FirstSpirit™
Your Content Integration Platform

UX-Bridge Technisches Datenblatt

Version	**1.0**
State	**RELEASED**
Date	**2013-02-19**
Department	Product Management
Author/ Authors	C. Feddersen
Copyright	2012 e-Spirit AG
File name	UX-Bridge_Technical_Datasheet_DE

e-Spirit AG

Barcelonaweg 14
44269 Dortmund | Germany

T +49 231 . 477 77-0
F +49 231 . 477 77-499

info@e-Spirit.com
www.e-Spirit.com

Table of contents

1 Introduction

This technical datasheet only applies to the current release of UX-Bridge Release Version 1.0.

In Chapter 2, it describes the components required for operating the UX-Bridge module and their system requirements. Chapter 3 provides some notes on sizing derived from the results of the load tests. These are found in Chapter 4.

2 System requirements

2.1 FirstSpirit

The UX-Bridge module was developed and tested on FirstSpirit 5.0. Therefore FirstSpirit Version 5.0 is recommended. It is compatible with FirstSpirit 4.2R4, however, with the following exception:

- In FirstSpirit 4.2R4, UX-Bus cannot be run on the internal Jetty server of the FirstSpirit server. More information about installing UX-Bus on a FirstSpirit server can be found in Chapter 2.1.2 of the UX-Bridge installation instructions.

System requirements for installing FirstSpirit can be taken from the technical datasheet for FirstSpirit.

2.2 UX-Bus

The system has to meet the following requirements in order to operate the UX-Bus.

2.2.1 Hardware

- 100 MB of free hard drive space.
  Depending on the configuration (persistence) and operation (master/slave), significantly more disk space may be needed.
- 2 GB of RAM

2.2.2 Operating system

- Windows Server 2008 R2
- AIX V6.1+7.1
- Solaris 10+11
- Red Hat Enterprise Linux 5+6
- Debian Linux 5+6

2.2.3 Java environment

- Oracle Java 1.6.0_18 or higher

3 Sizing

The minimum requirements listed above should be sufficient for all scenarios in which the UX-Bus is primarily used for writing the data from FirstSpirit into one or more content repositories. One such use case was tested as part of the load tests. The results can be found in the next chapter.

If the UX-Bus is to serve as a central message component within the website infrastructure, then more resources may be needed. This may apply to scenarios in which one message is sent to the UX-Bus per web application request, for example, to analyze user behavior. You can find details on scaling the UX-Bus in the UX-Bridge Installation Manual, Chapter 3.3 High availability. For such scenarios, a corresponding load test should always be carried out within the project in order to ensure that the requirements for performance and scalability are also met.

4 Results of the load test

Load tests were conducted as part of developing UX-Bridge. The objective was to develop a scenario as realistic as possible that simulates a specific use of UX-Bridge as part of a website that contains both dynamic and pre-generated content.

4.1 Test scenario

4.1.1 Editing system

The scenario operates on the basis that 100 editors are working on the FirstSpirit Client simultaneously. In the process, 1,000 changes are created in the system per day (new contents or editing existing contents). In addition, there is an automatic

content import mechanism, which also generates 1,000 changes per day. All changes are released and deployed via a workflow. Generation includes pre-generated HTML files that update the necessary assets (graphics, CSS, JS) and the messages required for updating the files in the content repository.

4.1.2 Website

The project developed in the "News scenario" tutorial was used as the website. Details on this can be found in the developer documentation, Chapter 4.2. This is a hybrid Internet presence with dynamic and pre-generated parts. The news area makes use of a dynamic web application that shows the current press releases to the user and enables dynamic filtering by categories. The remaining pages contain pre-generated HTML.

It is assumed that the dynamic web application accounts for 30% of the page views. The remaining 70% are views of pre-generated pages. This appears to be realistic, because the actual press release is a pre-generated page, and only the overview of the press release is delivered dynamically.

In order to model user behavior as precisely as possible, the test scenarios contain corresponding "think times". These were extrapolated from real web surfing behavior. The length of time spent on a page varies between 1 to 4 seconds.

A simulated user always surfs multiple pages per visit. During a visit, a browser cache is simulated so that design elements, JavaScript, and CSS do not have to be requested again. In order to simulate browser behavior, a maximum of 5 simultaneous requests per user are generated to reload resources. The cache is emptied again after ending such a session.

On average, a requested website results in 44 HTTP requests with an average transfer volume of 180 KB. If the files are in the browser cache, the transfer volume is reduced to 25 KB on average.

