Varnish and aiScaler, by aiScaler, December 16, 2013

1. Introduction and features

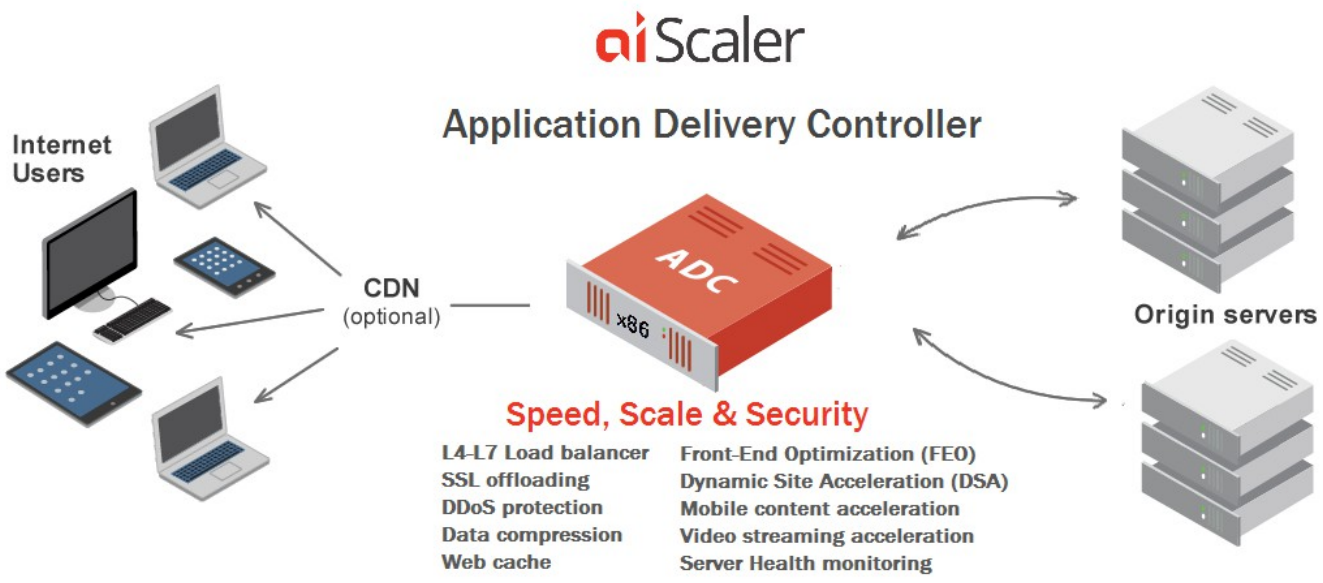
Varnish and aiScaler are very different products. Varnish is an open source caching solution, usually installed on the origin server and sharing resources with HTTP server software, like Apache or Nginx.

aiScaler is an all-in-one application delivery controller (ADC), normally installed as a reverse proxy on a dedicated machine. It features caching, but caching is only one feature of the entire package of an all-in-one ADC. Other key features include: SSL offloading, DDoS protection, multiplexed session management, mobile device detection and IP-based geocontent delivery. It has built-in support for SNMP to communicate with industry standard routers, switches and servers and supports peer-aware clustering for redundancy, high availability and horizontal scaling. All the features above are not available in Varnish, as Varnish is mostly a pure caching solution. The cluster awareness makes aiScaler suitable for a private cloud, multi-cloud or hybrid cloud environment. Many aiScaler customers don't even use the caching features and only buy aiScaler for it's other features.

aiScaler being a for-profit corporation, also provides professional support, which are billed by the hour or included in the purchase of a license. aiScaler is much more similar to other manufacturers of Application Delivery Controllers, like Citrix and F5, than it is to Varnish.

Since Varnish doesn't have all of aiScaler's features, the performance and installation comparison in this reports is limited to the caching feature.

