Basho Bench Performance Testing Results

문서 개정 이력표

문서명		Performance Testing Results	
버전	**날짜**	**내용**	**작성자**
0.1	2012. 03. 23	최초 작성	Rune Skou Larsen (rsl@trifork.com)

목 차

1. Introduction

This is the results of the Basho Bench performance tests run on March 23rd, 2012.

1.1 Setup

1.1.1 VMs and Hardware

The physical hardware used in the test are two virtual hosts with the following specification:

- 4 x 7200 RPM, 2TB harddrives (no raid)
- 24GB RAM
- 8 core Intel Xeon X5670

These two hosts each host 4 VM's (8 VMs in total) with the following specs:

- 6GB RAM
- 4 CPU cores
- 1 harddrive mapped from the host.

1.2 Test strategy

The purpose of the tests is to compare read performance of *bitcask*, *leveldb* and *leveldb with secondary indeces* under different conditions.

Before the tests, bitcask and leveldb were loaded with 10 million with data size randomly chosen between 1000 bytes and 2000 bytes.

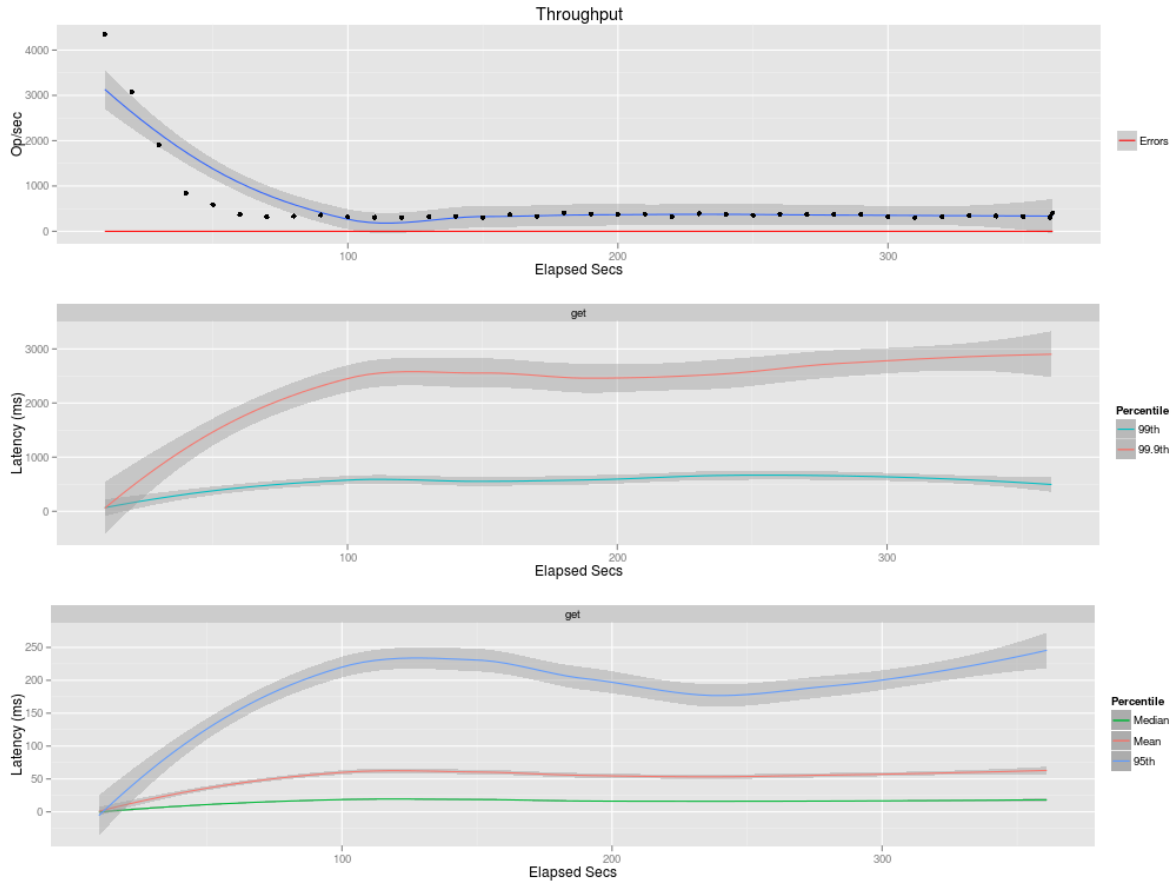
Tests were done with 20 threads in the following scenarios:

- Random reads
- Pareto distribution (20% keys requested 80% of the time)
- Partitioned sequential read, where each threads is assigned a range in the key-space.