Up to 5,000 simultaneous users on the website are simulated during the test. The content modifications by the editing system are made during the test at the same time as the load tests on the web server.

4.2 Test infrastructure

In order to be able to scale the number of users easily, the entire test was conducted on Amazon Cloud infrastructure. In doing so, all components were purposely

installed on dedicated instances to be able to identify any bottlenecks more easily.

You can find details on Amazon EC2 instance types, visit http://aws.amazon.com/de/ec2/instance-types/. You will also find information there about which hardware roughly corresponds to the instances. An EC2 compute unit offers CPU capacity equivalent to a 1.0-1.2 GHz Opteron or Xeon processor from 2007.

The participating systems were:

FirstSpirit-Server: m1.small with FirstSpirit 5.0.102.53034

UX-Bus: m1.small with ActiveMQ 5.6.0

Webserver: m2.4xlarge with nginx 1.0.15

Tomcat for web application: c1.xlarge

Tomcat for adapter: m1.small

jMeter (for user simulation): m1.xlarge

MySQL for FirstSpirit content sources: Amazon RDS instance of type db.m1.small with MySQL 5.1.61

MySQL instance as content repository: Amazon RDS instance of type db.m1.small with MySQL 5.1.61

Each system (except for the jMeter instances for the client simulation) was present only once; in other words, no clustering of any kind has been carried out yet to improve performance here.

The web server was configured as a proxy for Tomcat. Since the dynamic web application does not deliver any personalized contents, the web server can cache the contents gained from Tomcat for a maximum of one minute. If the editing system modifies content during this time, the cache is invalidated beforehand accordingly, so that the modification immediately becomes visible on the website.

The number of website visitors was increased to 5,000 simultaneous users in a ramp-up phase lasting 30 minutes. This load was maintained for 1 hour and followed by a ramp-down phase lasting 10 minutes. There were no ramp-up or ramp-down phases for content modifications in the editing system. Contents were created constantly and then a deployment was carried out in this case.

All relevant measurement data was acquired by a number of tools and centrally

collected and visualized in a Graphite instance (see http://graphite.wikidot.com/).

