Randomisation of cache durations reduces peak load variance on origin systems

Matthew Neale

EstarOnline Limited

mneale@estaronline.com

Abstract

Caching of data within computer systems is commonly used to minimise the number of executions of expensive data generation operations. This is useful where response time is more important than the timeliness of the data itself.

In this paper we show how using random variance around a desired cache duration reduces the variability and peak loads incurred on the origin system when compared to using a fixed cache duration.

Introduction

High performance and high scale information retrieval systems all use various forms and levels of caching to insulate the underlying origin systems from high dynamic loads.

A number of strategies are employed for cache expiration, these are typically chosen based on timeliness requirements and employ fixed cache durations to achieve this.

In highly parallel environments, and specifically, high-volume websites this fixed cache duration can result in highly variable loads on origin systems (often database servers) where it is typical for multiple cached data objects to expire at the same time, resulting in a peak loading as data is refreshed from the origin.

This is visibly exhibited by unpredictable and highly variant response times from the origin systems, especially where the resultant peak loads approach or exceed the resources available.

We explore the results of peak load variance on origin systems by introducing various levels of random variance to the desired cached duration.

A highly parallel and high demand load environment is simulated for running tests over a period of time to establish;

i. Whether the overall load on the origin system is equivalent when a fixed cached duration is used vs. where a degree of random variance to the desired cache duration is introduced.

ii. Whether the introduction of random variance to the cache duration reduces the peak loads on the origin systems.

iii. Whether the peak load variance on the origin system is reduced when using random variation on the desired cache duration.

Methods

Testing was carried out using a single isolated system, and involved running a query of fixed duration against a Microsoft SQL Server database (the origin system) from a simulated high usage environment.

The test environment itself involved the continuous running of batches of 100 parallel requests through individual caches to the database query over a fixed period.

This provided a consistent load situation to allow for measurement of the request metrics in isolation from external influences.

The same load condition was set for all tests, the only variable was the amount of random variance applied to the cache duration.

Tests were run for 10 minutes, +/- 15ms (accounting for variability in the time limiter).

Measurements were taken of;

i. Number of requests made to the origin server (database) every second.

ii. Number of total requests serviced (including requests serviced from the cache) every second.

The nominal cache duration was set at 7s.

Cache variances tests were run at 5 levels of random variance; 0% (equating to a fixed cache time), 7.5%, 15%, 20% and 30%

Each test was repeated 3 times to reduce natural variance over a total of 15 individual tests and the median result taken.

Results

The data confirmed the assertion that the overall workload on the origin server across all tests should have been substantially the same, this is evidenced by the data presented in Table 1.

It was also observed that the maximum peak load (MAX ORPS) was identical across all tests – this is consistent with the initial cache load. For those tests with a non-zero degree of random variance on the cache duration, it can be seen in Figures 2 – 5 that this peak load only occurred once during these test runs.

Figure 1 shows that with a fixed cache duration, MAX ORPS is achieved following every cache expiration over the test period.

Table 1 Shows that for cache durations that have some degree of random variance, the VAR ORPS (the Variance of the Origin Requests Per Second) decreases as the degree of random variance increases.
Figures 2 – 5 demonstrate the VAR ORPS reducing at various rates as the tests progress.

Table 1. Statistical summary of requests across all tested cache duration variances.

RPS = (Total) Requests per Second, ORPS = Origin Requests per Second, OR = Origin Requests, R = Requests.

Cache Variance (%)	*0*	*7.5*	*15*	*20*	*30*
AVG RPS	658,276.4365	655,776.4331	649,005.8144	666,509.1472	655,124.6572
AVG ORPS	14.15385	14.33782	15.33445	14.36532	15.42977
VAR ORPS	1186.93	87.21193	46.10554	30.07698	29.29858
MAX ORPS	100	100	100	100	100
Total OR	8464	8531	9170	8533	9227
Total R (Millions)	393.649309	392.154307	388.105477	398.572470	391.764545