2. Bitcask tests

2.1 Bitcask Random read

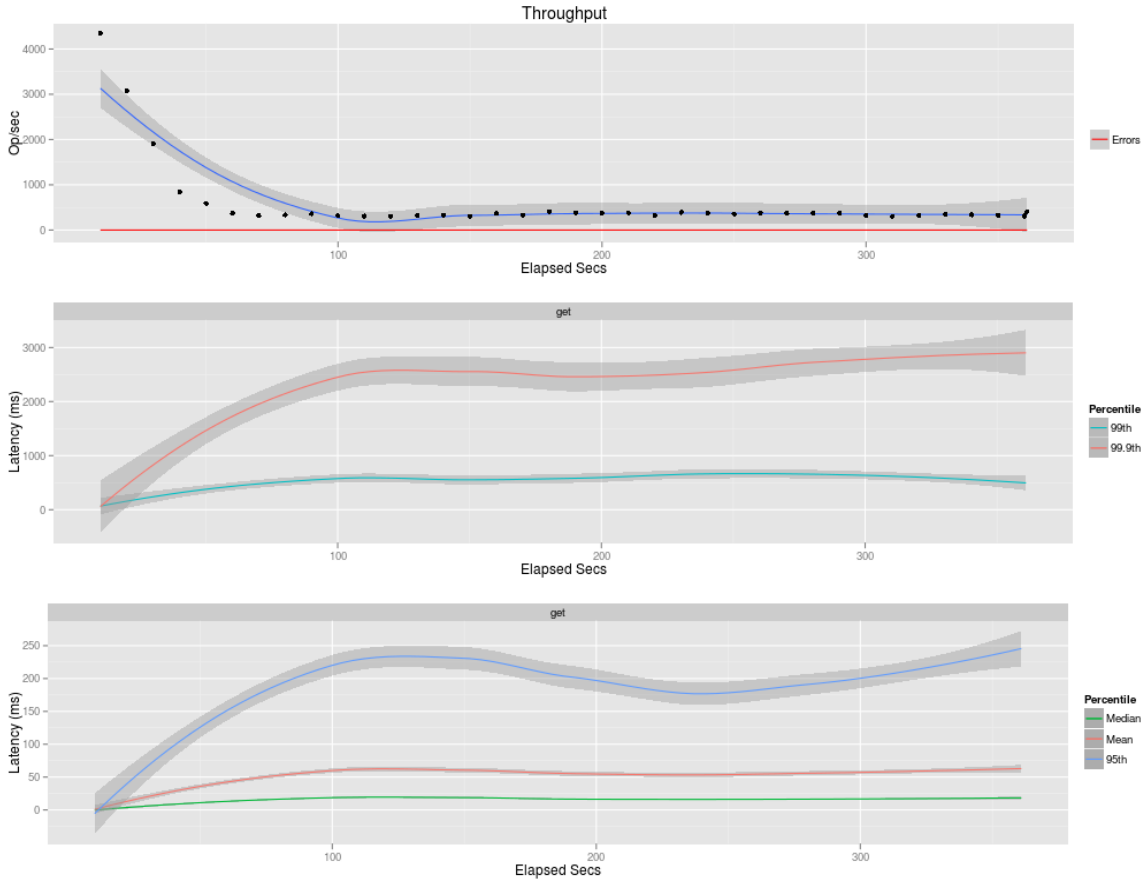
20 Threads were used.



After initially starting out high, throughput stabilizes at about 300 ops/sec. 95th percentile express similar characteristics – starts low, but then rises to about 200ms.

