig
Klostergasse 3, 04109 Leipzig, Germany
{book, gruhn, huelder, koehler}@ebus.informatik.uni-leipzig.de

Abstract

When considering the addition of a mobile presentation channel to an existing web-based application, a key question that has to be answered even before development begins is how the mobile channel's characteristics will impact the user experience and the cost of using the application. If either of these factors is outside acceptable limits, economical considerations may forbid adding the channels, even if it would be feasible from a purely technical perspective. Both of these factors depend considerably on two metrics: The time required to transmit data over the mobile network, and the volume transmitted.

The PETTICOAT approach presented here provides decision makers with indicators on the economical feasibility of mobile channel development. In a nutshell, it involves analyzing interaction patterns on the existing stationary channel, identifying key business processes among them, measuring the time and data volume incurred in their execution, and then simulating how the same interaction patterns would run when subjected to the frame conditions of a mobile channel. As a result of the simulation, we then gain time and volume projections for those interaction patterns that allow us to estimate the costs incurred by executing certain business processes on different mobile channels.

1. Introduction

As thin client applications, web-based applications have the advantage of independence from the user and his preferred device. Just the existence of a browser and a suitable network connection are needed. Thus, web-based applications seem to be convenient for mobile use. But in hands-on trials of such scenarios, the response time of the application is often notably worse compared to its use in a LAN environment. Beyond, an unpredictable amount of mobile radio costs occur. A company that plans to provide mobile access to its web-based applications for a large group of mobile workers needs detailed information about the behavior and the estimated costs of the application in a mobile environment – before investing any effort in building it. Therefore, the expected performance as well as the expected costs of the application on different mobile radio networks need to be quantified at an early stage. With the PETTICOAT (**PE**rformance **T**uning and cos**T** d**I**s**C**overy of m**O**bile web-based **A**pplica**T**ions) approach, we present a procedure that can be used in the outlined situation.

The procedure can be conducted by software developers as well as software project managers. During the execution of PETTICOAT, indicators are calculated that reveal the application performance as well as the radio network costs in the mobile environment. Based on these results, a decision about the possible mobile use of the application can be made. If the application is classified as not suitable for mobile use, the project manager can decide whether it is reasonable to fine-tune the application's performance. Therefore, PETTICOAT shows the specific optimization potentials of the application. In order to implement this potential, the tuning can be conducted for single features or the whole application.

In the following sections, the PETTICOAT approach is presented in more detail. In chapter 2, the related work regarding this topic is described. Chapter 3 presents the approach itself. We show how to model the application structure as a dialog flow (3.1), the identification of typical user paths (patterns of workflows) through the application (3.2), the measurement of data volume in the LAN environment (3.3) and the simulation of the application within different mobile networks (3.4). Chapter 4 gives a short summary and describes the next steps in the PETTICOAT project.

2. Related Work

In [6], it is shown how frequent and thus critical user paths can be identified in e-commerce applications. The authors provide a model of the user behavior in form of session graphs and conduct analyses regarding the most frequently used user paths as well as critical edge sequences. This technique could be quite useful for our approach, because the identification of the most frequently used subset of all possible user paths in the application model is needed.

Furthermore, there are lots of approaches for web log analysis aimed at classifying user paths, e.g. [14], [3], [9], [8], [5], [7]. Especially the identification of long sequences [13] seems to be an important topic for the PETTICOAT concept. The identification of actually choosen user paths vs. all possible user paths in the application model is needed in order to obtain meaningful results from the following simulation. In this context, the work of [12] is of specific interest. They present a notion for a cluster-based online monitoring system for web traffic. The target-oriented analysis of web traffic is a task to be solved within the PETTICOAT approach.

As PETTICOAT particularly addresses the analysis of dynamic web applications instead of static web pages, the analysis of web traffic is even more difficult. This problem is adressed e.g. in [2], which deals with dynamic web content generation and website analysis.

Other approaches to improving the performance of web-based applications have focused on using thin clients to transmit just the image of the application (see e.g. [11]). The findings of this work are of relevance for the deduction of consequences (application design, bandwidth restriction) based on the simulation results. In this context, [1] and [10] report interesting results from an analysis of large websites regarding performance, cache and cookie issues. These results could be used for the creation of a package of measures in order to modify the analyzed website regarding performance issues in the mobile environment.

3. The PETTICOAT Approach

In a nutshell, the PETTICOAT approach uses a dialog flow model and web server log files to identify user paths and compile volume statistics, which are then run through a tool that simulates the volume and time that would be incurred on a mobile channel by the observed interaction patterns. From the simulated volume and time data, we can then calculate the cost of accessing the application on a mobile channel (Fig. 1).