In the feature comparison table, we also included Varnish+, which is the paid version of Varnish. It has a bit more features and professional support, than it's free version. However, as you'll see in this report, it still lags behind aiScaler, both in terms of features and performance, while the price starts at $13,000 USD, making it more expensive (aiScaler starting at $9,995USD or $0.14 USD/h on the AWS cloud)



2. Feature comparison table

	aiScaler	Varnish	Varnish+
Load Balancing	✓	✓	✓
SSL Offloading	✓	✗[1]	✗
SSL Origins	✓	✗	✗
SSL Cache	✓	✗	✗
POST Cache	✓	✗[2]	✗
URL Rewrites	✓	✓	✓
Multiple Hosts and Origin Tagging	✓	✓	✓
Cookie Driven Caching	✓	✓	✓
Cache Expiration Using a Request	✓	✓	✓
Cache Expiration Using a File	✓	✗	✗
Cache Expiration Using a Web Form	✓	✗	✗
Gzip Compression	✓	✓	✓
Clustering	✓	✗	✗
CLI	✓	✓	✓
SNMP	✓	✗[6]	✗
Web Stats	✓	✗[3]	✓[4]
Real-Time Stats	✓	✓	✓
Server Health Checking	✓	✓	✓
HTTP DoS Mitigation	✓	✗	✗
Fallback on Error	✓	✓	✓
Slow Start App Protection	✓	✗	✗
Customer Plugins	✓	✓	✓
Mobile Devices Detection	✓	✗	✓[4]
Geo-IP Location	✓	✗[5]	✗
Web-Based Deployment	✓	✗	✓[4]
Resource Prefetching	✓	✓	✓
Email Alerts	✓	✗	✗

[1] Why no SSL?
[2] Varnish replace POST with GET
[3] Varnish Administration Console (only for customers with service agrement)
[4] Varnish Software Prices
[5] Available with a custom module from GitHub
[6] Available with a custom module from SourceForge

3. Caching modules compared: features and ease-of-installation
aiScaler has more ways to configure caching, yet being easier to deploy (see configuration example below). aiScaler performs especially well with the caching of dynamic websites. Dynamic websites are difficult to scale by nature, but aiScaler manages to obtain a high cache ratio, thanks to a native support for SSL websites, POST request caching and more ways to invalidate cache, using content driven or web-form driven cache expiration.

aiScaler features a web-based deployment tool, with an intuitive GUI, which allows you to configure, accelerate and test any website (www.aiscaler.com/deploy). Varnish does not have a web-based deployment method. Instead you have to learn Varnishes own coding language, called “Varnish Configuration Language” (VCL)

Below is a configuration file example for caching a Wordpress site. aiScaler has this configuration file as a default template in their deployment tool, just like it has templates for Joomla, Magento, rightscale, vBulletin and PrestaShop. aiScaler configuration is so simple and quick, that aiScaler engineers give away free tailored configuration files. A customer can request this for his or her website, after which aiScaler creates a configuration that is specific to the customer application. Most sysadmins will actually not even need this support, as long as they have some knowledge of regular expressions. However, if a customer still needs support for whatever reason, then aiScaler includes 2 hour of free personal support with every license. This may even go so far as walking a customer through the installation with a screen-sharing program.

Configuration examples giving exactly the same instructions.

aiScaler

```
pattern ^wp-(admin|login) regexp 0
pattern ?preview=true simple 0
pattern .js simple 7d no_log
pattern .css simple 7d no_log
pattern .jpg simple 7d no_log
pattern .gif simple 7d no_log
pattern .png simple 7d no_log
pattern / simple 1h
```

Varnish

```
sub vcl_recv {

  if (req.request != "GET" &&
      req.request != "HEAD" &&
      req.request != "PUT" &&
      req.request != "POST" &&
      req.request != "TRACE" &&
      req.reque`st != "OPTIONS" &&
      req.request != "DELETE") {
    return (pipe);
  }

  if (req.request != "GET" && req.request != "HEAD") {
    return (pass);
  }

  if (req.url ~ "wp-(login|admin)" || req.url ~ "preview=true") {
    return (pass);
  }

  remove req.http.cookie;
  return (lookup);
}

sub vcl_fetch {

  if (beresp.status == 404) {
    set beresp.ttl = 0m;
```

```
    return(hit_for_pass);
  }

  if (req.url ~ "wp-(login|admin)" || req.url ~ "preview=true") {
    return (hit_for_pass);
  }

  if (req.url ~ "\.(js|css|jpg|gif|png)") {
    set beresp.ttl = 7d;
  }

  set beresp.ttl = 24h;
  return (deliver);
}
```