2.2 Bitcask Pareto distribution

20 threads



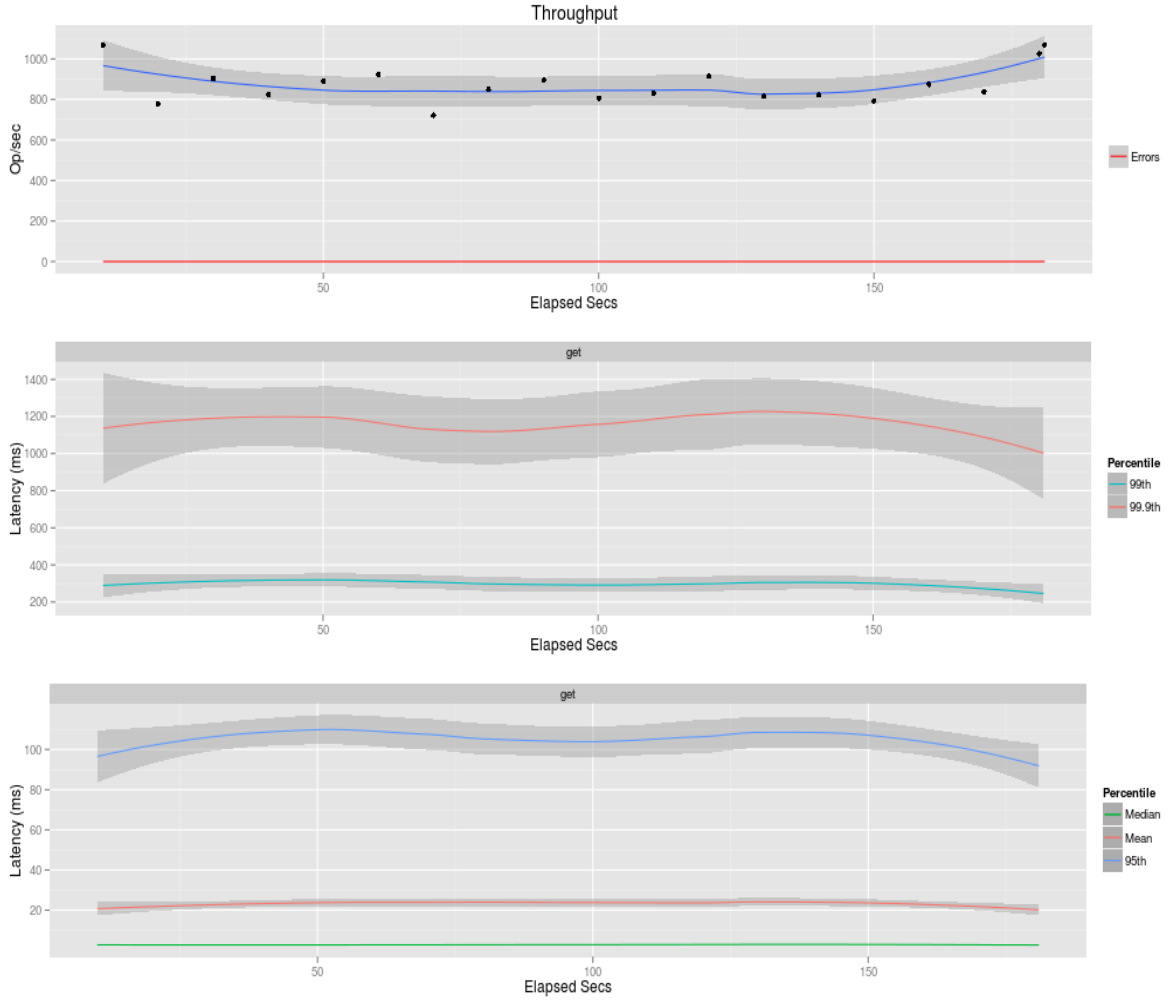
Using the Pareto distribution would expect caching to be more of a factor than random reads. This seems also to be the case, as throughput stabilizes at about 400 ops/sec.

The same start-high behaviour as seen in the random test seems to be the case here. 95th percentile is at about 200ms like in the random test.

The initial high performance is unexplained and might indicate something is wrong – for instance with the file system caching.

2.3 Bitcask partitioned sequential

Each of the 20 threads was given a key-range to search through sequentially.



