

Dude, please try to be accurate.

Posted At : January 29, 2011 10:38 PM | Posted By : Jared
 Related Categories: Tech Aggregators, Adobe ColdFusion

People who skew the truth and deliberately mislead people make me angry. This post is a direct result of that phenomenon. If I sound irritated, well, it's only because I am. People who make a deliberate attempt to denigrate **any** platform based on artificial or manipulated results **need** to be called out on it. Letting a lie grow is, in my opinion, the same as telling it yourself...

In a **recent post on his blog**, Mr. Phil Parsons, self-professed CF "disliker" ("hater" seems a bit strong lol) and PHP aficionado, released several bits of code that claimed to demonstrate ColdFusion's performance inferiority to PHP. So I decided to see what was up with that myself. I wanted to see if it was true. I would have been happy to report that, yes, PHP is in fact faster than ColdFusion, here's my results. However the more I experimented and optimized both the PHP version and the ColdFusion version, the angrier I got about Mr. Parsons' blatant misrepresentation of the facts.

(NOTE: I didn't want to disprove him, I wanted to find out if it was true. If it's true, so what? This was a thought exercise, not an attempt to redeem ColdFusion (which really doesn't need redeeming anyway!))

Much thanks to Elliot Sprehn for helping me get a decently optimized prime number scanner that's practically bit-for-bit identical between PHP and ColdFusion. It's funny, because even his PHP code was pretty poorly written. According to the Mr. Parsons, the best PHP could do to calculate prime numbers up to 10,000 was 128ms. According to our optimized version of the scanner, PHP is capable of calculating prime numbers up to a value of 10,000 in **11.51ms**.

Here's where the results get interesting:

ColdFusion, pathetic, ugly, poorly performant ColdFusion, despised and unpleasant to write with, actually did the same thing in **8.214ms**.

Sorry Mr. Parsons... your "benchmarks" suck. ColdFusion is a full **3 and a quarter milliseconds faster!**

I'm sure this will make or break a preponderance of web applications currently in development. /sarcasm

And incidentally, Mr. Parsons' professed goal with his post was specifically to prove ColdFusion inferior to PHP.

Ooops.

The test platform was:

- **Hardware:** Apple MacBook Pro 2.8 GHz Core II Duo w/8GB RAM
- **Web server:** Apache/2.2.15 (Unix) x64 (OS X Snow Leopard Default)
- **PHP:** PHP 5.3.3 (cli) (OS X Snow Leopard Default)
- **CFML Engine:** Adobe ColdFusion 9.0.1 x64 on JRun 4

And, for the record, none of the 3 server components were optimized in any way. Out-of-the-box and go.

Here's the code I used for each:

PHP:

```

<?php

function & primes($to) {
    $primes = array(2, 3);
    for ($i = 5; $i <= $to; ++$i) {
        $root = sqrt($i);
        for ($j = 0; $primes[$j] <= $root && $i % $primes[$j] != 0; ++$j);
        if ($primes[$j] > $root) {
            array_push($primes, $i);
        }
    }
    return $primes;
}

function arrayAverage($array) {
    return array_sum($array)/count($array);
}

$times = array();

for ($i = 0; $i < 1000; $i++ ) {
    $t = microtime(true);
    $primes = primes(10000);
    array_push($times,microtime(true)-$t);
}

echo number_format(arrayAverage($times) * 1000, 5) . " milliseconds for " . count($times) . " iterations.";

?>

```

Didja notice I had to roll my own average function? WTF?

ColdFusion:

```

<cfscript>
function primes(to) {
    var primes = [2,3];
    for (var i = 5; i <= to; ++i) {
        var root = sqrt(i);
        for (var j = 1; primes[j] <= root && i % primes[j] != 0; ++j);
        if (primes[j] > root) {
            arrayAppend(primes, i);
        }
    }
    return primes;
}

times = [];
for (c=1;c<=1000;c++) {
    start = getTickCount();
    primes(10000);
    arrayAppend(times,getTickCount()-start);
}
writeOutput( arrayAvg(times) & " milliseconds for " & arrayLen(times) & " iterations.");
</cfscript>

```