4. Performance benchmarks

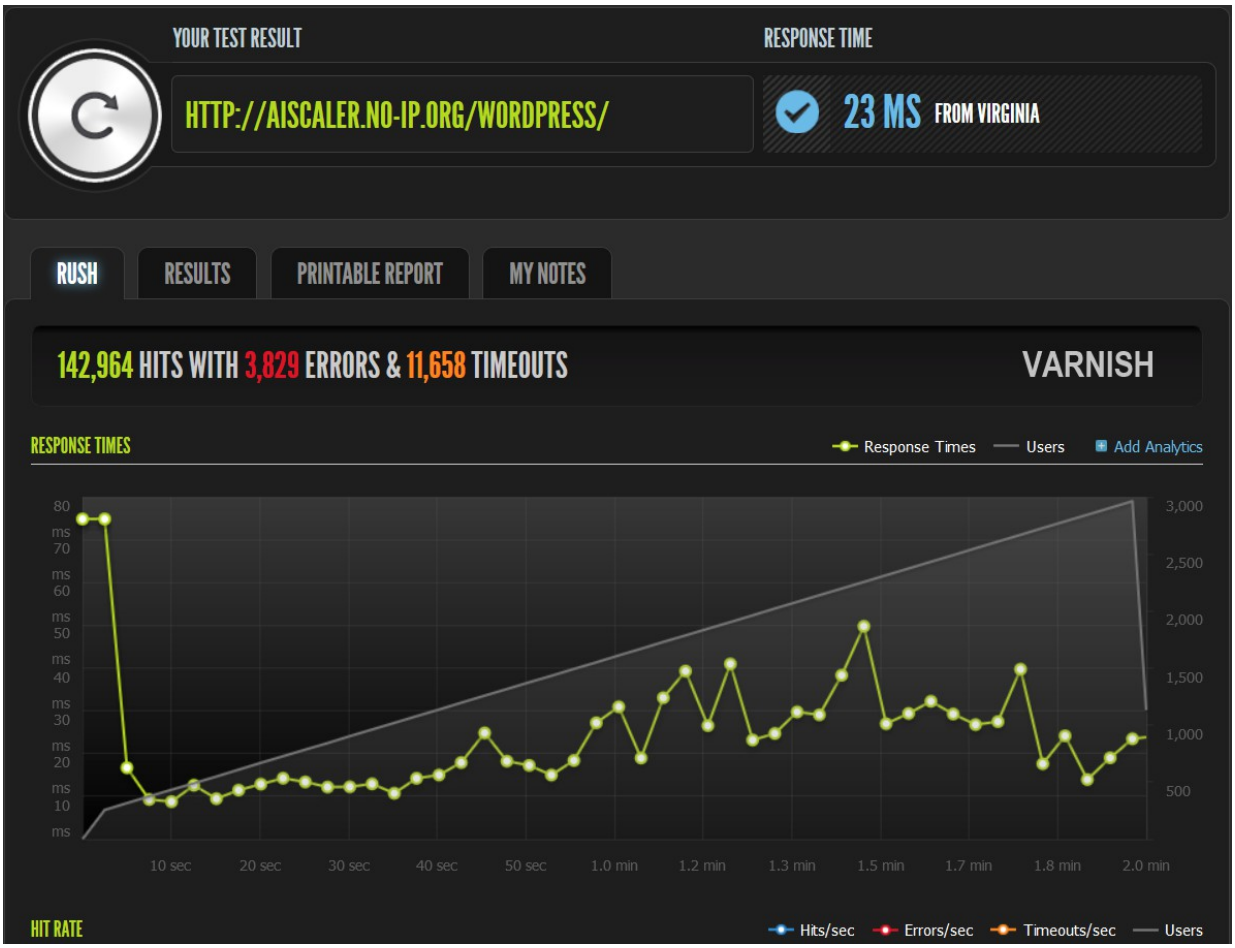
aiScaler and Varnish were both tested on the Amazon cloud in North Virginia, on the same instance type, meaning the same environment and system settings. We also used the same EC2 data-center to launch a load-generating instance and to simulate the origin server.