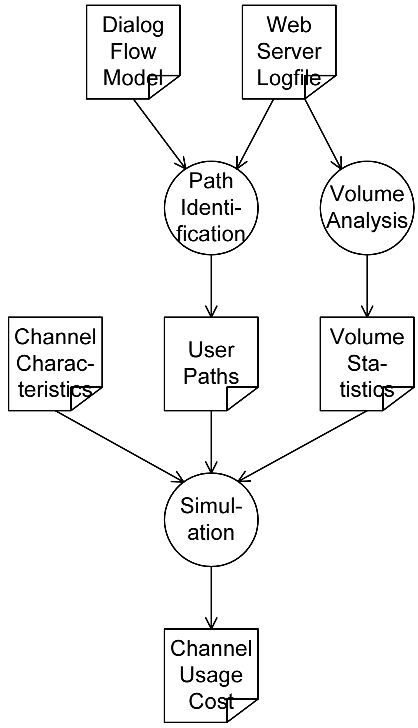

Figure 1. The PETTICOAT approach.

3.1. Modelling the Dialog Flow

As a basis for the identification of interaction patterns, we need a model of the application's complete dialog structure. We use the Dialog Flow Notation (DFN) [4] for this purpose. This graphical notation models an application's dialog flow as a directed (yet not bipartite) graph of states that are connected by transitions. We call the transitions *events* and the states *dialog elements*, distinguishing *masks* (web pages rendered on the client) and *actions* (business logic executed on the server). Events can carry parameters that transport business data such as form input.

By building such dialog graphs from masks, actions and events, the developer can specify all possible user interactions with the application. To increase the expressive power of the specification, dialog graphs can be encapsulated in *dialog modules* that can be reused in different contexts within the same application by nesting them into the dialog flow at arbitrary levels. This allows the developer to model complex dialog structures that closely mirror the users' mental model of the complex business processes supported by large-scale web applications.

The DFN can be used both constructively to model an application's dialog flow prior to building it, or descriptively to document an existing application's dialog flow. In order to use these graphical specifications as input for the following steps, they can be easily translated into the XML-based Dialog Flow Specification Language (DFSL).

3.2. Identifying Interaction Patterns

The dialog graphs of an application specify all possible ways of interaction that the user interface allows. In the multitude of possible traversal paths, there will typically be some more and some less frequently traversed paths. Also, the same business process may be accomplished in two similar, but still different ways. To arrive at a representative cost projection of executing individual business processes, we therefore need to analyze the actual interaction patterns that occur in the application. In order to weigh the cost incurred on the various paths, we not only need to identify the paths that the users traverse most frequently, but also assign probability values to any branches based on accumulated interaction data.

The interaction patterns can basically be identified automatically based on user tracking information that is routinely collected in web server log files, as suggested in [13]. In complex web applications, however, the logged URLs may not always indicate unambiguously which page was ultimately presented to the user. To increase the quality of the path identification, it may be necessary to log interaction data directly in the dialog control logic instead at the web server level. We're currently investigating ways of accomplishing this, ideally with non-invasive methods that do not require changes to the existing application logic.

3.3. Measuring Time and Data Volume

As mentioned above, the two main factors influencing the cost of interaction with an application over a mobile channel are the time spent online and the data volume transmitted. To project these metrics on mobile channels, we must measure them on the existing stationary channel and then input them into the simulation.

However, there are a few challenges in the details of this measurement process that we are currently working out: Most importantly, for the volume measurement, we need to distinguish between static and dynamic content. While static content (such as images) always incurs the same volume[1], dynamic content (such as search results pages) can produce a different volume with each request. To obtain accurate estimates, we need to deduce a probability distribution or an average value from the accumulated volume data. Also, web server log files only log the net volume of the content, but not any overhead introduced on lower levels of the protocol stack that still counts for billing purposes. This overhead can either be ascertained by observing the data flow directly on a sufficiently low protocol level instead of relying on server log files, or by factoring it into the simulation in close accordance with the respective protocol specifications.

The time measurement is subject to even more influence: While mobile channel providers typically bill the entire time that a device is online, this time is the sum of user activity (e.g. filling in forms), upstream and downstream transmission time, channel latency and server processing time. To accurately distinguish all these contributing factors, we would need synchronized timing on both the server and the client. Fortunately, however, only the user and server activity matter for the subsequent simulation, since the observed transmission time and latency already depend on the stationary channel that we measured on. We can thus deduct them from the overall time during the simulation based on our knowledge of the stationary channel characteristics and volume transmitted. This way, we are left with the user and server activity time, to which we can add the newly calculated transmission time and latency based on the mobile channel's characteristics. The refinement of these algorithms is a focus of our ongoing research.

1 apart from caching effects, but those can be accounted for with a certain percentage in the simulation

3.4. Simulation of Mobile Channels

After having analyzed the most important user paths and the volume and time incurred on them, we can now simulate the same interaction patterns on other mobile channels, i.e. other communications networks. In order to perform such a simulation, we developed a tool which is fed the data gained during the previous steps – a description of the dialog flow, the observed interaction patterns and the volume measurements. As additional input, the simulator also requires the mobile communication channel's profile, i.e. its characteristic values for bandwidth, latency, billing model etc.

The tool performs the simulation for the interaction paths of each business process by calculating the time it takes to load the specified masks in the order indicated by the interaction patterns, taking into account that loading inline objects (e.g. images embedded into the page) may cause latency and network traffic. The simulator also takes into account how long a user works with this mask on average before the request for the next mask is simulated. All the results produced during the simulation are saved for later inspection.

