CASE STUDY

DOTINUM

PROJECT

MEDtube is an innovative knowledge-sharing site for physicians that uses various multimedia techniques.

MEDtube has become the largest online library of professional medical films of high quality and practical educational value. The site enables its users to upload their video materials, photos and presentations.

What one can find there includes: videos and live surgery broadcast, photos of medical cases and conditions, medical animations, interviews, presentations and documents in all medical specializations.

The materials are supplied by physicians, medical associations and medical schools, and are also produced internally by **MEDtube**.

CHALLENGE

In the beginning, the budget of the site was so limited that the system could use only one shared server on which there was a database, a file repository (images, video and documents), as well as the site's files.

Due to the increasing popularity of the site, a shared web hosting slowly proved to be insufficient and it was necessary to transfer the site to separate virtual machines.

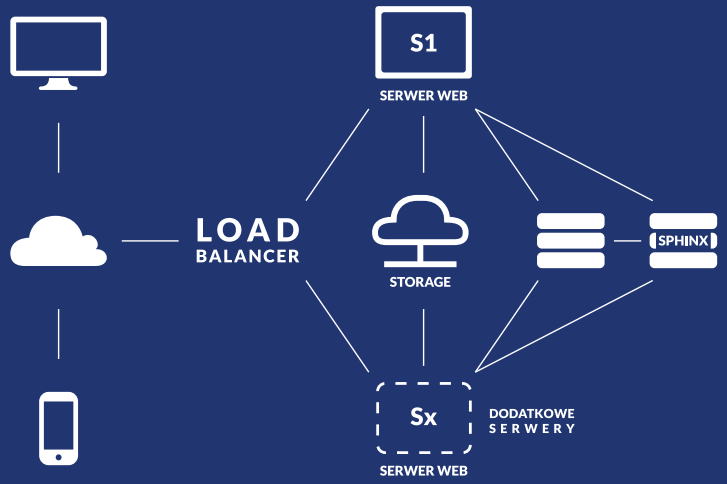
It turned out to be challenging to transfer all the videos within a reasonable time, as their total size at that time was about 1 TB. The site could not afford to go offline for so long. It was necessary to implement a data transfer procedure that ruled out any downtime.

DOTINUM

IMPLEMENTATION

First, we prepared a new system architecture, breaking it down into the following parts:

- replicable virtual machines with WEB files of the site
- storage of all the image files, videos and documents
- a database server
- SphinxSearch server



Outline of the current MEDtube architecture

Replicating virtual machines with WEB files of the site and using a load-balancer makes it possible to increase the number of machines required to serve an individual user depending on demand.

Additionally, implementing a fast search mechanism based on SphinxSearch considerably relieved the database server and hugely increased the speed of the site.

Another migration stage was moving the database to a new database server. It was the only time when one could notice an interruption in the system's operation. We could have used other solutions (e.g. replication), but the database was so small that its transfer took about 30 seconds, so implementation of some other complicated mechanisms was not necessary. Also, we set up a WEB system for the new database server.

The next step was moving the WEB files of the site to a new device and changing the IP addresses on the DNS servers. Beforehand, we modified the new system so that all the materials were served from the old server. As, except for the above modification, both systems were identical, using one database and one file repository, the change

MORE

IMPLEMENTATION

was unnoticeable to the end user. It did not matter to them whether they used the old or the new server because the content served was the same. Propagation of the new DNS records may take up to 72 hours in extreme cases, so during this time we refrained from making any changes in the system.

Once we made sure that there is no longer any traffic in the old WEB system, we began to migrate the files. We started with images and documents, as they were the smallest in size. After transferring those files the longest migration of video files began.

The client wished to convert all the video materials to two mp4 formats (SD and HD) and .webm so to prepare the system for the HTML5 player and play the materials on mobile devices. The so-far used solutions - a flash player and video in the FLV format - were not modern enough. Also, the client wished to add an intro and an outro with the **MEDtube** logo to each video.

So, we wrote a special program running on an external server that searched the database for video files in the old format, retrieved those files from the old server, added to them the intro and outro, then created three video files in the required formats along with a thumbnail image. The program then transferred all the files to the new storage, and finally updated the record in the database informing the system that this material can already be used in its new version.

This program was autonomous. It operated automatically without any intervention of administrators. Such a migration took over a month.

After transferring all the video files and making sure that the system worked properly the old shared hosting was terminated.

RESULTS

DATABASE TRANSFER WITHOUT SYSTEM DOWNTIME

INCREASED PERFORMANCE SPEED OF THE SITE

CONVERSION OF ALL MATERIALS TO MP4

In case of very complex projects, such as **MEDtube**, it is extremely difficult to perform a zero-downtime migration. The above example has proved, however, that it is doable with good planning, developing the right architecture and using well-thought solutions.

> *By developing the right architecture and implementing well-thought solutions we succeded in transfering the client's data without any downtime.*

DOTINUM