**Article ID**Tip-002

**Article Topic**Website Optimization

**Article Title**Serve Resources from Different Hosts

# Preface

Today all modern browsers are multithreaded. Means they can server the contents while downloading more than one resource simultaneously. But there are still a few restrictions. There can’t be more than 2 parallel requests threads from the same host.

# Explanation

When your browser finds an embedded resource in your web page it creates a request from the host to deliver that resource. The browser adds these requests in the host wise resource download queue where they are served on the FIFO basis. If two resources are on the same host and the third one is on the other host it may be possible that you are receiving the third one before the first or second are downloaded. Here are few facts:

1. Resource Download Queue

Each browser adds the resource request in a download queue. While parsing the HTML contents if the browser comes across any embedded resource like an image file or a style sheet file. It will add it to the download queue.

1. Priority on the Bases of Resource Type

Every embedded resource download request will go into the queue as and when the parser comes across it and continue to parse the rest. In case of a JavaScript file however the case is bit different. If the browser finds a script file embedded in the Webpage it will stop parsing the rest of the page and will wait for that file to be downloaded first. So if that file is late in the queue, the wait and parsing time is high.

1. Number of Requests per Host

Today the browsers are multithreaded to serve the content by parallel download and reduce the wait time. But according to the ISDN rule this limited to 2 parallel connections to the same host. So if there are more files embedded in a Webpage. It is going to take a considerably high time to fetch those resources.

# How to Resolve

If the contents are served from different hosts the browser can create separate threads for them and have them downloaded parallel. However actually it impossible to have the separate host for each resource, we can have the separate hosts for different type of resources. In real scenarios having separate resource type hosts are also difficult, so we can fool the browser by serving the resources from different Subdomains, like Images from *image.domain.com* and scripts from *script.domain.com*. Here are a few points in short

1. Create the separate Subdomains for different type of objects.
2. Separate user images from the interface images and serve them from different Subdomains.
3. You can use the alias in place of actually creating subdomain in the DNS server like image.domain.com -> domain.com. However efficiency is at its max when you have not more than 4 alias per domain.

# Conclusion

In case of rich user interface sites where there are high number of images and media files are embedded this technique is quite useful. Simply split the resources across multiple hosts and serve your users a much faster.