- Origin: m1.large. This was hosting Wordpress.
- Varnish: m1.large
- aiScaler: m1.large
- load generating instance for ApacheBench: m1.xlarge

aiScaler employs a right-threaded IO design. In practice this means aiScaler limits itself to six threads, no matter how many users are connected. Varnish on the other hand, opens a new thread for every incoming connection, which is inherently not a scalable architecture. Operating systems tend to work very well with thousands of processes running. So at least in theory, aiScaler should perform a lot better under heavy load.

Test 1: Blitz.io

The first test is done with the commercial version of www.Blitz.io. This load testing benchmark simulates an increasing number of users, sending requests to the origin. The point of this benchmark is to go from 0 to 3000 users over a period of two minutes, and measuring the increasing number of errors and timeouts.



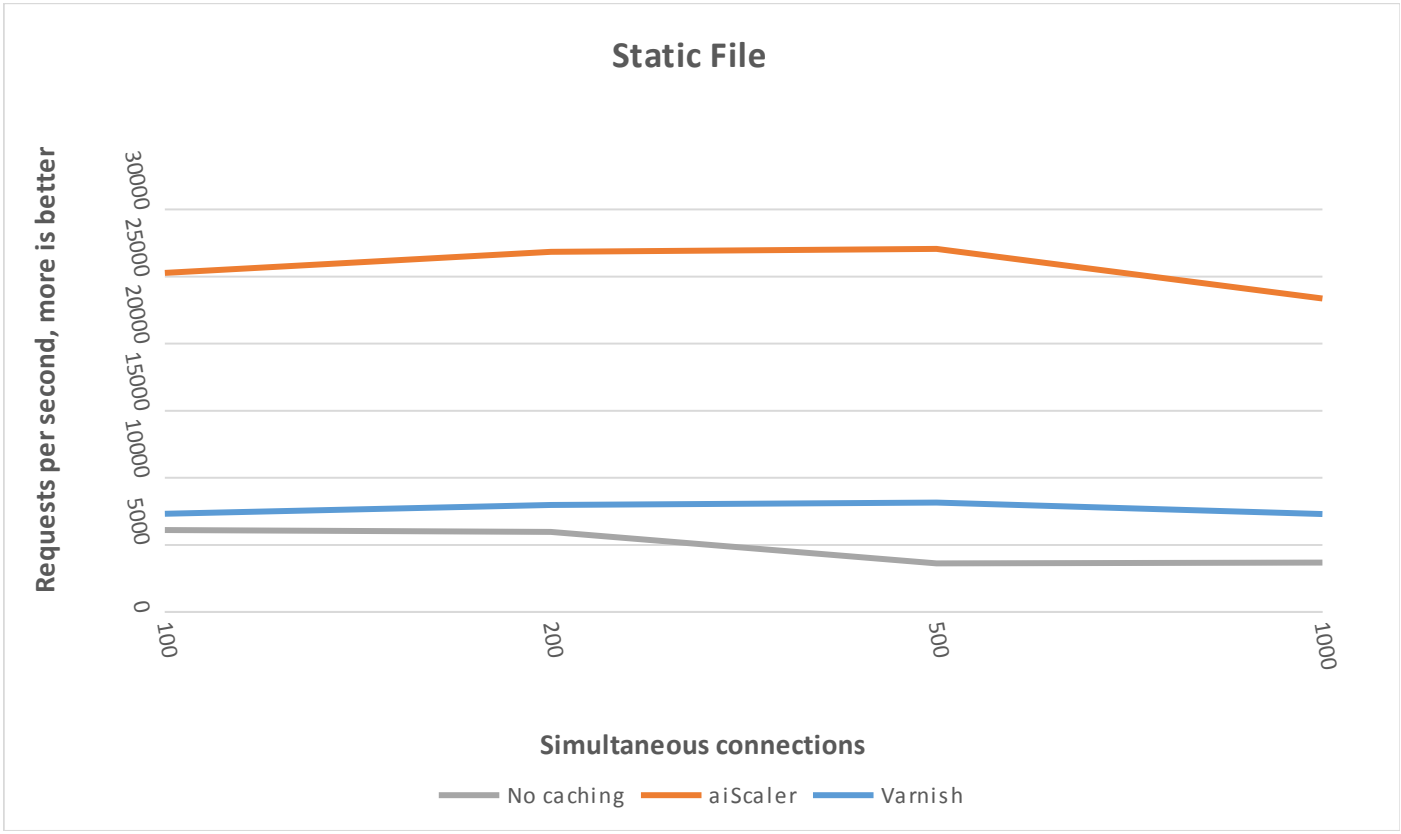
aiScaler manages to serve more requests in the same amount of time (164,516 vs 142,964). What's more alarming though is the difference in errors and time-outs:

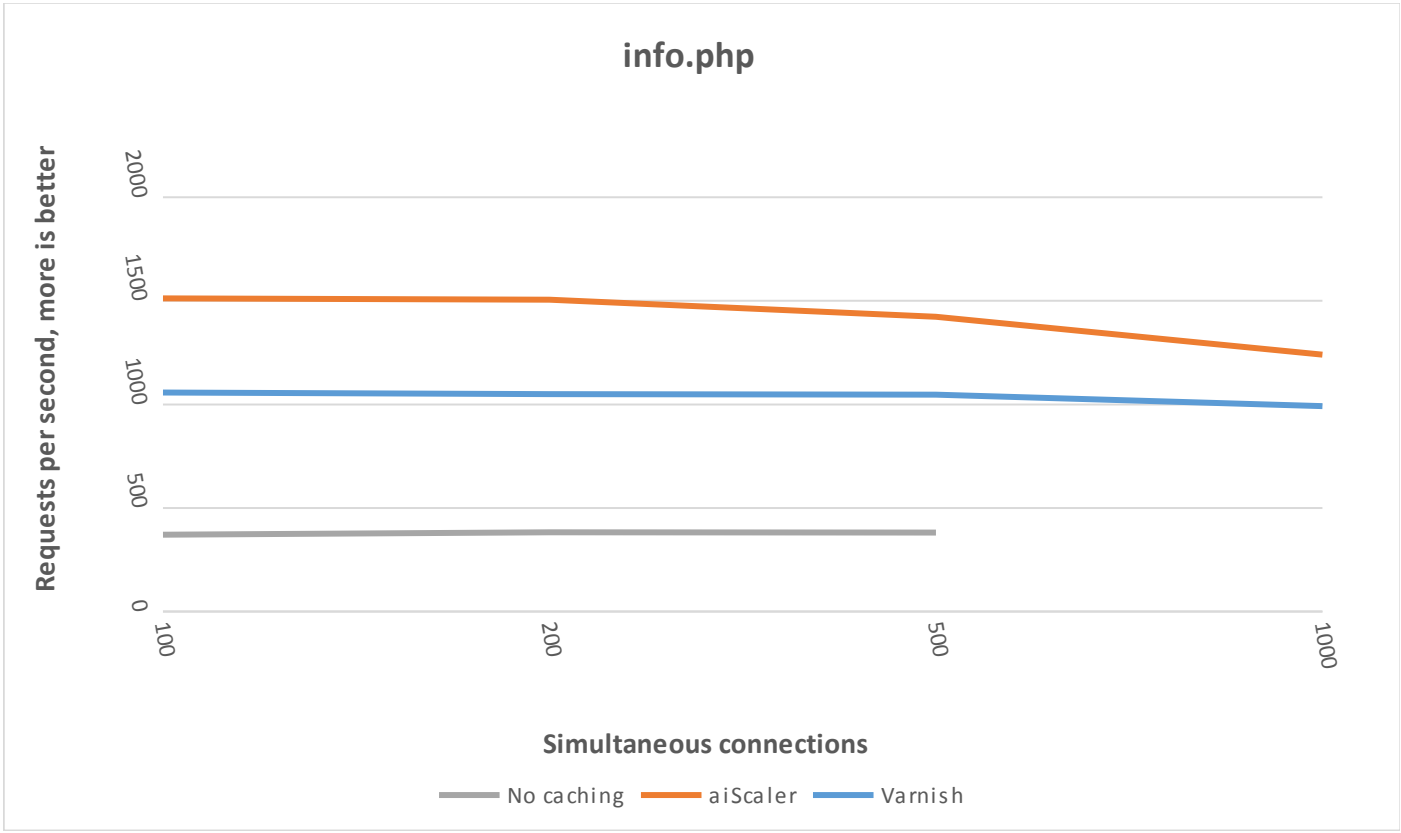
aiScaler: 100% * (1440+279)/164,516= 1.0% errors and timeouts
Varnish: 100% * (3829+11,658)/142,964= 11% errors and timeouts.

