SolrPerformanceData

See also: SolrPerformanceFactors, BenchmarkingSolr, SolrPerformanceProblems

Solr Performance Data

Solr users are encouraged to update this page to share any information they can about how they use Solr and what kind of performance they have observed.

Pleae try to give as many specifics as you can regarding:

- The Hardware and OS you used
- The version of Solr you used
- The Servlet Container and JVM you used
- Your index
- The types of operations you tested (ie: updates, commits, optimizes, searchers – the RequestHandler used, etc...)
- What's your greatest performance bottleneck: CPU? Disk speed? RAM?

See also: SolrPerformanceFactors

See also: Lucene's benchmark page and this page on hardware considerations from Summa (which is also based on Lucene)

JobHits job search engine

JobHits is a Solr powered search engine for jobs since the start on May 2009. JobHits has 3 localized version for UK, US and Canada. Each website run on a 2.0 GHz Dual Quad Core dedicate server with 16GB RAM. The server is used for continuously crawling new data, indexing and for the public job search site.

New jobs are being added to the index every 1 minute at the rate of more than 40,000 new documents per day. On the highest loaded site JobHits.co.uk, there are about 2,1 million queries to Solr daily at the rate of 24.235846 requests per second and the average query time is only 34 ms. Below are one sample statistic of the standard search handler component:

```
description:   Search using components: org.apache.solr.handler.component.CollapseComponent,org.apache.solr.
handler.component.FacetComponent,org.apache.solr.handler.component.HighlightComponent,
stats:         handlerStart : 1326769855670
requests :     925736
errors :       11
timeouts :     0
totalTime :    31520551
avgTimePerRequest : 34.04918
avgRequestsPerSecond : 24.235846
```

CNET Shopper.com

The numbers below are from testing done by CNET prior to launching a Solr powered Shopper.com search page. Shopper.com uses a modified version of the DisMaxRequestHandler which also does some faceted searching to pick categories for the page navigation options. On a typical request, the handler fetches the DocSets for 1500-2000 queries and intersects each with the DocSet for the main search results.

The plugin itself uses configuration nearly identical to the DisMaxRequestHandler. To give you an idea of the types of queries that it generates:

- The qf param is used to search across 10-15 fields with various boosts.
- The pf param is used to phrase search across 10-15 fields with various boosts.
- The bq param contains a fairly complex BooleanQuery containing ~20 terms
- The bf param contains two separate boosting functions, one of which contains two nested functions.
- The fq param is used to filter out ~15% of the records that we don't want to ever surface.

The index used in these tests contained ~400K records, took up ~900MB of disk, and was fully optimized.

During the tests, a cron job forcibly triggered a commit (even though the index hadn't changed) every 15 minutes to force a new searcher to be opened and autowarmed while the queries were being processed.

Solr was running on a 2.4GHz dual opteron (DL385) w/ 16GB memory Linux (2.6.9) using Resin 3.0.??. (I don't know the specific resin release or JVM options used)

Each remote client queried the server continously using randomly selected input from a dictionary built using live log files.

```
Number of Concurrent Clients:   1       2       4       6

    Throughput (queries/sec):  33.9    49.2    58.2    60.1
    Avg Response Time (secs):   0.030   0.041   0.069   0.100

    99.9th percentile (secs):   0.456   0.695   1.015   1.418
      99th percentile (secs):   0.245   0.301   0.496   0.661
      98th percentile (secs):   0.173   0.225   0.367   0.486
      95th percentile (secs):   0.095   0.124   0.220   0.323
      75th percentile (secs):   0.027   0.040   0.072   0.108
      50th percentile (secs):   0.017   0.024   0.042   0.063
```

Mailing list post "Two Solr Announcements: CNET Product Search and DisMax" describes a little more about Solr and CNET.

Netflix

Walter Underwood reports that Netflix's site search switched to being powered by Solr the week of 9/17/07:

- Here at Netflix, we switched over our site search to Solr two weeks ago. We've seen zero problems with the server. We average 1.2 million queries/day on a 250K item index. We're running four Solr servers with simple round-robin HTTP load-sharing. This is all on 1.1. I've been too busy tuning to upgrade.

