

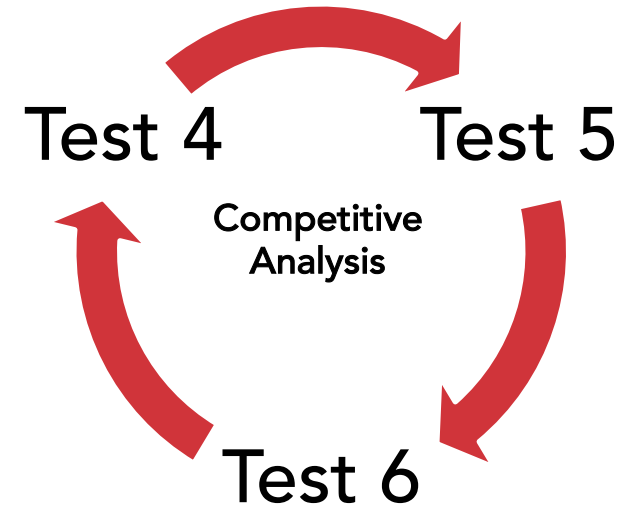
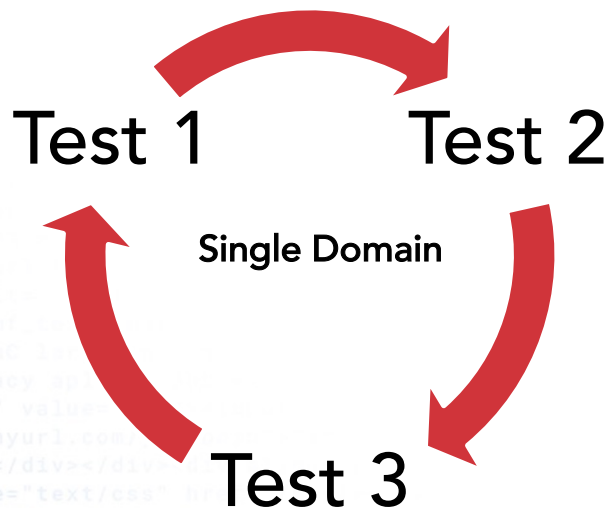
GS Advanced Services Site Analysis Portal

