BenchmarkingSolr

See also: SolrPerformanceFactors, SolrPerformanceData, SolrPerformanceProblems

Benchmarking Solr

Perl script (originally by ShawnHeisey)

It's not very polished. It will not work on Windows as written, not sure if fork() will work right on Windows anyway. Requires the URI::Escape, Time::HiRes, Statistics::Descriptive, IO::Handle, and LWP::Simple Perl modules.

The first part of the script has a number of options that you will probably need to change for your environment. From what gets output, the median, 95th percentile, and 99th percentile values seem to be the most useful.

During a recent online Solr training session (July 15th, 2011), this script was mentioned by Tom Hill and classified in the "not very useful" category. I was amused. 😉

{{{#!ignoreme
#!/usr/bin/perl

#--- start config —
my $forkCount = 8; my $queryCount = 512; my $outputDir = "/tmp/zot"; my $querySource = "/usr/local/etc/queries"; my $urlHost = "localhost"; my $urlPort = "8983"; my $urlCore = "live/"; # set to "" to not use a core my $urlOptions = "rows=50&fl=score,*"; my $uriEscape = 1; # Enable if queries are not already URI escaped my $writeResponses = 0; # Enable to write responses to disk
#---- end config ----

use strict; use URI::Escape; use warnings qw(all); use Time::HiRes qw (time alarm sleep); use Statistics::Descriptive; use IO::Handle; use LWP::Simple;

my %kids; my $pid; my $i; my $j = 0; my $r; my $query; my @queries; my $size; my $num; my $veryStart = time(); my $start; my $elapsed; my $stat; my %cfg;

my $urlTemplate = "http://HOST:PORT/solr/COREselect/?q=QUERY&OPTIONS"; my $url;

###
Main routine
###

mkdir $outputDir; mkdir "$outputDir/result" if $writeResponses;
The length test ensures that we don't clobber the root filesystem. system "rm -f $outputDir/* 2> /dev/null" if length $outputDir; system "rm -f $outputDir/result/* 2> /dev/null" if length $outputDir
and $writeResponses;

$urlTemplate =~ s/HOST/$urlHost/;
$urlTemplate =~ s/PORT/$urlPort/;
$urlTemplate =~ s/CORE/$urlCore/;
$urlTemplate =~ s/OPTIONS/$urlOptions/;

for ($i = 0; $i < $forkCount; $i++) {
$pid = fork();
if ($pid) {
$kids{$pid} = 1;
} else {
@queries = `cat $querySource`;
$size = @queries;

open OUT, ">$outputDir/$$.zot";
OUT->autoflush(1);
for ($i = 0; $i < $queryCount; $i++) {
$num = int(rand $size);
$query = $queries[$num];
chomp $query;

$query =~ s/ /+/g; # use + for space
$query = uri_escape($query) if $uriEscape;

$url = $urlTemplate;
$url =~ s/QUERY/$query/;

```
$start = time();
$r = get ($url);
if (defined $r and $r) {
$elapsed = time() - $start;
print OUT "$elapsed\n";
if ($writeResponses) {
open R, ">$outputDir/result/$$.$j";
print R $r;
close R;
$j++;
}
}
}
close OUT;
exit 0;
}
}

foreach (keys %kids) {
waitpid($_, 0);
}

$stat = Statistics::Descriptive::Full->new(); foreach $i (keys %kids) {
open IN, "<$outputDir/$i.zot";
while (<IN>) {
chomp;
$stat->add_data($_);
}
close IN;
}

system "rm -f $outputDir/* 2> /dev/null" if length $outputDir; system "rm -f $outputDir/result/* 2> /dev/null" if length $outputDir
and $writeResponses;

printf " Req/s: %1.03f (%1.03f sec, requests %d/%d)\n"
, $stat->count() / (time() - $veryStart)
, time() - $veryStart, $stat->count(), $forkCount * $queryCount; printf " Avg: %1.03f\n", $stat->mean(); printf "Median: %1.03f\n", $stat->median(); printf "
95th: %1.03f\n", $stat->percentile(95); printf " 99th: %1.03f\n", $stat->percentile(99); printf " Max: %1.03f\n", $stat->max(); print "\n";

exit 0;
}}}
```

Sample output:

```
 Req/s: 139.682 (29.302 sec, requests 4093/4096)
   Avg: 0.055
Median: 0.046
  95th: 0.119
  99th: 0.210
   Max: 1.180
```

SolrMeter

NOTE: SolrMeter is still available to download, but It's currently not under development

- SolrMeter is a stress testing / performance benchmarking tool for Apache Solr installations. It is licensed under ASL and developed using JavaSE and Swing components, connected with Solr using SolrJ. The main goal of this open source project is bring to the Apache Solr user community a tool for dealing with Solr specific issues regarding performance and stress testing like executing queries and adding documents to make sure that your Solr installation will support real world's load and demands. With SolrMeter you can simulate a work load over the Apache Solr installation and to obtain useful visual performance statistics and metrics.
- Check the project site: http://code.google.com/p/solrmeter
- Sample View:
- `http://solrmeter.googlecode.com/svn/wiki/Screenshots/pie-chart1.PNG`
  .