(See http://www.nabble.com/forum/ViewPost.jtp?post=13009485&framed=y)

Walter also reported some figures from their testing phase:

- We are searching a much smaller collection, about 250K docs, with great success. We see 80 queries/sec on each of four servers, and response times under 100ms. Each query searches against seven fields.

At least for these test figures, they were not using fuzzy search, facets, or highlighting.

(See http://www.nabble.com/forum/ViewPost.jtp?post=12906462&framed=y)

Discogs.com

Solr powers keyword search on Discogs.com. From the email archive (alternate copy on nabble)...

```
I've been using Solr for keyword search on Discogs.com for a few
months with great results.

As of today Solr is running under Tomcat on a single dedicated box.
It's a 2.66Ghz P4, with 1 gig ram. The index has about 1.2 million
documents and is 1.2 gigs in size. This machine handles 250,000
queries per day with no problem. CPU load stays around 0.15 most of
the time.
```

HathiTrust Large Scale Solr Benchmarking

HathiTrust *makes the digitized collections of some of the nation's great research libraries available for all.* We currently have slightly over 5 million full-text books indexed. Our production index is spread across 10 shards on 4 machines. With a total index size of over 2 Terabytes, our biggest bottleneck is disk I /O. We did reduce that significantly using CommonGrams, but disk I/O is still the bottleneck for performance.

On our production index, the average Solr response time is around 200 ms, median response time 90 ms, 90th percentile about 450 ms, and 99th percentile about 1.4 seconds. Details on the hardware are available at New hardware for searching 5 million plus volumes Some details on performance are available at: Performance at 5 million volumes. Background and updates available at:The HathiTrust Large Scale Search blog

Zvents

Zvents serves more than 8 millions users monthly with engaging local content. We've used Solr for several years and have achieved very high performance and reliability. User queries are served by a cluster of 8 machines, each having 16Gigs of memory and 4 cores. Our search index contains over 4 million documents. An average week day sees a maximum 80qps with an average latency of 40ms. Leading up to New Years, we'll see ten times this level. To support huge fluctuations in our capacity needs, we run a nightly load test against a single production class machine. The load test itself uses JMeter, a copy of production access logs, and a copy of the production index. The load testing machine is subjected to 130qps and delivers an average latency of 150ms.

Danish Web Archive

Netarkivet is the national Danish Web Archive with 500TB+ harvested web resources. We are using Tika to index this in Solr 4.8 in shards of 900GB / 300M documents. A single 24 core 256GB CentOS machine builds the shards, which takes about 8 days each. Nearly all the CPU power is used on the Tika processes, with the Solr indexer easily keeping up. Each shard is optimized down to a single segment. The machine has ~170GB free memory and would likely work just as well with a total memory of 128GB or less. The Solr indexer has 32GB heap, needed for the final optimization step. Currently (2014-09-19) there are 17 finished shards for a total of 15TB / 5 billion documents.

Search is handled by a single dedicated 16 core CentOS machine with 256GB RAM (currently ~130GB free for disk cache). Each shard has its own Solr instance running in Tomcat with a Xmx of 9GB and resides on a dedicated SSD (Samsung 840). All instances are in a single SolrCloud. There are 25 SSDs in the machine and it is currently undecided if the setup will be scaled up (more RAM, more SSDs) or out (another similar machine), when we reach that number of shards.

Access to the archive is limited and at most 2 or 3 users are active at a time. Searches are faceted on 6 fields: URL (nearly 5 billion unique values), domain, host and 3 smaller ones. With Solr 4.8 and SOLR-5894, median and average response times during load testing are currently below ½ second and expected to stay below 1 second as the index grows. Special searches, such as *:*, take up to 2 minutes. For non-faceted searches, IOWait get as high as 10%. For faceted searches, IOWait stays below 0.5% and CPU-load is high. It has not been determined if the high CPU-load is due to processing (easily scalable) or memory access congestion (not easily scalable). See Even sparse faceting is limited for most recent performance figures.