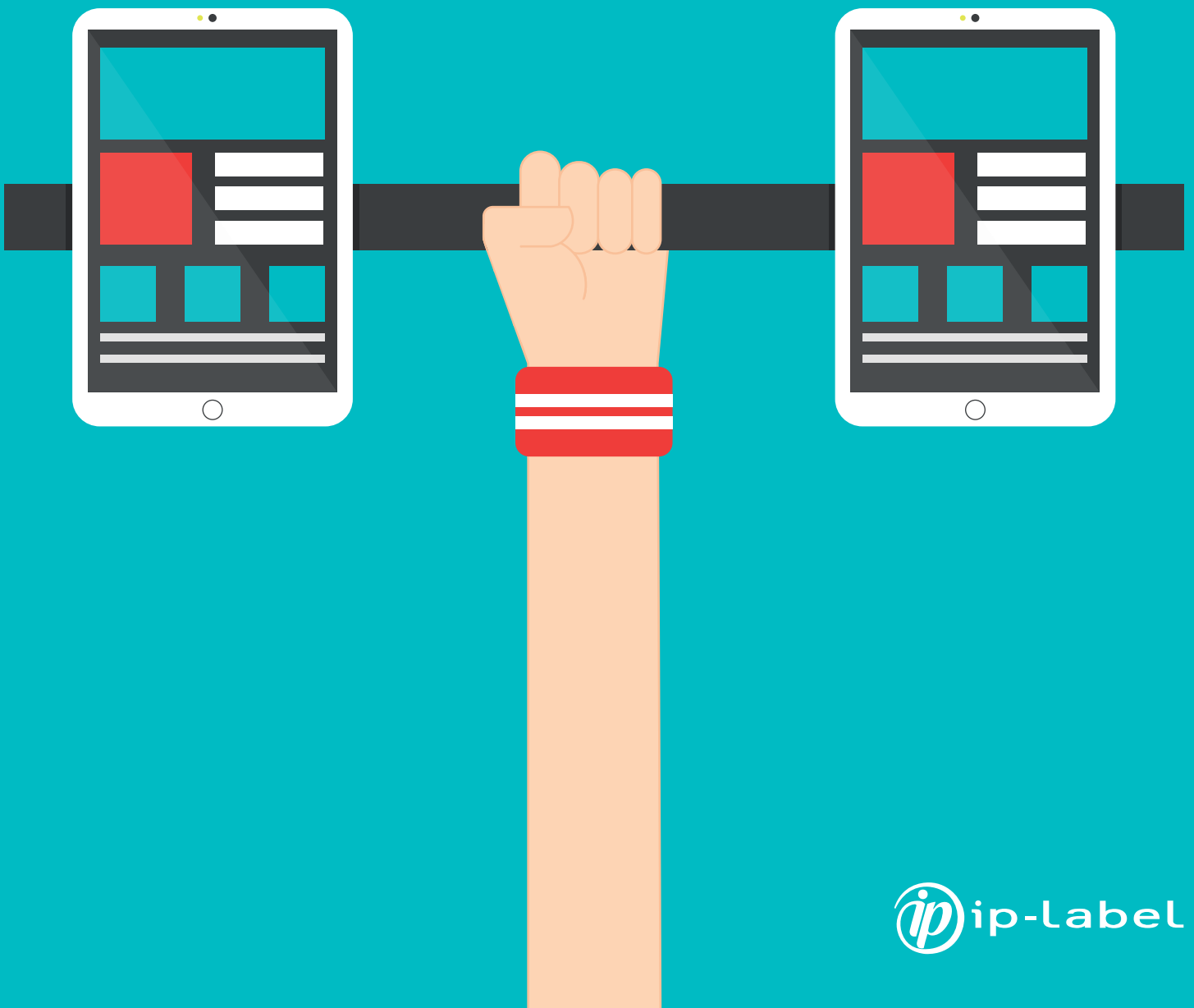


How to make your web site a top performer

Architecture & hosting, design & optimization,
indicators & technologies, and more...



What matters most is to take part in the game...



BUT WE ONLY
REMEMBER THE WINNERS

This famous saying, attributed to Baron Pierre de Coubertin, founder of the modern-day Olympic games, doesn't hold true anymore. Nowadays elite-level athletes are out there to win. Nothing is left to chance. Every detail of their training, every ounce of effort they spend counts.

The same goes for other areas, most especially **the world of the internet where competition is fierce**. If a web site is not fast enough, visitors will look elsewhere for service they feel is better. **How can a web site be made to perform better?** What can you do to make it operate smoothly and efficiently to fulfill users' expectations?

In this white paper we share a few good practices for bringing web sites to top performance level. Because what matters isn't simply to play the game, but **to win** it.