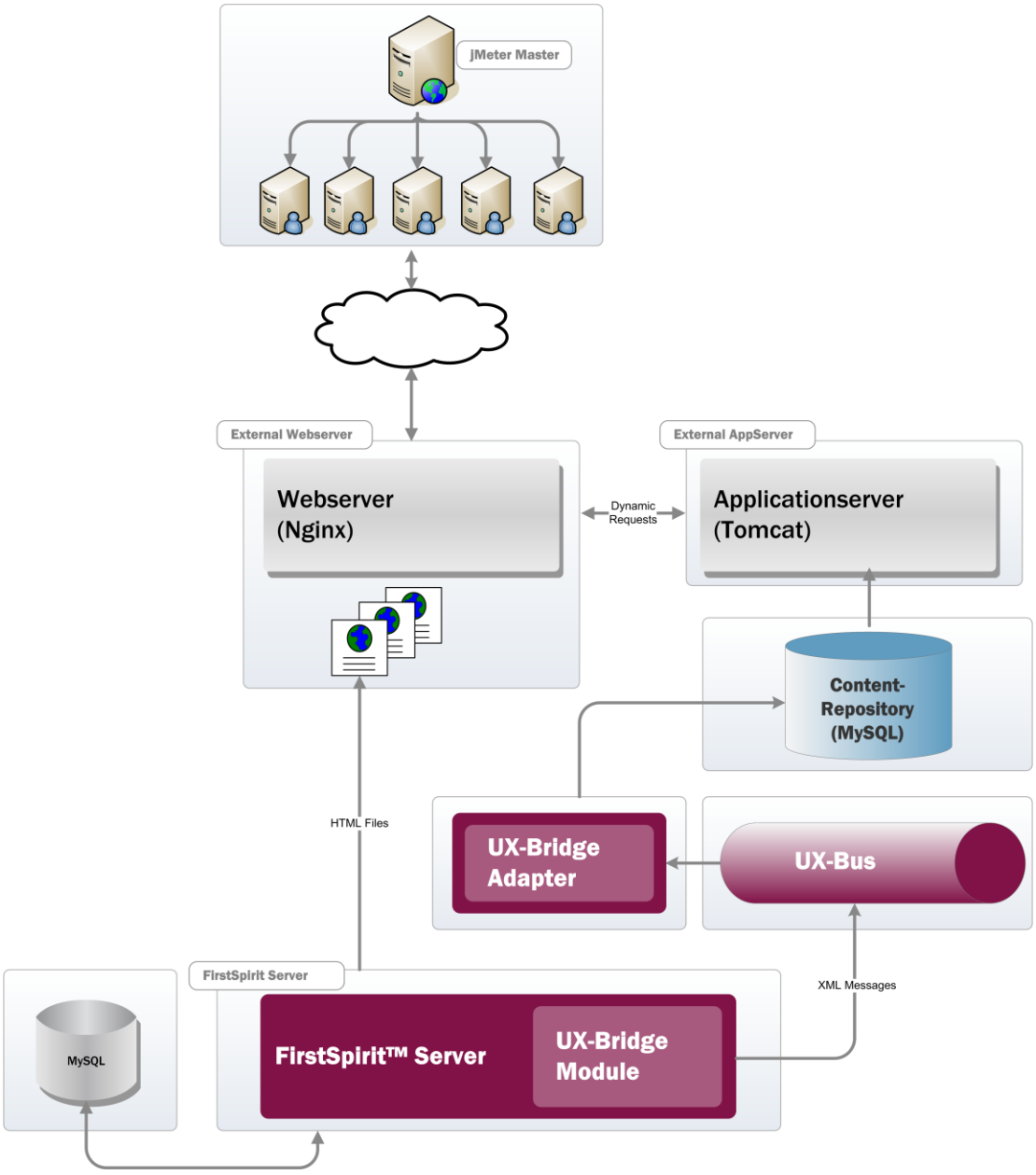


Figure: Load test infrastructure

4.3 Results

Analysis of the measurement data gained led to the following insights:

- When under a full load of 5,000 users, the web server responded to approximately 13,900 requests per second. If this is extrapolated to one day, it corresponds to traffic of about 27 million page views per day. The pages,

including all resources, were available on the client within 117 ms on average. Thus the UX-Bridge can also be used in scenarios with high access statistics.

- Even if there is a very high load on the website, there is no adverse effect for the editors. Content deployment times are constant throughout the entire test run. This can be traced back to the strict separation of the editing system from the live systems. Inversely, frequent content modifications do not affect the performance of the website.

- The high number of editorial content modifications and deployments poses no problems for the UX-Bridge, even in the standard configuration. All modifications were available on the website and in the content repository of the UX-Bridge within 357 ms on average. In many customer projects, significantly fewer modifications and deployments are made per day.

- If caching of the application data is not possible in projects, horizontal scaling of the application servers is advisable, since experience shows that these form the first bottleneck. This applies all the more if the amount of traffic on the dynamic part of the website is higher than assumed in the load test.

In the appendix you will find the essential key data in graph form.

Y-axis, left: Units are milliseconds

Y-axis, right: Requests per second or the number of simulated users

X-axis: Time

4.3.1 UX-Bridge deploy time

This graph shows the time required to write the content modification into the content repository. The time between the release and writing into the content repository was measured. The left y-axis applies.

The spike right at the beginning can be explained by the overhead of the initial connection setup for the individual components. It can be clearly seen that the load on the web servers has no impact on this metric.

4.3.2 Average response time (jMeter)

Average response time to all requests for websites (30% dynamic, 70% static). The value refers to the complete page, including Javascript, style sheets, and graphics.

The left y-axis applies.

Slightly longer response times can be identified only as of approximately 4,000 simultaneous users. At the same time, an increase in the load on the jMeter instances and the web server can be observed. Even so, the load for all machines is still in the acceptable range. Moreover, the web server statistics indicate that, as of this number, the web server's connection handling results in an adverse effect on response times.

4.3.3 Web server req/sec:

Number of HTTP requests per second that arrive at the web server. The right y-axis applies.

The amplitudes under full load still correlate to an increased load on the jMeter. This was caused by the variable "think times". The measurement data shows that significantly more requests were generated at this time by jMeter instances.