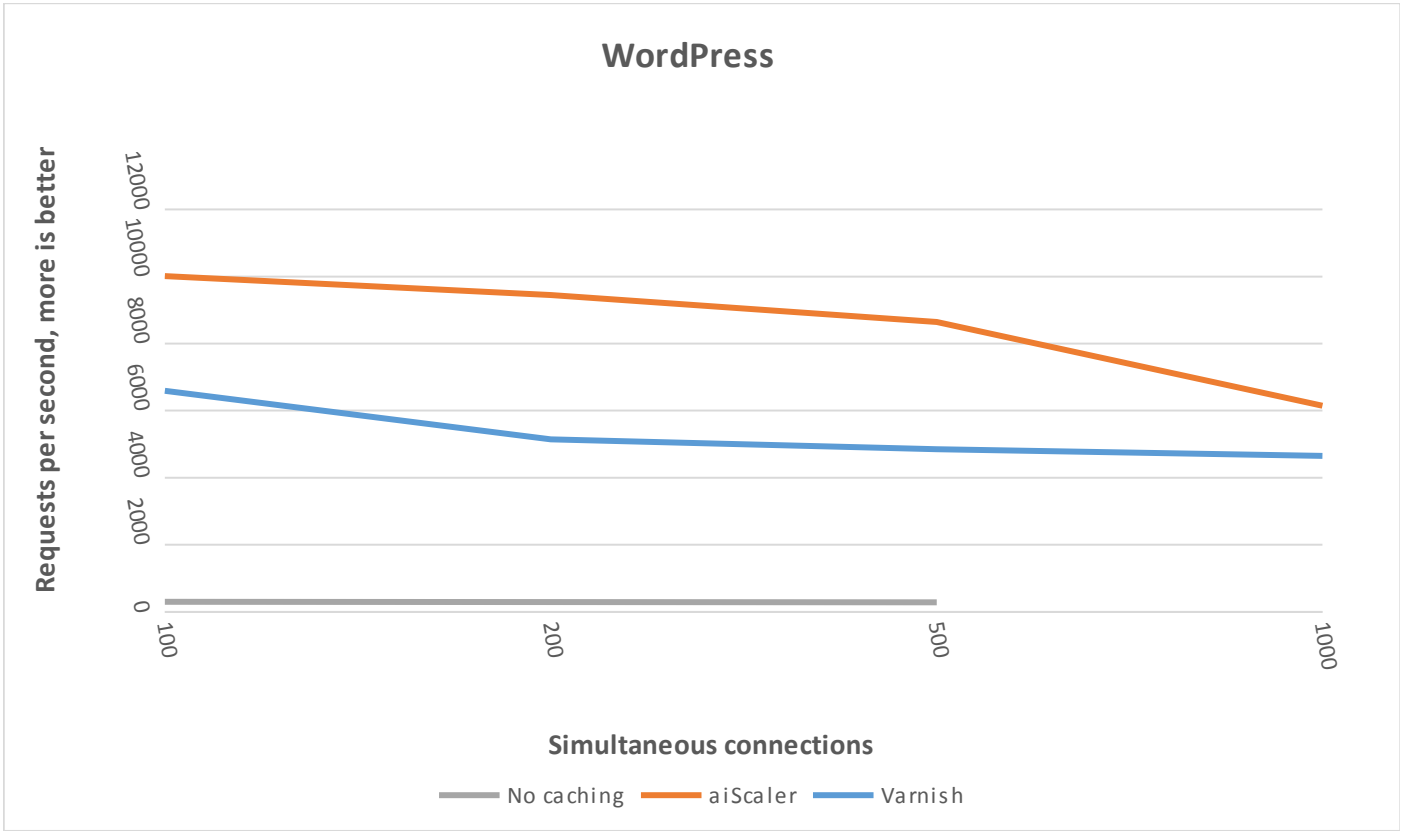
This test was designed was to force errors under heavy load, and it's clearly showing the difference in caching performance between aiScaler and Varnish. This does not even take into account that aiScaler instances can be clustered for horizontal scaling and redundancy. With aiScaler, when you reach the limit of a server, you can easily launch another server next to it, with the same configuration file. Varnish does not offer the possibility of clustering, making it even harder to scale – apart from the worse performance of the algorithm itself.