June 2022

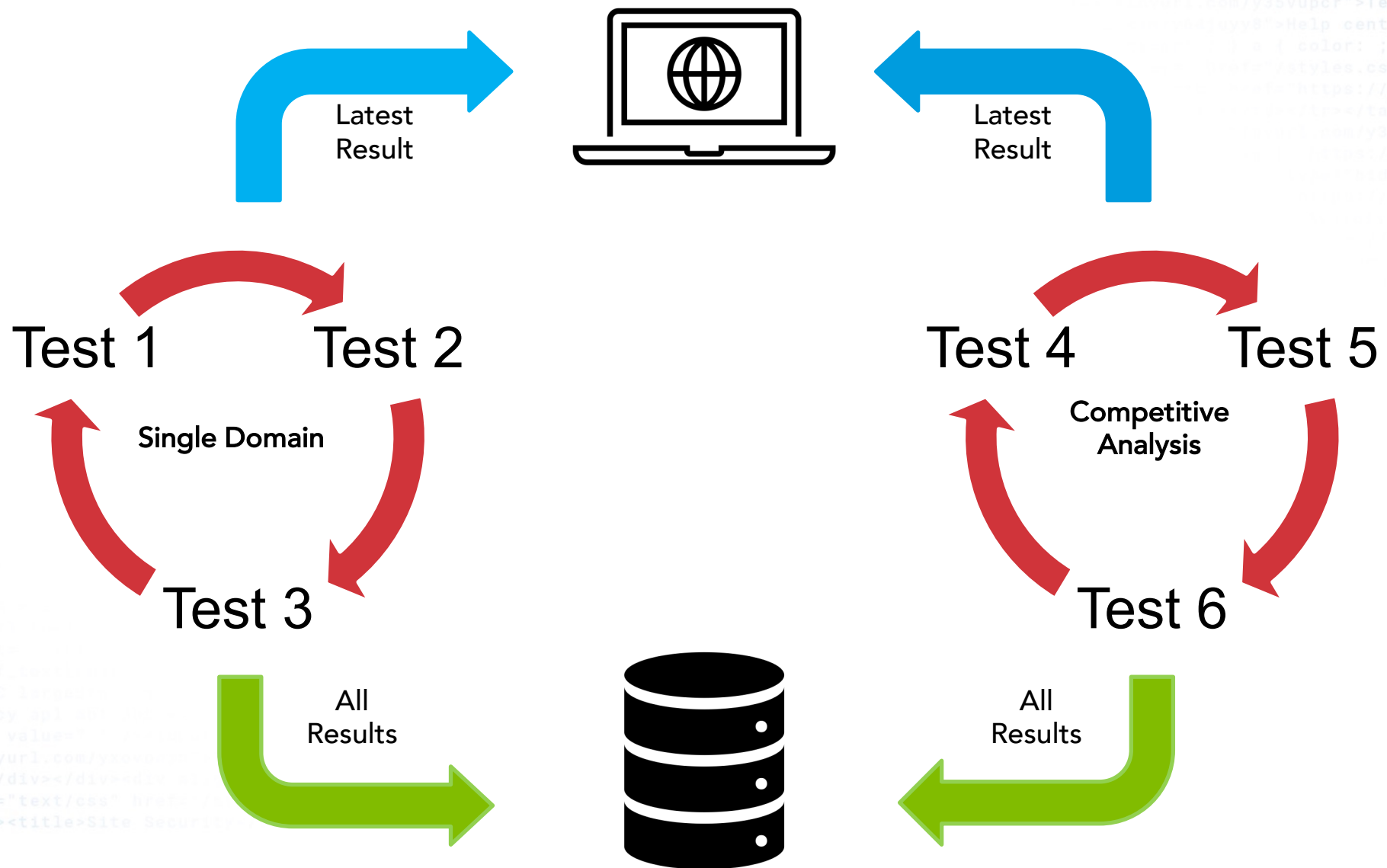


Continuous Testing

- Single Domain and Competitive Analysis analysis is driven by Sitespeed.io
- URLs are tested 3 times on an hourly basis over a LAN and LTE connection
- The number of unique tests and the number of URLs tested in parallel depends on the number of other tests running on the system

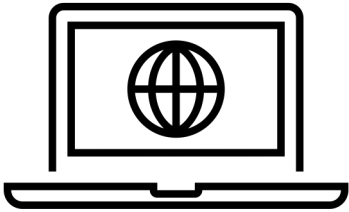


Data Flow



Site Analysis Portal

sitespeed.akamai.com

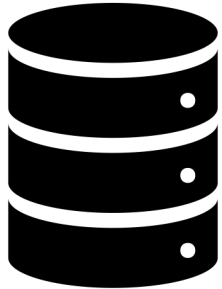
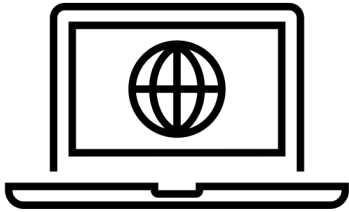