4.3.4 Concurrent users (jMeter)

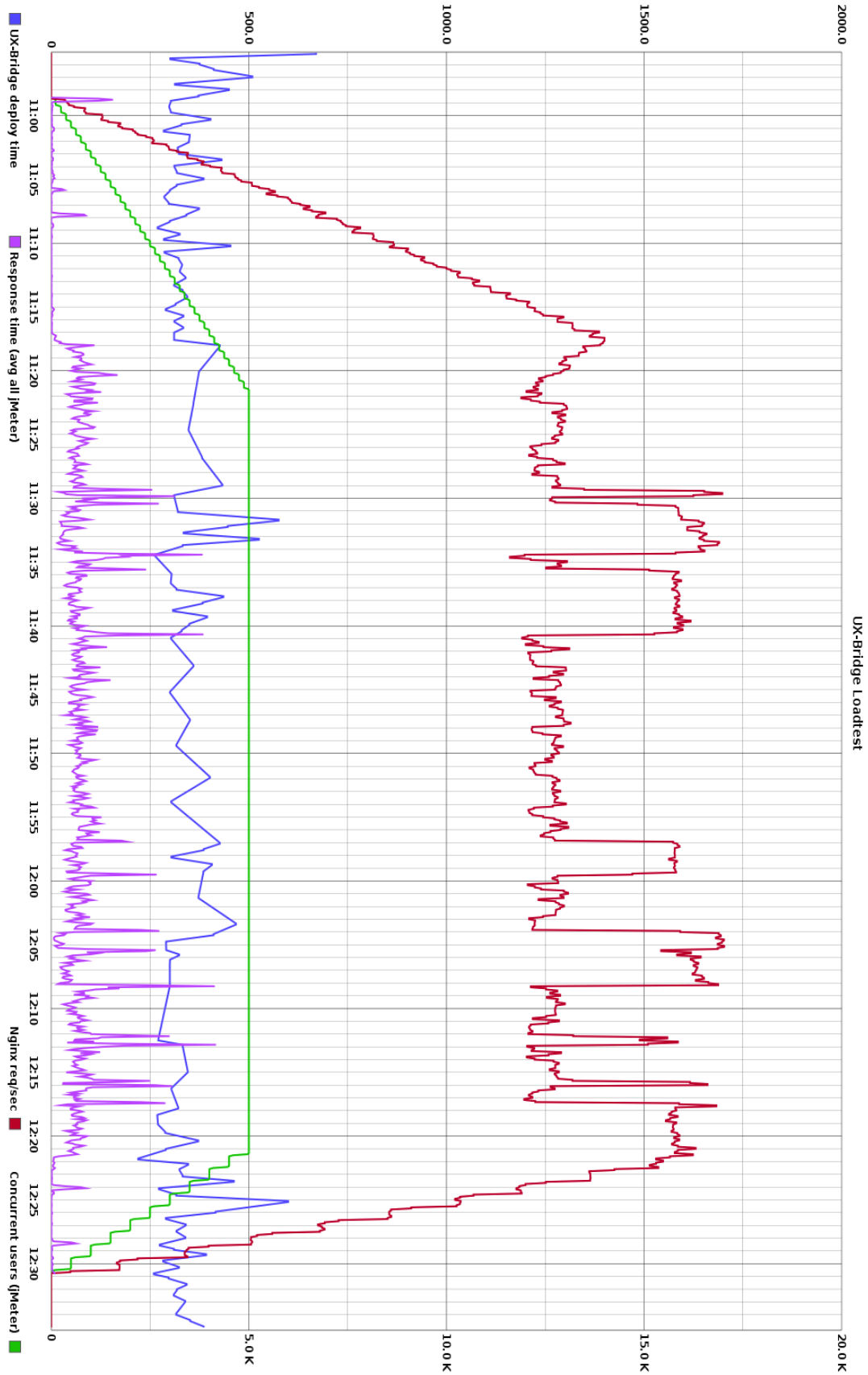
The number of simultaneous users on the website. The right y-axis applies.