Contents

1 What makes a site good in terms of **performance**

2 How to **prepare** for your web site (architecture & hosting)

3 How to **optimize** your web site (design)

4 **Tools** for measuring and comparing performance

5 Conclusion

1

What makes a
site good in terms
of **performance**

A few statistics

Statistics accumulated by ip-label over the past 15 years show that a site is considered to be performing well if:

- it begins displaying the page within **300 to 750 ms** after the user clicks 'GO'
- its pages load in **under 3 seconds**
- it consists of **less than 100 objects** per page
- it calls on **not more than 25 different domains**
- page size is **under 1300K**



Audiences are directly affected by web performance. Online shoppers are very sensitive to delays. **The conversion rate, the number of visits, and the customer's loyalty all depend on how fast the web site is.**

Fluidity and speed as the user perceives them result not only from **how your site is designed**, but also depend on **the device and software in the user's context**. A web site's performance also has a significant impact on the brand. This influence is all the greater as a study finds that dissatisfied users tend to perceive a web site's performance as being **15% slower** than it really is. Even worse: when they talk about it, the shortcomings are amplified by as much as **35%**!



2

How to **prepare** for your web site (architecture & hosting)

Above and beyond considerations at the application level, a high-performance web site requires careful attention to some basic rules about certain aspects of the infrastructure.

The architecture

First of all, it is essential to:

- **Determine the site's audience level and foreseeable traffic**, which should take marketing campaign cycles into account and align the planned infrastructure with traffic.
- **Zone strategic and international markets, and look closely at the competition.** It is important to note that user contexts may vary. For example in China 47% of IE browsers are in version 8. In Europe 89% of IE browsers are in version 11. Browser loading performance differs by version.
- **Set priority performance goals and determine their compatibility with deadlines.**

Hosting

Hosting is a crucial part of a web site's success because it weighs in considerably on **page loading time**. Selecting a provider is not an easy task.

There are many possibilities, such as **hosting on a shared server (low cost)** or a **dedicated server**, a **VPS (Virtual Private Server)**, or even a **cloud solution**.

With such a diversity of choices, it is highly advisable to identify your real needs and what your priorities are **in terms of performance**.



Things to watch out for

The choice of hosting provider also depends on your **budget**, the **technical skills available within your company**, and which **performance issues** you want to focus on.

The hosting company must be able to **secure your platform** (to keep you safe from attacks and ensure backups and more), and especially **deliver on its promises** to maintain a certain level of **availability of your service** (Service Level Agreement).

These undertakings are vital to your customers' satisfaction, and allow you to ensure your revenues.

The web server

For a little while now HTTP2 has brought beneficial changes particularly to **mobile user contexts** where, because of greater delays, optimization of TCP/IP connections is crucial. HTTP2 offers interesting functionalities such as **Server Push**, which lets the server decide on critical resources in order to optimize the user's experience. Setting up HTTP2, however, means updating the web server with dependencies to the application libraries; this might require a whole other dedicated project.

CDN

For solutions that resort to deployment via a **CDN (Content Delivery Network)**, the things to watch out for are essentially the same as those described above. Operators must be able to **protect data** against attacks, theft, or loss, and ensure optimum availability of the CDN. In addition, the CDN provider can facilitate **HTTP2 implementation** for some static resources.

3

How to **optimize** your web site (design)

Performance should be on developers' minds at all times and an ever-present goal from the outset of the web site project.

Good optimization means applying a few good practices, whether they are of a technical nature or simply common sense. Obviously everyone who works on the technical side of web sites knows that the kind of device, usages, network quality and type, and the version and type of browser greatly influence the level of performance.

Nevertheless, in the current state of browser technology there are several 'universal' good practices and strategies **which can improve performance regardless of the context and environment of use.**

Streamline content

When people talk about reducing the 'size' of a page or content, they mean **streamlining the items – images or text**, for instance – which have to be displayed in the browser so that the visitor can see them.

Developers also think about **compacting the code and the resources** that underlie the page in order to make it faster to load. Images are usually among the first objects to target because it is fairly simple to reduce their size.

One approach consists of **converting the image to a lighter format**, like WebP. It's also important to select an **image size that fits the screen**. To do that, developers can intervene in the **'srcset' attribute in HTML5**, for example.