Single Domain

Competitive Analysis



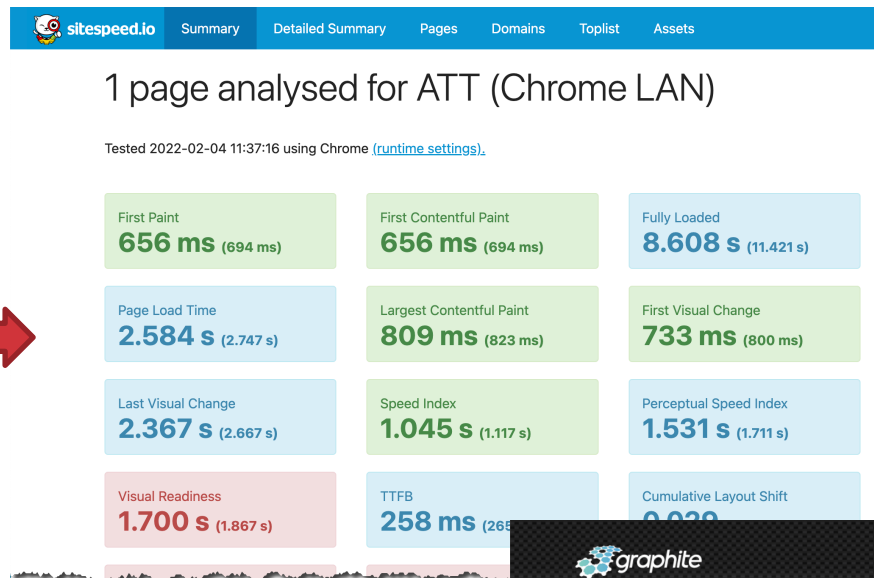
Grafana



graphite



Interfaces

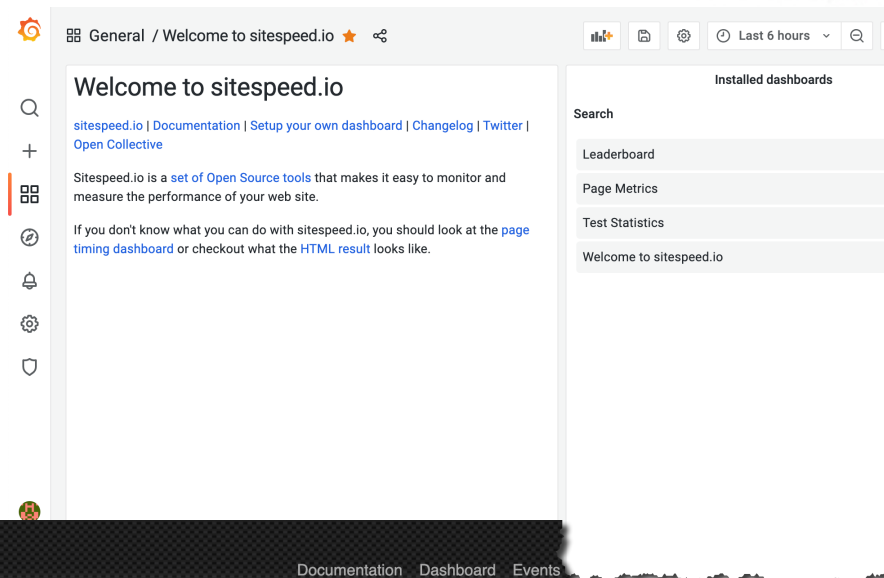


Sitespeed.io

- Open access
- Dynamically updated

Graphite

- Restricted access
- Used only for health check



Grafana

- Restricted access
- Rich dashboard capabilities

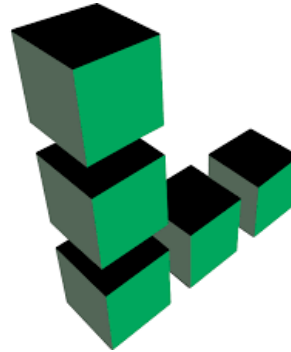
Hardware Architecture

Sitespeed.io



- CentOS 7
- 8 CPU, 16GB RAM, 320GB HDD
- Static public facing IP

Linode



- Sitespeed machines located in:
 - New Jersey
 - Dallas
 - California
 - Toronto
 - London
 - Frankfurt
 - Singapore
 - Tokyo
 - Mumbai
 - Sydney

Graphite/Grafana



- CentOS 7
- 16 CPU, 32GB RAM, 1TB HDD
- Static public facing IP
- Located in New Jersey

Software Components

Sitespeed.io



- Sitespeed.io running in Docker
- nginx web server
- HTTPS 443



Graphite/Grafana



- Graphite running in Docker
- HTTP 8888
- Carbon TCP 2003



- Grafana runs natively
- Embedded Apache server
- HTTPS 443



Use Cases

- Single source of data for TBAs, web site analysis, general research, etc.
- Collect data for competitive benchmark studies
- One stop for all vital metrics
 - Domain and Page Metrics
 - Site Construction
 - 1st & 3rd Party Information
 - Largest JS, CSS, and Images
 - Slowest First and Third-Party Resources
 - Prescriptive Guidance
 - Core Web Vitals
 - Waterfall
 - Filmstrip
 - Video
- Synthetic testing is a perfect complement to mPulse RUM data
- Data can be used for reports and/or used to drive live customer discussions

Why Sitespeed.io?