Inspecting the simulation results yields important insights into the structure of the application. The results indicate how long it takes to load certain masks, and how much data is transferred. If the profile for the communication channel is specific enough, we can also see how much is charged by the communication provider not only for the complete process, but also for any single mask. From the simulation results we can also deduce where certain lags occur, for example, where the application has to wait for an inline object to be loaded. This may indicate a different implementation of the according mask is advisable, so that the request to load the inline object is sent earlier, trying to make network latency occur at a different time.

It can also be tested if certain constraints are met, for example if a constraint is given that mask x has to be loaded within y seconds after leaving mask z, this can be checked by examining the simulation results. The insights into the structure of the application named above may help to find ways to redesign the masks that are responsible for failing the constraints. If no redesign seems feasible, the application cannot be used on this communication channel according to the specified constraints.

Simulating different communication channels allows developers to figure out which channels can be used without making changes to the application, because all constraints are met. It also allows to determine the cost of using the application on a certain channel.

3.5. Experience

We already applied the PETTICOAT method to the prototype of a web-based offer management system of an insurance company in order to estimate the cost that will be incurred each month by field staff accessing the system using mobile devices, and to determine the most suitable mobile network technology. We found that UMTS with a volume-based pricing plan would be the best solution for the interaction patterns we examined for that system. The use of GPRS would be cheaper by a negligible fraction, but incur latency times that are 2.5 times as high as those experienced with UMTS, and thus yield worse usability. According to the simulations, time-based pricing plans for HSCSD and GSM would have resulted in costs that are 5.5 (HSCSD) or even 7.6 times (GSM) as high as the volume-based UMTS plan, which therefore turned out to be the recommended mobile network technology in our case study.

4. Conclusion and Further Research

In this paper, we have shown the necessity of analyzing given web applications to assess the feasibility of adding mobile channels: From the analysis, information can be gained that allows us to draw conclusions on whether the application can be used on certain mobile communication channels. Thus, we can tell if using a given application in a mobile environment is feasible or what has to be done in order to reach this goal. The simulation tool also allows to make assumptions about the costs of using the application in a mobile environment. This helps to decide whether introducing a mobile channel will pay off for an enterprise in the future.

In our future work, we will focus on automated analysis of web applications, leading to more easily generated application profiles. We will also take a look at the possibility of revealing business processes automatically from the usage data, rather then explicitly defining the dialog flows that belong to a business process.

If the necessary profiles used by the simulation tools are specific enough, it should also be possible to calculate the interesting values, rather than gaining them from simulation. This could be a faster approach and might also lead a better usability as the interesting results can be calculated on demand, instead of parsing them from a log file.

5. Acknowledgements

The Chair of Applied Telematics/e-Business is endowed by Deutsche Telekom AG.

References

[1] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao. Characterization of a large web site population with implications for content delivery. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 522–533. ACM Press, 2004.

[2] B. Berendt and M. Spiliopoulou. Analysis of navigation behaviour in web sites integrating multiple information systems. *The VLDB Journal*, 9(1):56–75, 2000.

[3] P. Berkhin, J. D. Beche, and D. J. Randall. Interactive path analysis of web site traffic. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 414–419. ACM Press, 2001.

[4] M. Book and V. Gruhn. Modeling web-based dialog flows for automatic dialog control. In *19th IEEE International Conference on Automated Software Engineering (ASE 2004)*, pages 100–109. IEEE Computer Society Press, 2004.

[5] E. H. Chi, P. Pirolli, and J. Pitkow. The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a web site. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 161–168. ACM Press, 2000.

[6] K. Dutta, D. VanderMeer, A. Datta, and K. Ramamritham. Discovering critical edge sequences in e-commerce catalogs. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 65–74. ACM Press, 2001.

[7] M. Gillenson, D. L. Sherrell, and L. da Chen. A taxonomy of web site traversal patterns and structures. *Commun. AIS*, 3(4es):5, 2000.

[8] J. Heer and E. H. Chi. Separating the swarm: categorization methods for user sessions on the web. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 243–250. ACM Press, 2002.

[9] D.-H. Kim, V. Atluri, M. Bieber, N. Adam, and Y. Yesha. A clickstream-based collaborative filtering personalization model: towards a better performance. In *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 88–95. ACM Press, 2004.

[10] B. Krishnamurthy and C. E. Wills. Analyzing factors that influence end-to-end web performance. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 17–32. North-Holland Publishing Co., 2000.

[11] A. M. Lai, J. Nieh, B. Bohra, V. Nandikonda, A. P. Surana, and S. Varshneya. Improving web browsing performance on wireless pdas using thin-client computing. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 143–154. ACM Press, 2004.

[12] Y. Mao, K. Chen, D. Wang, and W. Zheng. Cluster-based online monitoring system of web traffic. In *WIDM '01: Proceedings of the 3rd international workshop on Web information and data management*, pages 47–53. ACM Press, 2001.

[13] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems*, 1999.

[14] M. Spiliopoulou. Web usage mining for web site evaluation. *Commun. ACM*, 43(8):127–134, 2000.