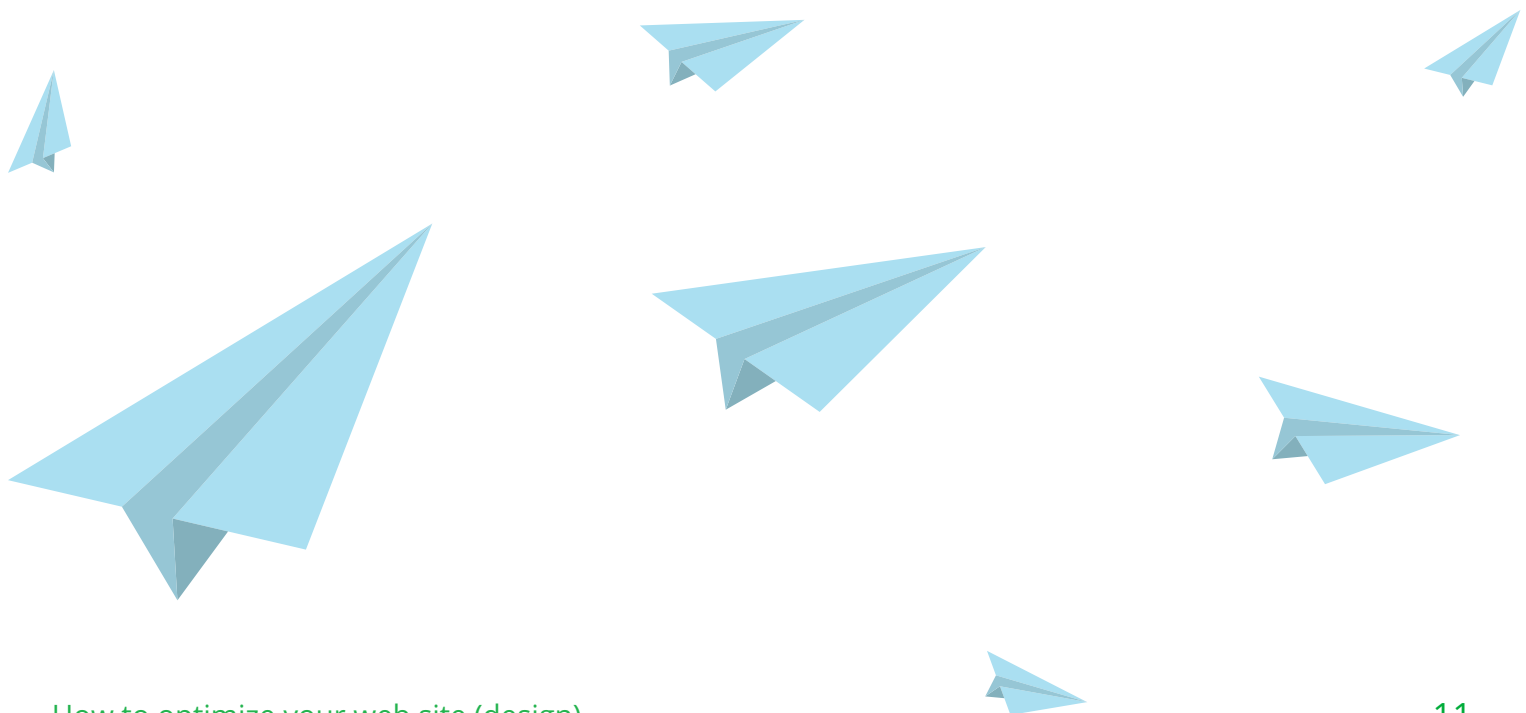
As for text, **compression** is the way to go. Very significant savings, on the order of 50% to 80%, can result from using compression software like gzip to **compress the HTML, JavaScript, and CSS** up to the client. These three types of content must not be overlooked. Let's not forget either that **compressing fonts, XML, and web services** can also considerably improve the loading speed of web pages.

Lastly, '**minification**' is the process of **trimming down elements that are invisible to end users**. The primary targets for this are **HTML, JS, CSS, RSS, Robot.txt, XML, and JSON**. Here, removing superfluous spaces and all comments can yield an economy of 10% to 20%, without affecting the visual aspect of the page. Deleting all useless text makes pages lighter; they therefore load faster.

Cut back on requests

In order for a web page to appear in full on the user's device, the client side must send the appropriate server(s) requests for the information to display. These round-trips of requests and information take time and slow the page down. This is why it is important **to optimize performance by reducing the number of these round-trips**.