- Created in 2012 as an open-source project
- Easy to configure and run!
- Capabilities continue to evolve/improve with 700+ developers
- Full integration with tools such as WPT, CrUX, PSI, Lighthouse, S3, GCS, Graphite, InfluxDB
- Well documented and great support (Github, Slack, Facebook, Twitter)
- All test results saved in a database
- Extensive scripting support for multi-page journeys
- SPA support
- *The list goes on ...*

Folder Structure

Single Domain

Competitors

Dynamically changing symbolic links

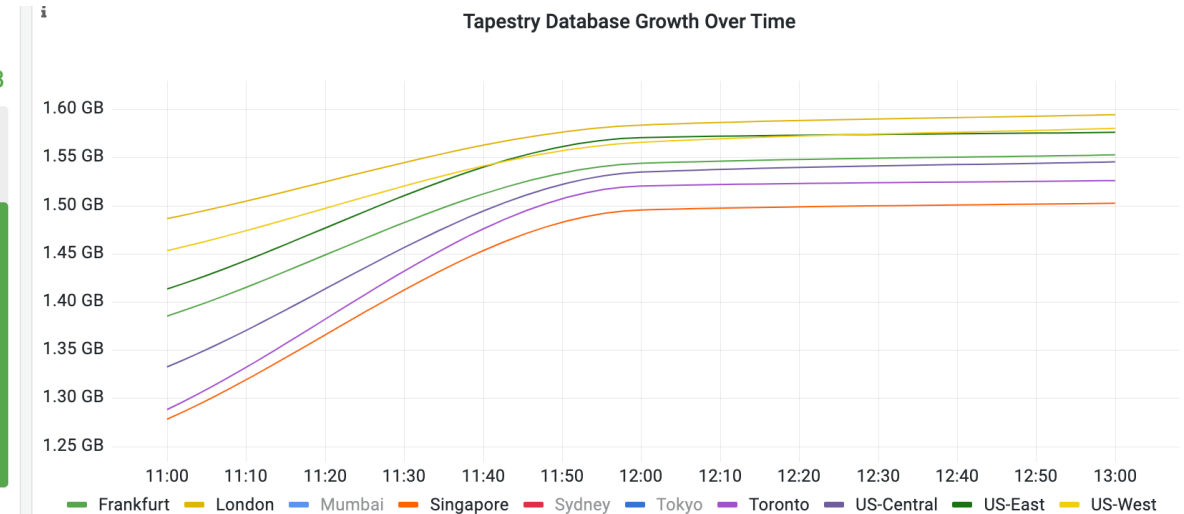
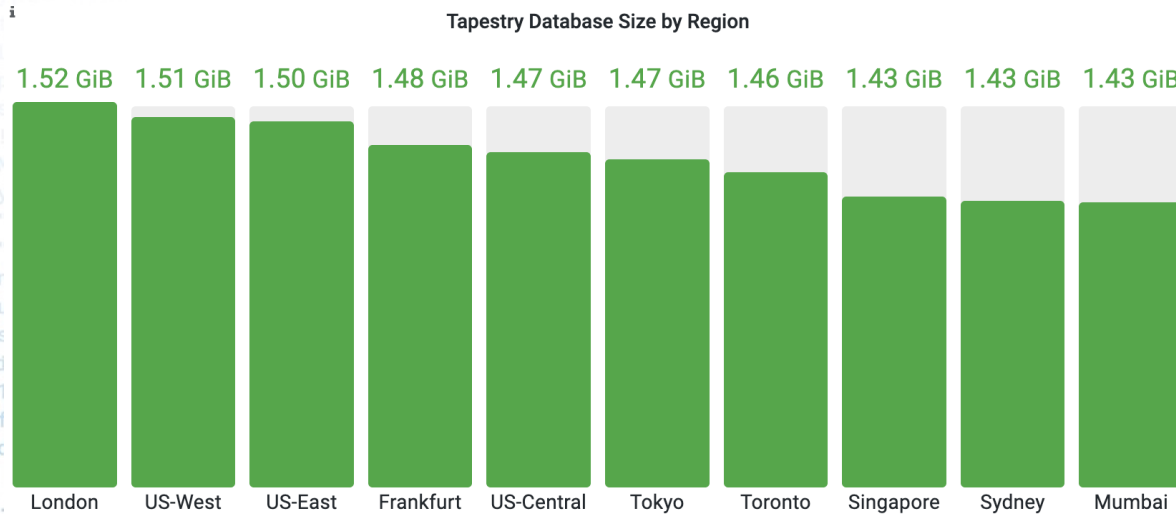
```
[greg@sitespeed2 ~]$ tree -d -L 2 tld comp portal
tld
├── sitespeed-result
│   ├── ATT
│   ├── MSN
│   ├── MSNAKA
│   └── Tapestry
├── comp
│   ├── sitespeed-result
│   │   ├── Airlines
│   │   ├── ATT
│   │   ├── eCommerce
│   │   └── Financial
│   ├── portal
│   │   ├── comproot -> /home/greg/comp/sitespeed-result/Financial/2022-02-04-09-44-48
│   │   ├── favicons
│   │   ├── images
│   │   │   ├── Airlines
│   │   │   ├── ATT
│   │   │   ├── eCommerce
│   │   │   ├── Financial
│   │   │   ├── MSN
│   │   │   ├── MSNAKA
│   │   │   └── Tapestry
│   └── tldroot -> /home/greg/tld/sitespeed-result/ATT/2022-02-04-09-36-52
└── 21 directories
[greg@sitespeed2 ~]$
```