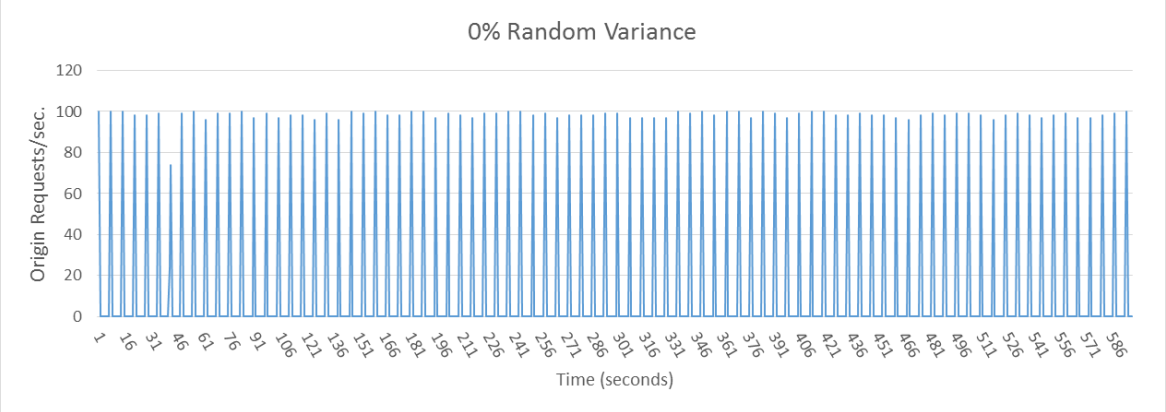

Figure 1. Origin Requests per Second with 0% random variance on desired cache duration. I.e. A fixed cache duration.

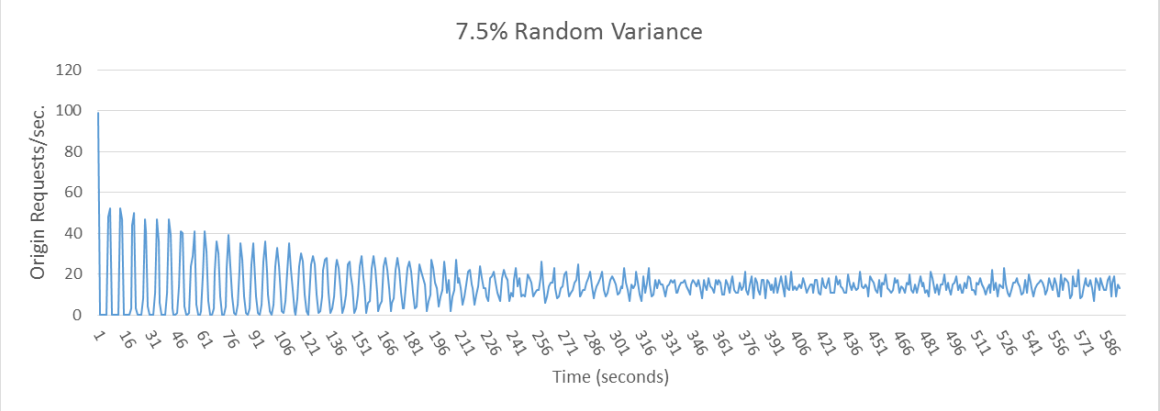

Figure 2. Origin Requests per Second with 7.5% random variance on desired cache duration.

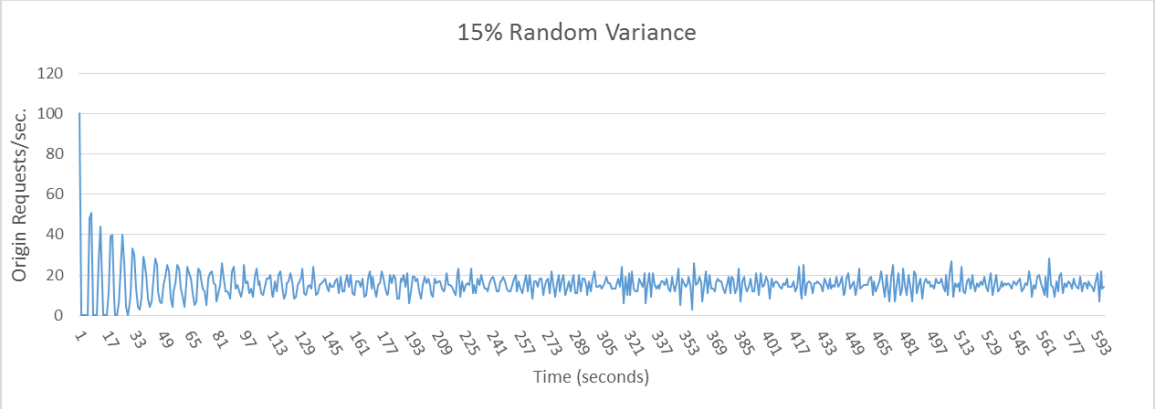

Figure 3. Origin Requests per Second with 15% random variance on desired cache duration.