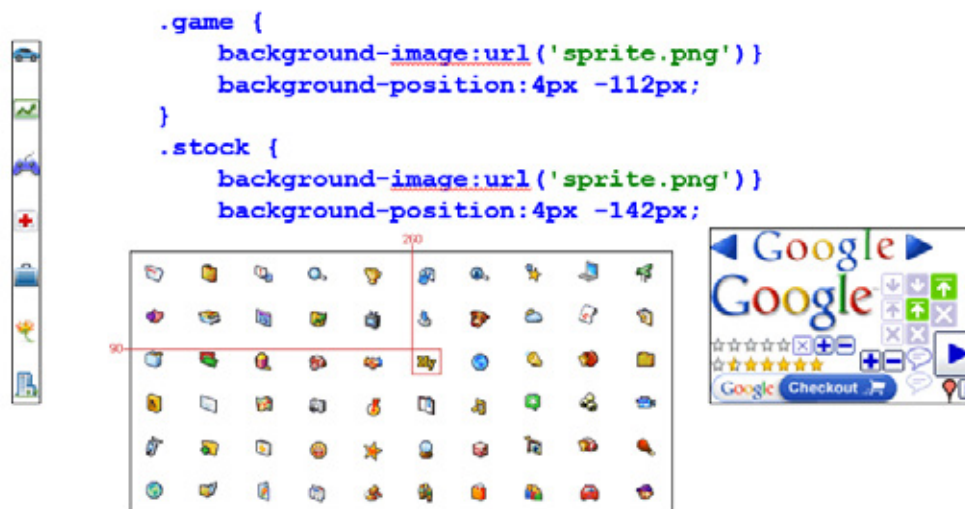
Developers can do this by using a **package manager** (like npm webpack, to name only one) which groups together files or modules in a single block. The block then loads asynchronously to **minimize loading time**. This way, instead of sending requests and loading piece by piece, this type of tool provides for more **economical management of the elements that make up a page**.



Using a similar kind of approach, **JS and CSS files can likewise be combined**. This **prevents data redundancy** and again limits the number of back and forth trips of requests/information.

```
<!-- ... -->
<script src="/script/common.js"></script>
<script src="/script/scriptaculous/prototype.js"></script>
<script src="/script/scriptaculous/scriptaculous.js"></script>
<script src="/script/scriptaculous/effects.js"></script>
<script src="/script/scriptaculous/dragdrop.js"></script>
<script src="/script/classes/Collection.js"></script>
<script src="/script/classes/Evt.js"></script>
<script src="/includes/omniture/s code.js"></script>
<script src="/script/common.menus.js"></script>
<!-- ... -->
```

A similar rationale can be applied to images. CSS sprites can **group them together to transmit them in a single block**, from which they can be retrieved by their coordinates.



Lastly, **significant optimization benefits** can be obtained by **using the browser cache wisely**. For example, attention should be paid to adjusting the **HTTP 'Expires' header**. If no date is specified, useless round trips are generated.

The developer may enter a date far in the future to store static images and other things in the cache, or specify a date that corresponds to a scheduled update to avoid useless requests in the meantime. When the browser knows its cache still contains a given object, it retrieves it there instead of issuing a request to a server.

Let the browser help

Another important strategy is to allow the browser to enhance site performance. To facilitate this, the web page's content must be organized so that the browser can display it as fast as possible.

Web developers know all about the chronology of browser operation and the order in which browsers 'read' and process the page's various components. They can therefore **adjust the HTML code and the moments when different objects come into play** so that these are taken in stride within the browser's normal operation.

By placing HTML tags judiciously, they can **make fonts and CSS load first**, before rendering starts (see '**start render**' in the following section).

Likewise, since **JavaScript blocks rendering**, the developer can intervene so that the **important JS are loaded upstream** and the execution of other JS is deferred. Lastly the developer can keep objects from loading while others are already being downloaded from the same host.

The three strategies described briefly above are part of the good practices of web design and development. Certain practices have been consolidated and are now known conjointly as '**Progressive Web Apps**', which aim to emulate the behavior of mobile applications, but use advanced web techniques like service workers, for instance, to handle offline mode.

Good practices are evolving constantly in pace with technology and other changes. What is true now, even without becoming false, may no longer be as effective tomorrow in whole new contexts! Web site optimization is a permanent, ongoing endeavor, a challenge that knows no downtime.



4

Tools for measuring and comparing performance

There are a number of sources of information for assessing the performance of web sites.