Logs

```
[greg@sitespeed2 ~]$ ls -l logs/
total 8932
-rw-r--r--. 1 greg greg 1237315 Feb  4 14:49 comp.msg.log
-rw-r--r--. 1 greg greg   13323 Feb  4 14:49 comp.run.log
-rw-r--r--. 1 greg greg 7855748 Feb  4 15:38 tld.msg.log
-rw-r--r--. 1 greg greg   24282 Feb  4 15:38 tld.run.log
[greg@sitespeed2 ~]$
```

Database Sizing

- Database sizing is driven by the frequency of data collection and the amount of time data will be saved
- Table size is defined at creation, which makes storage planning deterministic
- Testing consisting of 10 URLs done on an hourly basis that will be saved for 13 months will have an initial size of 1.5GB and then will increase ~75MB per run



Setup and Configuration

Step 1 – Create URL seed file

eCommerce.txt (Competitive Analysis)

<https://www.abercrombie.com/shop/us> Abercrombie Abercrombie
<https://shop.lululemon.com/> Lululemon Lululemon
<https://www.gap.com/> Gap Gap
<https://www.zara.com/> Zara Zara
<https://www.ae.com/us/en> American-Eagle American-Eagle
<https://www.apple.com/> Apple Apple
<https://www.etsy.com/> Etsy Etsy
<https://www.nike.com/> Nike Nike

ATT.txt (Single Domain)

<https://www.att.com/> Home ATT
<https://www.att.com/deals/> Deals ATT
<https://www.att.com/wireless/> Wireless ATT
<https://www.att.com/buy/phones/> PLP ATT

Step 2 – Push seed file(s) out to Linode servers

Step 3 – Configure a testing schedule and push schedule to all Linode servers

All scripts and documentation are available on Github

Operation

- Testing is driven by one script called master.sh
- Administration of scripts and server functions are driven by a script called push.sh

```
loth-mpurw:Sitespeed gwolf$ ./push.sh

USAGE push arg1 [arg2 arg3]

DESCRIPTION Automates the distribution and/or execution of key scripts across all Linodes.
Intended to run from a local machine. If running from a Linode, be sure to modify the
source path of the scripts.

The following options for arg1 are available:

    all      Copies Sitespeed scripts across all Linodes
    config   Copies config.json to ~/tld and ~/comp across all Linodes
    cron     Modifies crontab across all Linodes. Requires:
              arg2 = list|update|delete
    docker   Executes docker commands across all Linodes. Requires:
              arg2 = Version of Sitespeed (xx.y.z)
    index    Copies index.html across all Linodes
    log      Checks for errors across all Linodes. Requires:
              arg2 = check|delete
    master   Copies master.sh across all Linodes
    nginx    Sets Web permissions across all Linodes
    reset    Deletes key data across all Linodes
    seed     Copies test URL seed file to all Linodes. Requires:
              arg2 = tld|comp
              arg3 = Name of URL seed file
    update   Updates YUM packages across all Linodes
```

Utilities & Documentation

- System check → syschk.sh
- Web permissions → nginx.sh
- Reset EVERYTHING → reset.sh
- Docker installation → install-docker.sh
- Graphite → start-graphite.sh
- Global administration → push.sh
- Fully documented → <https://git.source.akamai.com/projects/A2S/repos/sitespeed>

