



The slide features a blue header with a photograph of a modern building on the left and the ISS logo on the right. The logo reads 'ISS Institute for Software Systems in Business, Environment and Administration'. The main title is centered in bold black text. Below the title, the authors' names are listed. At the bottom, the event name and location are specified. A blue footer contains the logo for 'Umwelt-Campus Birkenfeld' and 'FACHHOCHSCHULE TRIER'.

ISS Institute for Software Systems
in Business, Environment and Administration

Green Web Engineering – Measurements and Findings

Markus Dick, Eva Kern, Timo Johann,
Stefan Naumann, Christian Gülden


EnviroInfo 2012, Dessau

Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

The project “Green Software Engineering” (GREENSOFT) is sponsored by the German Federal Ministry of Education and Research under reference 17N1209. The contents of this document are the sole responsibility of the authors and can under no circumstances be regarded as reflecting the position of the German Federal Ministry of Education and Research.

This presentation corresponds to the following paper:

Dick, Markus; Kern, Eva; Johann, Timo; Naumann, Stefan; Gülden, Christian (2012): **Green Web Engineering - Measurements and Findings**. In: Arndt, Hans-Knud; Knetsch, Gerlinde; Pillmann, Werner (Hg.): EnviroInfo 2012: Man Environment Bauhaus: Light up the Ideas of Environmental Informatics. Proceedings of the 26th International Conference on Informatics on Informatics - Informatics for Environmental Protection, Sustainable Development and Risk Management, Part 2; Federal Environment Agency, Dessau, 2012. Volume 1. Aachen: Shaker Verlag, pp. 599–606.




ISS Institute for Software Systems
in Business, Environment and Administration

Motivation

- Power consumption of data centers in the world increased from 58 TW h in 2000 to 123 TW h in 2005
- Different efforts in the field of computer hardware
- Some activities to support the development of a “Green Web”

→ Do common techniques that reduce resource consumption of websites also reduce the energy consumption of the computer serving the websites?

 Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

2

The power consumption of data centres in the world increased from 58 TW h in 2000 to 123 TW h in 2005, and is still increasing. Hence, reducing the consumption of energy and natural resources caused by ICT is necessary.

Where manifold efforts exist in the field of computer hardware (that is: Green-IT), there is a lack of efforts in the field of computer software.

Therefore, methods are necessary that enable different stakeholders like developers, purchasers, administrators or even users to consider energy consumption induced by software in their decisions on software products.




Outline

- I. Green Web Engineering
- II. Test Rig and Measurement Method
- III. Measuring Joomla!
- IV. Summary & Outlook




I. Green Web Engineering



ISS Institute for Software Systems
in Business, Environment and Administration

Definition of “Green Web Engineering”

“Green Web Engineering describes the art of developing, designing, maintaining, administrating, and using a website in such a manner that direct and indirect energy consumption within the complete life cycle of a website is reduced.”


 Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

5

Direct energy consumption or the so-called “first order effects” comprises energy consumption that is caused by the infrastructure, which is necessary to provide the service or to present the website to the end-user. This includes data centre and networking infrastructure as well as the infrastructure on the client-side.


Indirect energy consumption is also called “second” and/or “third order effects” or “enabling effects”. This means that the service offered by a website may be an enabler of change in production processes or consumption with the potential to mitigate environmental impacts. For example a service can help to better organize traffic or to replace traffic, which has a positive effect on the environment.

However, a service may also cause negative effect on the environment, for example if it generates additional demand for traffic.



Some Principles and Suggestions of GWE

- **Configure HTTPs Caching Support for Static Content**
 - setting far future expiration dates and cache control headers for resources that infrequently change
 - setting Last-Modified headers and Entity Tags for resources that do not need recalculation
- **Configure HTTP Compression Support for Static Content**
 - server compresses its response with GZIP / DEFLATE
 - reduces pay load and transmission time of responses
 - compressing content consumes additional processing time
→ higher power consumption



For Administrators we suggest to

- Configure HTTPs caching support properly.
 - That means: setting far future expiration dates and cache control headers for resources that infrequently change.
That's the so-called "expiration mechanism".
For example, Google Germany uses for its logo expiration dates of one year ahead in the future.
 - Further we suggest to set Last-Modified headers and Entity Tags for all resources that do not need recalculation on subsequent requests.
That's the so-called "validation mechanism".
This applies mainly to static content.

Both mechanisms enable properly configured browsers to cache content effectively and to reduce the amount of transferred data on subsequent requests, which reduces energy consumption

- Furthermore we suggest to use HTTP compression and
- To apply classic "green IT" concepts like server virtualisation or webhosting with renewable energies.

ISS Institute for Software Systems
in Business, Environment and Administration

Example: HTTP Compression

Example files	File size (KB)	GZIP size (KB)	Savings
index.html	5.45	2.44	55.2%
style.css	2.73	0.68	75.1%
prototype.js	126.00	29.51	76.6%
ida-logo.png	24.80	24.86	-0.2%
ucb-logo.png	9.27	9.28	-0.1%
Total	168.25	66.70	60.4%

Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

For Example HTTP Compression...