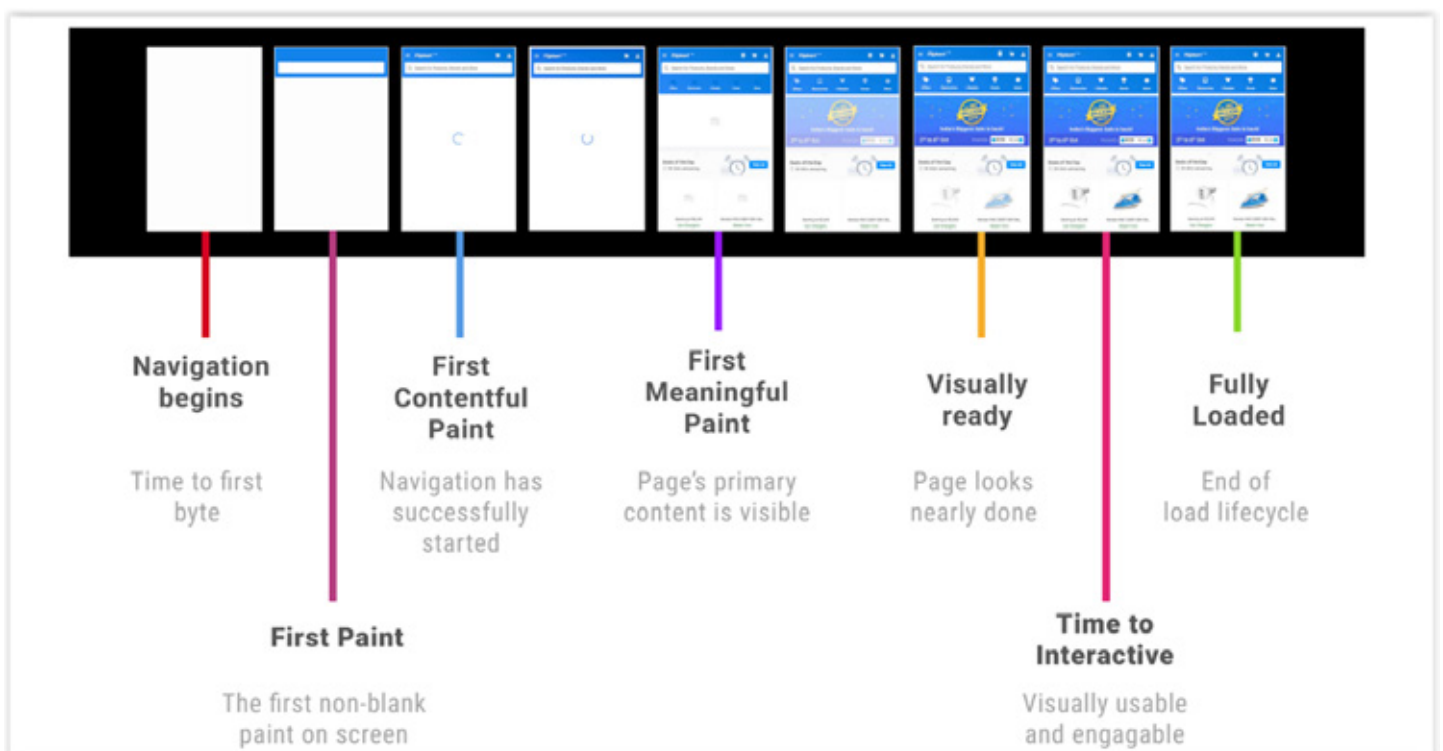
Performance indicators are supplied either by the browser itself or by online services. These metrics help situate a site's level of performance. Analyses then must be conducted to understand what these technical measurements mean in terms of the quality of performance as it is experienced by users.

After we look at indicators, the second part of this section offers an overview of the **tools and technologies** which provide **continuous measurement of web site performance** and statistics that are useful for managing and improving performance.

Performance indicators

Detecting and understanding weaknesses to optimize performance is a technical matter. Various metrics are available to developers and operators so that they can analyze application performance. Below we describe just a few that enable them to assess and ensure web site performance.

At the conceptual level, page loading can be schematized as follows:



Source: Google

A variety of metrics, some of which are standardized, is supplied by the browsers themselves (**W3C Navigation Timing API**). Such indicators let you measure and compare the performance of your site against the performance of competitors' sites.

Speed Index

This 'visual' indicator lets you **measure display progression or the average time of the visible parts of a page**. It calculates an overall score in milliseconds, adjusted for the size of the screen. This metric expresses right up close how users experience the site. It is very useful for comparing a site to that of a competitor, or for assessing the effects of page optimizations during an A/B test. <https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index>

Start Render

This is the moment when the **blank page starts displaying the first elements of the web page**. The quicker the items appear, the more the visitor feels that the web site is fast. Visitors are less likely to leave the page after the rendering starts. This metric is available on IE only.

Onload

This is the historical metric for page loading which arises from JavaScript events. It is nevertheless of limited use for pages that load asynchronously (Ajax) after the onload event; they therefore cannot be counted in the metric!

Time to Interact (TTI)

This is when the user can begin interacting with the page's content. This is the metric favored by ip-label for RUM measurements (see below).

The best metric depends on the characteristics of your site. To get an idea of how these metrics correlate with real-life perceptual experience, the following platform provides very interesting calibrated tests (to conduct on Chrome or Firefox): <https://eyeorg.net/>

Tools and technologies

Developers and operators, primarily – but also business lines for which web performance is an important issue – can take several technical approaches to **ensure automated measurement of a site's performance**.