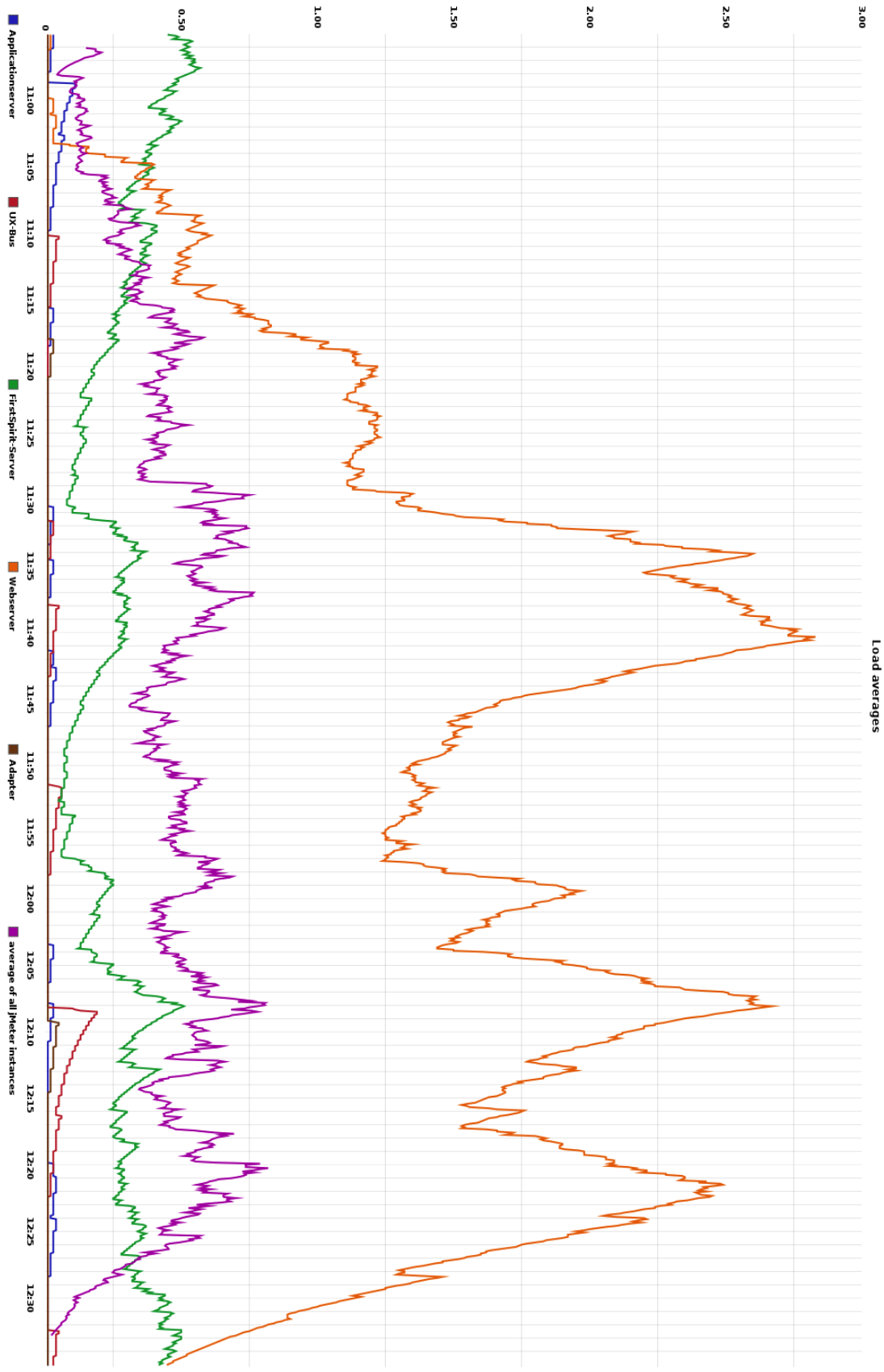
The graph precisely matches the expected curve. The ramp-up and ramp-down phases can be clearly identified.

4.3.5 Load on the participating machines

The load of the participating machines is depicted in the second graphic. None of the machines was brought to its load limits during the test run. The content modifications did not present any problems for FirstSpirit Server, the adapter or the UX-Bus. The application server was put under load only occasionally by using the web server as proxy, so that the average load also turns out to be very minor.

5 Appendix


UX-Bridge deploy time
Response time (avg all jMeter)
Nginx req/sec
Concurrent users (jMeter)
UX-Bridge Loadtest
0
500.0
1000.0
1500.0
2000.0
11:00
11:05
11:10
11:15
11:20
11:25
11:30
11:35
11:40
11:45
11:50
11:55
12:00
12:05
12:10
12:15
12:20
12:25
12:30
0
5.0 K
10.0 K
15.0 K
20.0 K


Applicationserver
UX-Bus
FirstSpirit-Server
Webserver
Adapter
average of all JMeter instances
0
0.50
1.00
1.50
2.00
2.50
3.00
11:00
11:05
11:10
11:15
11:20
11:25
11:30
11:35
11:40
11:45
11:50
11:55
12:00
12:05
12:10
12:15
12:20
12:25
12:30
Load averages