By enabling HTTP Compression we achieved for this simple example website, which comprises five files, data transfer savings from 50 to 76 per cent depending on the file type.

It is clear that files which contain many repetitions like the JavaScript file in this example, are better compressible than those with less repetitions like the HTML file.

It is also clear that compressing already compressed files like the two image files is even counterproductive so that these files should not be compressed again by the web server.

Compressing the data stream reduces the transfer volume and thus reduces power consumption.

But compressing the data stream on the fly depending on the capabilities of the user's web browser may lead to a higher processor usage and thus possibly increases power consumption.

Hence, we suggest to pre-compress static content wherever possible and to configure the web server in such a manner that it serves the pre-compressed or uncompressed versions depending on the capabilities of the browser.



Some Principles and Suggestions of GWE

- **Implement Application Level Caches**
 - store generated output retrieved from subsystems in a cache
 - saves CPU time
 - lower energy consumption / higher energy efficiency
- **Implement HTTP Compression for Dynamic Content**
 - compression supported by the browser
 - choose most energy efficient compression



Some Principles and Suggestions of GWE

- Optimize Graphical Elements, Logos and Photographs
 - **graphical elements** or logos: PNG format instead of JPEG (better compression algorithm)
 - indexed palette instead of RGB full colors
 - realising text by CSS and HTML

 - **photographs** stored in JPEG format
 - blurring areas of an image that do not carry important information




Example: Optimizing Graphical Elements





Color Palette	Size(KB)	Savings
with text		
RGB	13.3	0%
without text		
RGB	5.47	58.9%
256	2.70	79.7%
16	1.15	91.3%
10	0.99	92.6%

PNG graphic in RGB colors with text vs. graphic with 10 color indexed palette, text is put to the foreground with means of HTML and CSS




ISS Institute for Software Systems
in Business, Environment and Administration

Example: Optimizing Photographs

JPEG Quality	Size(KB)		Savings
	normal	blurred	
with text			
85%	51.5	33.4	35,1%
65%	32.5	21.2	34,8%
without text			
85%	48.4	29.7	38,6%
65%	30.5	18.9	39,0%

JPEG image with 85% quality vs. optimized image (85% quality, blurred background, text is put to the foreground with means of HTML and CSS)



Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

11

photo 1: 85%, with text + BG: 51.5 KB
 photo 2: 65%, without Text + blurred BG: 18.9 KB
 => Savings: 63,30%

The second example shows how an advertisement image or a photograph in JPEG format can be optimized.

The relevant content of this image is the person and the blue banner in the foreground. The blue banner may be overlaid with an advertising slogan. The background transports no necessary information for an observer.

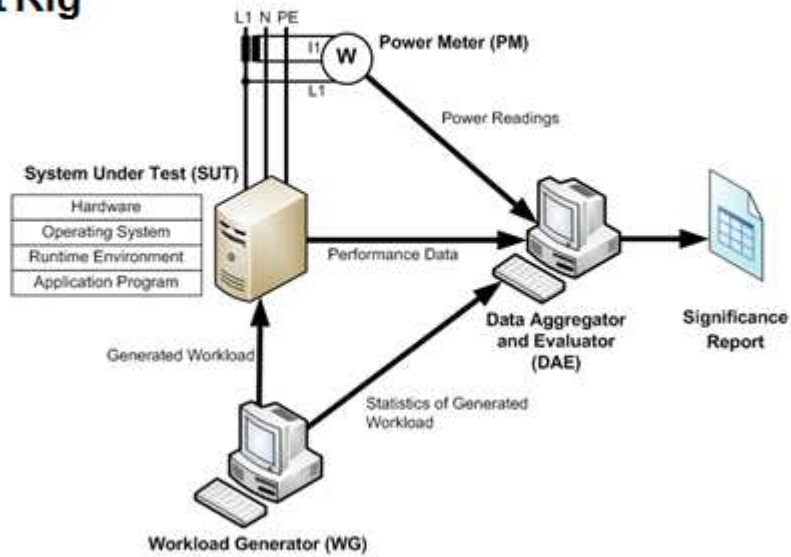
So, blurring the background supports JPEGs compression algorithm which results in a lower file size.


For this example which was stored at 80% JPEG quality we achieved file size savings of 47% compared to the normal files size simply by blurring the background.



II. Test Rig and Measurement Method

Test Rig






ISS Institute for Software Systems
in Business, Environment and Administration

Evaluation Process

- **Aggregation**
 - Power/energy readings, CPU performance, workload log
- **Validation**
 - Checks relative chain frequencies
 - Checks task preparation times (mean, std. dev.)
- **Evaluation**
 - t-Test, 2 test series for each SUT
→ spitted into three time series of 10 minutes
 - H_0 : Mean energy consumption of SUT 1 & SUT 2 is equal
 - H_1 : Mean energy consumption of SUT 1 & SUT 2 not equal



Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

14

Validation:

Answers the question if generated workloads comply with parameters predefined in the workload definition

For each of the three values, the maximum acceptable tolerance is given.