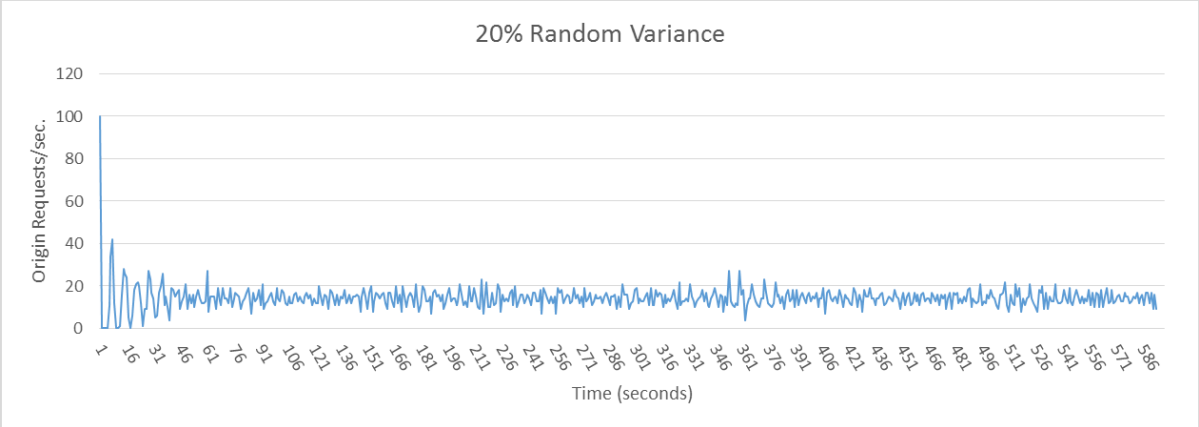

Figure 4. Origin Requests per Second with 20% random variance on desired cache duration.

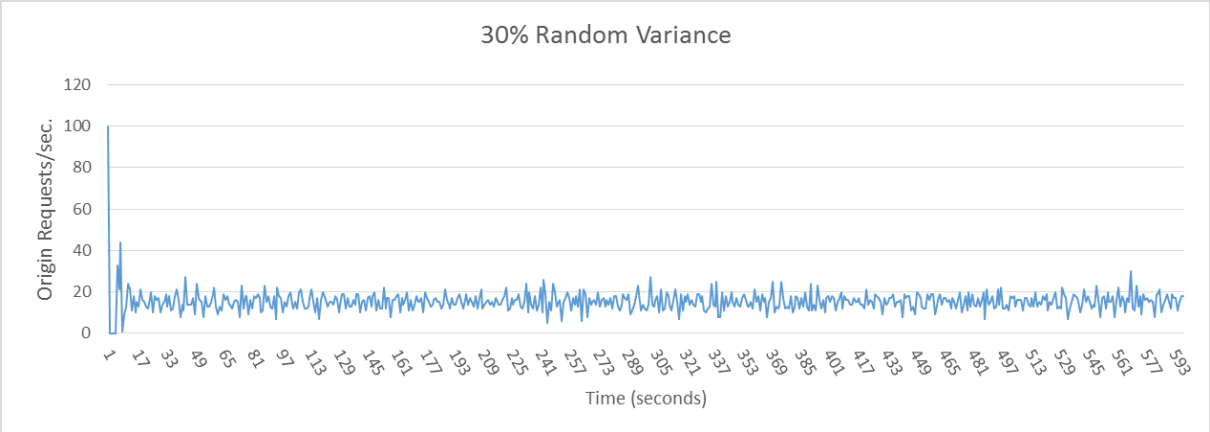

Figure 5. Origin Requests per Second with 30% random variance on desired cache duration.

Conclusions

Introducing random variance to fixed cache durations reduces the variance in peak loading on the origin system versus a fixed cache duration with no variance.

This variance in cache duration essentially spreads the cache stagnation and refresh load on the origin server such that a constant level of load is achieved, whilst maintaining the same overall workload and throughput.

We have established that;

i. The overall load on the origin server is equivalent over a given period when using either a fixed cache duration or an equivalent randomly variant cache duration.

ii. The peak loading on the origin server is unchanged when using a variant cache duration (due to the initial un-cached load).

iii. The variance in the peak load on the origin system is significantly reduced when using a randomly variant cache duration.

iv. The higher the amount of random variance in cache duration, the faster the system settles to a constant load condition.

It would be worth running these tests using an origin server operating with a baseline load at close to its' maximum capacity. It would be expected, based on the results presented here, that using a cache duration with random variance will attain greater overall throughput by the elimination of repeated peak loads that exceed the origin systems maximum capacity.