Test 2: ApacheBench
ApacheBench is an industry standard computer program for measuring the performance of HTTP throughput. Because Varnish doesn't offer any support for SSL - neither offloading nor for caching- we were forced to compare only HTTP performance.

We tested different types of files, with an increasing number of simultaneous connections, while measuring the Requests per second.


Static File
Requests per second, more is better
30000
25000
20000
15000
10000
5000
0
100
200
500
1000
Simultaneous connections
No caching
aiScaler
Varnish


info.php
Requests per second, more is better
2000
1500
1000
500
0
100
200
500
1000
Simultaneous connections
No caching
aiScaler
Varnish



- Both Varnish and aiScaler were faster than the instance that did not use any caching.
- aiScaler results were limited by the network, which was reaching 850-900 Mbps.
- aiScaler had a smaller CPU usage footprint, than Varnish
- The biggest difference between aiScaler and Varnish is visible in the Static File test. This means that aiScaler is able to serve small files much faster than Varnish.
- The results depend greatly on the instance size, server configuration and application. We were using a fairly powerful instance and pages like info.php and a fresh WordPress installation. Real-world results will normally be worse and the difference between a caching and a non-caching solution will become bigger.
- aiScaler was faster than Varnish in every test.

5. Conclusion

Varnish is a great caching product, especially because it's free an open-source. However you will need to spend a lot of time to learn the Varnish Configuration Language, and even then it's limited in terms of caching features, scalability, professional support and performance.

aiScaler is much more than a caching solution. It's more similar to other Application Delivery Controllers, like Citrix Netscaler and F5 products. It has SSL offloading, DDoS protection, multiplexed session management, mobile device detection, IP-based geocontent delivery, support for SNMP and support for clustering.

However, it is possible, to compare Varnish and aiScaler, when it comes to pure HTTP caching (not HTTPS, since Varnish does not support HTTPS caching). aiScaler has more options to expire cache and is much easier to configure, as you can see by the example configuration files. Under heavy load, Varnish served 11% faulty requests, while aiScaler kept faulty requests at 1%. With small static files, aiScaler was 4-5 times as fast, while in PHP and Wordpress the difference was around 50% in favor of aiScaler. aiScaler performed better in all benchmark.

Then there is Varnish+, the commercial version of Varnish. Since it contains the same algorithm as Varnish, performance will be equally mediocre, while features are still not nearly as abundant as with aiScaler. There would be no reason to buy Varnish+, especially given that it's priced a bit higher than aiScaler. However, we do see that there is a place for the free Varnish: It is for sysadmins who have plenty of time, no money, and that are looking for a rather limited HTTP-only caching solution.