Here is a brief overview of some of the technologies at the service of web site performance management:

Real-User monitoring (RUM)

RUM **analyzes the performance of a web application or web site on the basis of real traffic** (quality of experience). Measurements are made from the user-side device via a component embedded in the application. This tracer **collects the response times of each critical step** and compares them by the user's location or hardware and software configuration. RUM can also correlate performance with the number of visitors in order to identify any degradations during heavy-traffic periods. **User-perceived response times thus can be measured in any type of context:** all navigation universes (URLs), all types of device (PC, smartphone, tablet) and software (OS, version and type of browser, etc.), internet access providers, and locations (city, subsidiary, store or office for intranet, country).

Synthetic (active) monitoring

Active, shared robots are connected to simulate any operation representative of ordinary usage by a typical user of your digital services.

As a true end user with added value, a robot can provide **performance indicators** of applications as they are experienced and **alert you to any errors, incidents or unavailability** that may arise, on the basis of integrity and data consistency checks, **24/7**.

Troubleshooting

Installing a software introspection agent (APM) on your servers enables you to quickly and straightforwardly identify all possibilities for optimizing your critical applications, whether the task is to locate the cause of an incident in production or set up an approach for testing and continuous improvement in the qualification stage. Troubleshooting provides an overall view of the **response times of each of the application's components as well as error codes, and digs down into the code of the applications** that are generating poor response times.

Performance panel results

Comparative studies are made of certain business sectors by measuring the user-perceived quality of web site performance. With this information you can compare your site's performance against the performances listed in a baseline representative of your business sector. ip-label has been publishing comparisons like these for some fifteen years conjointly with international press partners:

BFM TV/01net in France: <http://bfmbusiness.bfmtv.com/services/indicateurs-ip-label/performances-des-sites-web/> (in French)

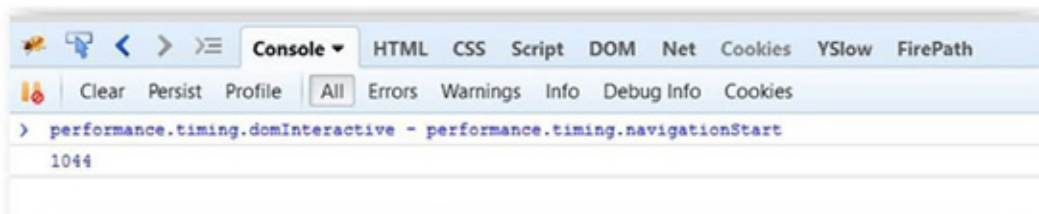
Media Centre: <http://www.ip-label.co.uk/category/benchmarks-press/>

Tools of the trade for monitoring, metrics, and analysis

Datametrie monitoring software from ip-label provides a set of tools which let you visualize the loading times of each item of a page. It pinpoints and highlights slow content (in the slowest 10% of the customer database):

| Start | | Time (s) | Object size (bytes) | Code | FT | |
|--------|--|----------|---------------------|------|----|--|
| +0,003 | | 1,019 | 26 825 | 200 | | http://www.aftenposten.no/ |
| +0,968 | | 0,211 | 84 952 | 200 | | http://www.aftenposten.no/seksjon/dist/main.min@1478771884.css |
| +0,969 | | 0,064 | 3 501 | 200 | | http://www.aftenposten.no/seksjon/dist/sabot.min.js |
| +0,969 | | 0,304 | 269 739 | 200 | | http://drfront-prod.schibsted.tech/ap/data/css/front-3.css |
| +1,302 | | 0,184 | 39 990 | 200 | | http://ap.mnocdn.no/drfront/images/e4c50fe7fc40edb6411beb96a9c49078.jpg |
| +1,304 | | 0,149 | 20 263 | 200 | | http://ap.mnocdn.no/drfront/images/b4fdd5b7cddb9611920471d89a2f2cd.jpg |
| +1,305 | | 0,162 | 16 659 | 200 | | http://ap.mnocdn.no/drfront/images/41b717ac375f85d95a9e1257553cdc58.jpg |
| +1,305 | | 0,160 | 15 202 | 200 | | http://ap.mnocdn.no/drfront/images/c1a6bf1ece890d13c114600236420924.jpg |
| +1,305 | | 0,178 | 35 597 | 200 | | http://ap.mnocdn.no/drfront/images/aaf843d1bb2990bd99f2ab7f60d20843.jpg |
| +1,305 | | 0,214 | 61 550 | 200 | | http://ap.mnocdn.no/drfront/images/275e96b710fa8cc912fb9e7873a4290b.jpg |
| +1,332 | | 0,238 | 89 448 | 200 | | http://www.aftenposten.no/seksjon/dist/main.min@1478771891.js |
| +1,333 | | 0,351 | 293 | 200 | | https://pixel.glimr.io/v2/ping/FF532F4DB5094A848928/ |
| +1,346 | | 0,078 | 4 996 | 200 | | http://www.aftenposten.no/seksjon/dist/core-header/ap/fonts/i/cons.woff?q525ru |