The validations are done for each user type



III. Measuring Joomla!



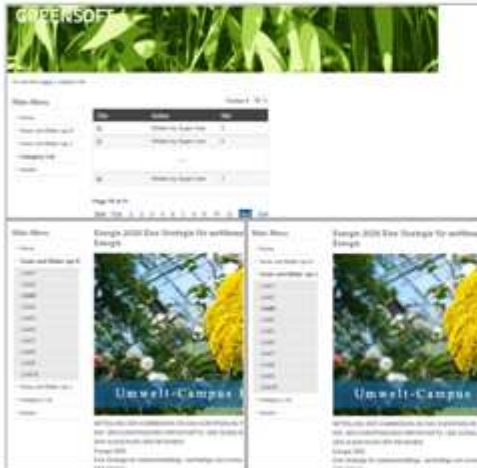


Measurement: Software on Servers


- Web CMS Joomla! 1.7
- Workload Generator
 - Apache JMeter load and performance test tool
- Environment
 - 2 x Intel Xeon dual core CPU @ 2.4 GHz, 4 GB RAM
 - Ubuntu GNU/Linux 10.04 SMP (Kernel 2.6.32)
 - Apache Web Server 2.2.14, PHP 5.3.2, MySQL 5.1



Setup of the Testing Website



- **Content:**
 - text passages from EU legal documents
 - self-taken photographs
 - graphics from R&D project
- **Navigation area:**
 - 10 articles without optimized images
 - 10 articles with optimized images
 - list of 100 generic articles to simulate paging




ISS Institute for Software Systems
in Business, Environment and Administration

Result overview

1. Joomla without any improvements
2. Joomla with application level cache
3. Joomla with application level cache, and optimized images
4. Joomla with application level cache, optimized images and compression


No.	Load level	Energy (AVG)	Load level	Energy (AVG)	Load level	Energy (AVG)
1.	50%	39.250 Wh	30%	35.286 Wh	10%	29.015 Wh
2.	50%	37.616 Wh	30%	33.818 Wh	10%	28.396 Wh
3.	50%	37.727 Wh	30%	33.905 Wh	10%	28.495 Wh
4.	50%	37.573 Wh	30%	33.777 Wh	10%	28.376 Wh

 Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

18

Projection to one year of 24/7 operation:
savings 153,9 kWh/a = 30,78€/a (0.20€/kWh)


Hochrechnung 1 Million Nutzer



Measurement results

- Comparing the reference system and scenario 4:
 - common techniques reducing resource consumption of websites do also reduce the energy consumption
 - approx. savings: 4.23 % (see table below)
 - may be further increased by implementing additional suggestions

No.	Scenario	Load level	Energy (AVG)
1.	Joomla without any improvements (reference system)	50%	39.250 Wh
4.	Joomla with application level cache, optimized images and compression	50%	37.573 Wh



WCMS and webapps should provide application level caches so that there is no need to query the database for each request (data that is often requested should be hold in the cache, whereas there is no need to cache data that is seldom requested)

WCMS and webapps should support compression for static text based or already uncompressed content. Static content should be compressed in advance so that the stream is not compressed for each request (which consumes CPU time and energy)

WCMS and webapps should support compression of dynamically generated responses

Even if not checked in this paper, WCMS and webapps should support browser caches to prevent repetitive downloads of an unaltered resource in subsequent requests

WCMS and webapp development tools should provide tips and hints for authors, administrators, and developers on how to improve resource consumption of images, accessibility of images for impaired people, or on update strategies



Findings

- Enable application level caches in order to reduce database queries
- Enable compression for static text or already uncompressed content, if possible in advance
- Enable compression for dynamically generated responses
- Support browser caches to prevent repetitive downloads of an unaltered resource in subsequent requests
- Enable tips and hints in IDE



IV. Summary & Outlook





Summary

- We presented
 - Suggestions to reduce the net load of websites: “Green Web Engineering”
 - Measurement and evaluation of Joomla! Configuration
 - different workload levels
 - different specifications of green web principles
 - Statistically significant difference in mean energy consumption in different website configurations
- **Saving great amounts of resources and energy might be possible by implementing of Green Web Engineering.**



Outlook

- Measure different WCMS, versions, configurations of web technologies
- Check web techniques out for their potential to support a “Green Web”
- Map more suggestions to a specific measurement setup

- What do these possibilities mean for...
 - ... cloud strategies?
 - ... web providers?



ISS Institute for Software Systems
In Business, Environment and Administration

Thank you for your attention!

GreenSoft
Green Software Engineering

Feel free to contact us:

Stefan Naumann
s.naumann@umwelt-campus.de
Trier University of Applied Sciences
Environmental Campus Birkenfeld
Institute for Software Systems
Germany

SPONSORED BY THE



Federal Ministry
of Education
and Research

greensoft@umwelt-campus.de
<http://www.green-software-engineering.de/>

Ref.-No. 17N1209

 Umwelt-Campus Birkenfeld
FACHHOCHSCHULE TRIER

24

If you have further ideas or suggestions on how to improve the measurement and rating method,

feel free to contact us at the Environmental Campus Birkenfeld of the Trier University of Applied Sciences in Germany.

Thank you very much for your time and attention.