Network optimisation for Google Apps

This document describes how to configure your network to maximise performance for users on Google Apps. Varying levels of technical knowledge are necessary depening on how important interactive performance of Google Apps is for your users.
Optimising your network configuration will help your users with Google Apps and possibly other Software as a Service tools. The following advantages can be achieved by considering the points made in this document:

- Latency to services provided by Google is reduced. Reduced latency normally results in quicker response from user actions in each application.
- Internal routing of requests to Google's services may be reduced.
- Upload and download times to and from Google can be reduced. This can help with users watching Google Apps internal videos as well as uploading attachments.
- Accessing services via the web is made more attractive than traditional thick client tools such as used for email.

The general rule for improving user performance for Google Apps is to design for each user accessing www.google.com over HTTPS as efficiently as possible. In general this approximates a simple user's home Internet connection.
In a corporate environment, it should be possible to provide at least as good network performance to Google as is possible from a standard home broadband connection. There is normally no benefit nor reason to slow this down.

Bandwidth

Most administrators of Google Apps concentrate on bandwidth implications when adopting Google Apps yet experience has shown that beyond a minimum bandwidth, latency is more important. Moreover, for companies which already make regular use of SaaS applications or provide users with general access to the Internet, bandwidth is not heavily impacted. In most cases, very limited bandwidth will turn into increased latency for end users. For this, we recommend focusing on latency first before turning towards available bandwidth.
Please see our document for *Evaluating external network bandwidth* to see how to check your bandwidth offered to users in preparation for Google Apps.

Latency

Network latency will affect users directly for each action they perform on most SaaS applications. Google goes to great lengths to minimise the impact of latency in the following ways:

- Constant analysis of user connections from the Internet to improve performance to and from Google.
- On-going efforts to get as close to end users on the Internet as possible in terms of network hops.
- Use of AJAX and other technologies in web applications to minimise the end-user impact of latency (e.g. pre-fetching and background loading of content in web browsers).
- Constant aggregate monitoring of latency to all of its services all over the world.
- Providing tools to reduce latency as much as possible (Google Chrome, Measurement Lab, etc.)

The simplest way to view latency for your users is to define it as *the time taken for a single, unique request to transmit from a user at a computer to Google Apps and for a result to be returned and rendered*. You can obtain a rough approximation of latency for end users in the following ways:

Traceroute

Traceroute (or on some systems tracepath) is a tool normally found on any modern operating system. From a user's machine run the following command:
`tracepath mail.google.com` on Linux
`tracert mail.google.com` on Windows

Example:

```
C:\>tracert mail.google.com

Tracing route to googlemail.l.google.com [209.85.229.18]
over a maximum of 30 hops:
  1    71 ms   102 ms   204 ms  192.168.1.254
  2     *        *        *     Request timed out.
  3    20 ms    17 ms    18 ms  10.1.2.222
  4     *        *        *     Request timed out.
  5    19 ms    17 ms    17 ms  87-194-0-206.bethere.co.uk
[87.194.0.206]
  6    20 ms    17 ms    18 ms  64.233.175.27
  7    27 ms     *       26 ms  66.249.95.170
  8    27 ms    23 ms    24 ms  209.85.251.231
  9    46 ms   203 ms   102 ms  209.85.243.81
 10    26 ms     *       27 ms  ww-in-f18.google.com
[209.85.229.18]
Trace complete.
```

The last trace line should say how long the request took to complete. A good value here is something less than 100ms. Ideally, it should be closer to 10ms. This approach may not include a route used by an HTTP proxy.

Timing a fetch

You can get a more accurate understanding for how long it takes to access Apps using a combination of two tools. Linux systems and Macs normally have some variety of these tools installed. You may need to obtain similar tools for Windows.

Example:

```
:~$ time wget --spider -r --user-agent="Mozilla/5.0 (X11; U;
Linux x86_64; en-US; rv:1.9.0.13) Gecko/2009080317 Ubuntu/
8.04 (hardy) Firefox/3.0.13 GTB5 GTBA" --header="Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8" http://calendar.google.com/a/MYDOMAIN.com
...several redirects because there are no cookies
available...
--13:36:23--  https://www.google.com/a/MYDOMAIN.com/
ServiceLogin?service=cl&passive=true&nui=1&continue=https%3A%2F%2Fwww.googl
          => `www.google.com/a/MYDOMAIN.com/
ServiceLogin?service=cl&passive=true&nui=1&continue=https:%2F%2Fwww.google.
Connecting to www.google.com|209.85.229.103|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 9,885 (9.7K) [text/html]
200 OK

FINISHED --13:36:23--
Downloaded: 0 bytes in 0 files

real 0m0.614s
user 0m0.008s
sys 0m0.004s
```

The time shown above is a relatively good time for this complete process.

DNS

A significant source of slow Internet requests are badly configured DNS clients and possibly servers. You can see how long it takes to resolve some of the commonly accessed hosts by issuing name resolution requests.

It is also worth timing how long a request to your configured DNS server takes. While caching will normally eliminate this source of latency, this can still have effects on overall performance.

Example:

```
~$ time host calendar.google.com
calendar.google.com is an alias for www3.l.google.com.
www3.l.google.com has address 209.85.229.113
www3.l.google.com has address 209.85.229.102
www3.l.google.com has address 209.85.229.138
www3.l.google.com has address 209.85.229.101
www3.l.google.com has address 209.85.229.139
www3.l.google.com has address 209.85.229.100

real 0m0.163s
user 0m0.000s
sys 0m0.000s
```

The time shown above is a relatively good time for a complete DNS lookup.