To perform a manual test using Firefox, the **Firebug** add-on is very popular with developers. This tool gives them a detailed view of page loading, component by component. For instance, they can access RUM metrics (W3C Navigation Timing API) right in the console. The metric shown in the example is TTI:



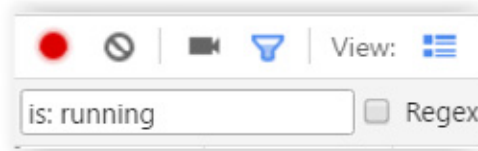
In Chrome, the tool is called **Developer Toolbar** and is delivered in the standard browser. Like Firebug, it offers access to performance data.

Among its numerous functionalities, the display featuring CPU and render (FPS = frames per second) shows any troublesome skips in the rendering of a web page:



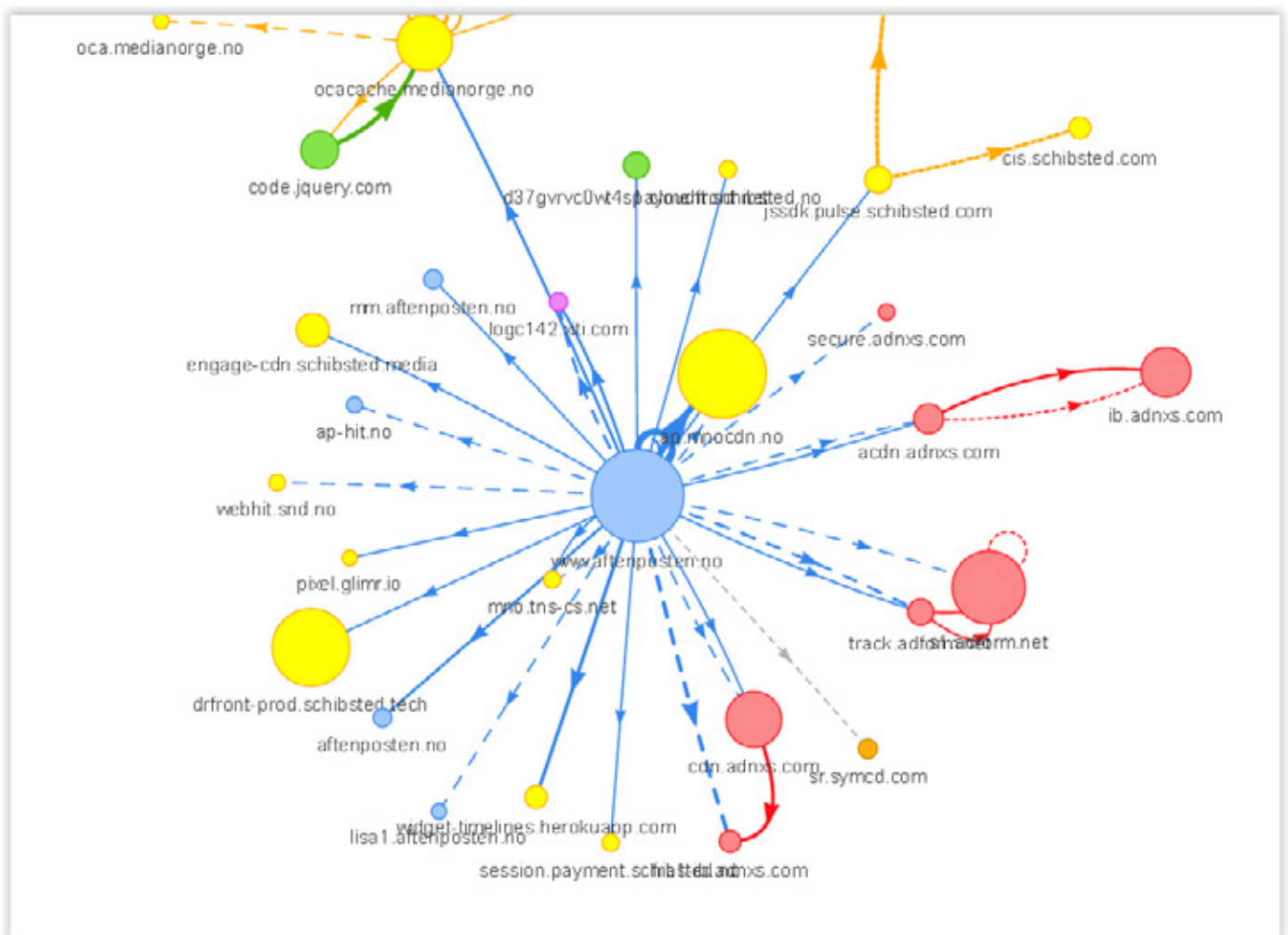


The filter option 'is: running' lets you see the objects that are being loaded at any given time:



To better monitor the third-party content of your pages, ip-label also provides continuous analysis tools (monitoring). In addition, analysis of third-party contents and the relationships between them can be visualized on the site

<http://requestmap.webperf.tools/>:



Source: Request map

5

In conclusion

A web site is now more than ever the direct link that brings the web service provider and its users together. Very often, these users are also customers. Income from the site therefore depends directly on how well your visitors are satisfied.

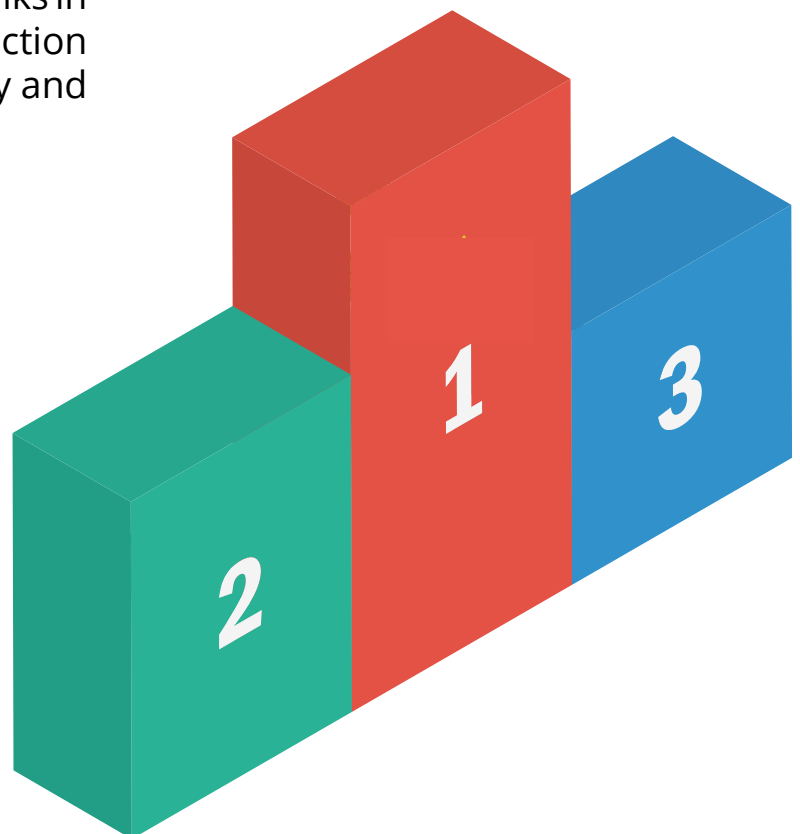
A site that performs below expectation can cause visitors frustration. They will not hesitate to switch to an equivalent service from your competitors, and could become loyal to them instead.

Poor performance can also have indirect technical and financial repercussions. Failures due to sluggishness among the various components of the web site may prevent an order from being placed.

User experience now deserves to be a major consideration for any enterprise that wants to keep its audiences and attract new ones. This is an all-out, continuous struggle to improve performance, gain a few hundredths of a second, analyze and understand the interactions between the various links in the chain, all to ensure the satisfaction of a public that is ever more finicky and demanding: your customers.

A web site is a dynamic, living entity which must evolve continuously. It is indispensable to **define a performance strategy at the outset to maintain the site in peak performance condition at all times.**

Best practices, such as those presented in this white paper, keep on developing. We therefore recommend that you check regularly our blog [Performance Wire](#) to stay informed of the latest good practices in digital performance.



About the ip-label group

ip-label has been assisting enterprises in their digital transformation for over fifteen years. A leading specialist of user quality of experience, the ip-label group offers an extensive range of APM (application performance management) solutions for analyzing and measuring the performance of all digital services including web, business apps, mobile, telephony, video and voice.

**Want to know more about application
performance management?**

Request a demo



www.ip-label.com
marketing@ip-label.com



Paris | Madrid | Stockholm | Helsinki | Shanghai | Tunis

©ip-label 2017. All rights reserved