SSL & Proxies

Google Apps is often configured to only allow SSL connections for all users. Moreover, regardless of this setting, Google requires that any user only authenticate over a secure connection (SSL or TLS). This applies to any client connection such as web browsers and email clients. For this reason, secure connections to Google are required.

When SSL is enforced through the setting in the Admin panel in Google Apps, all content data is transferred over secure connections between the user's browser and Google. Standard, common blocks of data such as common JavaScript files and images may still be transferred over non-encrypted transport.

For this reason, routing such data through a proxy or other application layer gateway will only serve to slow down communication and also provides a single point of failure. In general, as the data transmitted within Google Apps is most often unique and state-based, there is no benefit to passing it through a proxy or other tool which inspects content.

Route to Google

Using the trace tools mentioned above (e.g. traceroute) it is possible to approximate the route from a user's computer to Google. The overall aim of this approach is to reduce the total round trip time. Identifying the hops that consume the most time is the best way to see how to improve latency. If you are in control of those hosts, we recommend either routing around them or else improving their performance for requests from users on Apps. If the source of the latency is beyond your network, you may wish to consider alternate Internet gateways and possibly a different ISP.

The tools available in the Measurement Lab mentioned can also provide a lot of detail regarding your Internet connection performance. We recommend using these as another method of assessing the overall Internet access performance available.

Peering

Going beyond this such as peered networks with Google and similar combinations normally makes little difference to the performance. Google continuously monitors end-user latency and performance and assumes no special arrangements have been made with those users. As a result, Google may change network paths, routes and load-balancing to provide the best performance. It is not feasible to maintain special connections with customers and normally would provide little benefit. One key point regarding this is that Google routinely shuts down entire data centres, so relying on specific connections with Google could reduce reliability.

Offline access

Enabling offline access for tools which require a reasonably large collection of recent data can dramatically affect overall network bandwidth. In particular, for technologies like email, network behaviour will be similar to traditional email clients as data syncrhonisation is used instead of immediate direct access.

Google Mail

As soon as offline is enabled in email, Google Mail will attempt to download 10,000 most recent mail messages the user's web browser. This process will start with most recent email and then work backwards in reverse chronological order, downloading attachments only at the end of the synchronisation. The offline email settings within Google Mail can change this behaviour considerably but is left to the user to customise directly. Initially, this process can take some time and will consume a consistent amount of bandwidth. It is a background process that may be interrupted and resumed automatically.

They key difference with offline enabled is that each message (if configured to be synchronised for offline access) will be retrieved to the client and subsequent synchronisations may cause the data to be retrieved (although this is not common). As a result, unlike the experience with offline disabled in which the user's total bandwidth is proportional to the new informaition presented within the browser, with offline enabled this will no longer be the case the bandwidth use will increase.

Google Calendar

Offline access for Google Calendar requires much less data to be exchanged with Google than is necessary for email and so synchronisation can complete far more quickly and overall bandwidth required is dramatically lower. Prior to offline in which total bandwidth is proportional to new information presented in the browser, enabling offline offline will increase bandwidth consumption to some degree.

Google Documents

Enabling offline in Google Documents will have similar affects as email, however data generally does not change as frequently with documents as it does with email. Similarly, as Google Mail must also handle potentially very large attachments, only binary additions to documents like images will cause further bandwidth demand. In general the same effects for enabling offline access in email will apply to documents, but this will still require significantly less bandwidth than Google email and somewhat more than Google Calendar.

Browsers

We have seen from customers that different browsers will provide very different results in terms of performance. Plenty of comparisons between browsers are available online. To help understand the comparisons as it relates to Google Apps it is wise to focus on some key points:

- Google Apps makes use of a lot of JavaScript for speed. Having very fast JavaScript engines makes a lot of difference.
- Users using Google Apps will tend to keep the browser open all day. This often means a user will have a few tabs open at any one time. As a result tabbed use is important while cold-start times may be less relevant.
- The total memory used often increases with time while a browser is open. This will typically extend through the day with Google Apps and involve several tabs open continuously.

Chrome or FireFox

Experience has shown that using Chrome, Firefox or Safari (in that order), provide the best user experience for the web in general (not even specifcally for Apps). Chrome can be used without actually installing it separately by adding a plugin for IE8: http://code.google.com/chrome/chromeframe/
Using one of the recommended browsers natively will normally save time, effort and the need for reading the sections below.

Settings in IE 8 that may be useful

Increasing the number of concurrent connections per web browser context:

There is a simple registry hack that will allow you speed of browsing
Step 1: Go to the registry from **regedit** in the run box
Step 2: Navigate to **HKEY_CURRENT_USER**.
Step 3: Navigate through **Software** ->**Microsoft**->**Windows** ->**Current Version**->**Internet Settings**.
Step 4: Change the value to 8 or 10, close the editor and reboot.
This is mentioned in one of the documents I've attached.

You shouldn't really see any warning prompts from Google Apps properties (that is normally an indication that a better browser setting is possible). While Google does not make use of any AcitveX controls, IE may classify some plugins as this (e.g. Flash, Gears).

"This page contains both secure and nonsecure items"

- When you receive the error message, click **Yes**.
- In Internet Explorer, go to Tools, **Internet Options**, click the **Security** tab; make sure that in "Select a zone..." window that **Internet** is selected.
- Click **Custom Level** and scroll down about half way to "Display mixed content" in the **Miscellaneous** section.
- Change it from **Prompt** to **Enable**.
- Click **OK**, **Yes**, and **OK**. The change should take effect immediately.

Open new windows as tabs:

- Click the **Tools** then **Internet Options**. There are four sections: Home page, Browsing History,
- Search and Tabs.
- Click the **Settings** button in the **Tabs** section then look for "**Open links from other programs in:**"
- Click the radio button marked "**A new tab in the current window**"