Throughtput is stable at approx. 850 ops/sec. 95th percentile is about 100ms.

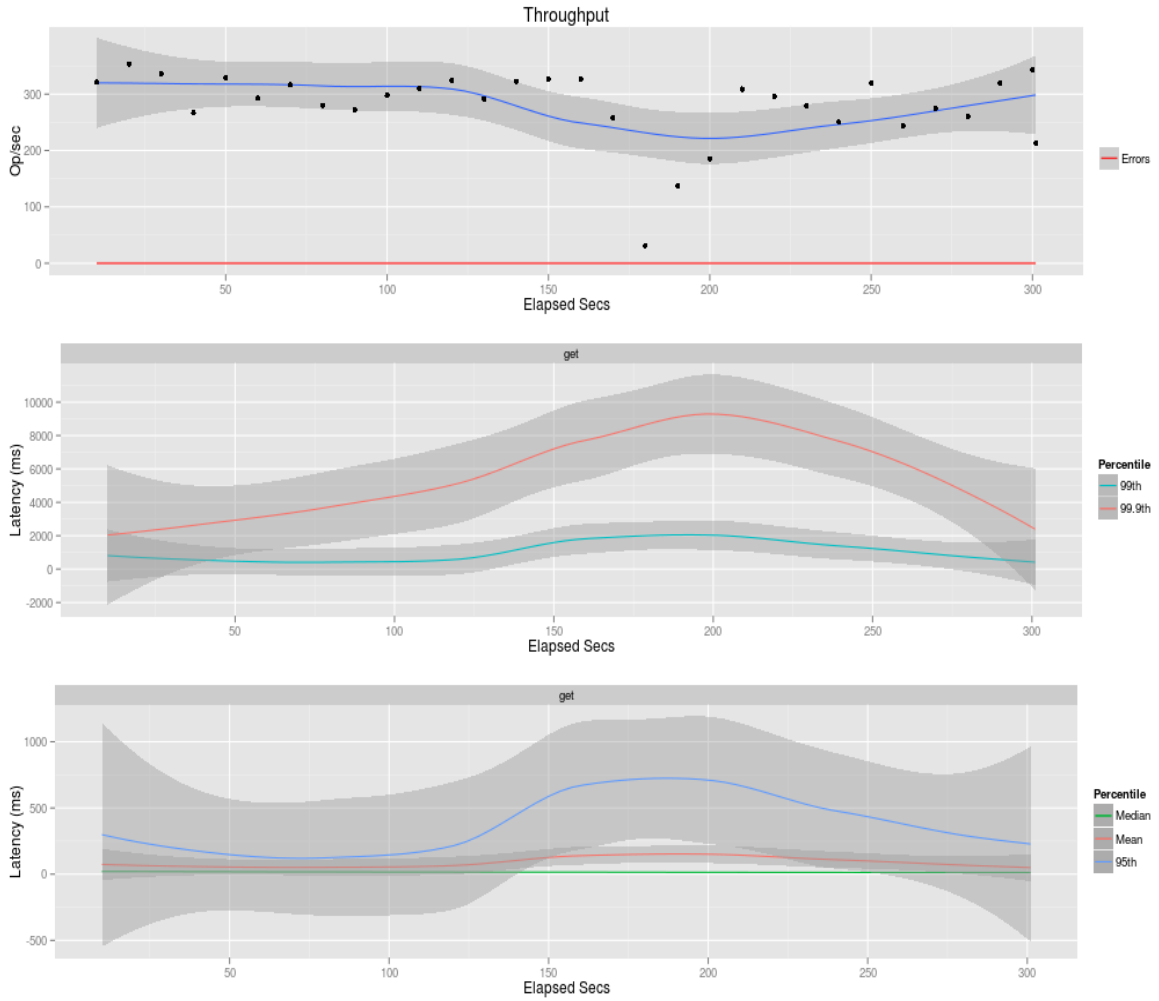
This is faster than both the random and pareto tests. Even though bitcask does not order its data, this higher performance from sequential reading is to be expected, because data was loaded by the workers in an ordered fassion.

3. LevelDB tests

Leveldb tests were run on uninitialized data, so all responses were 404's.

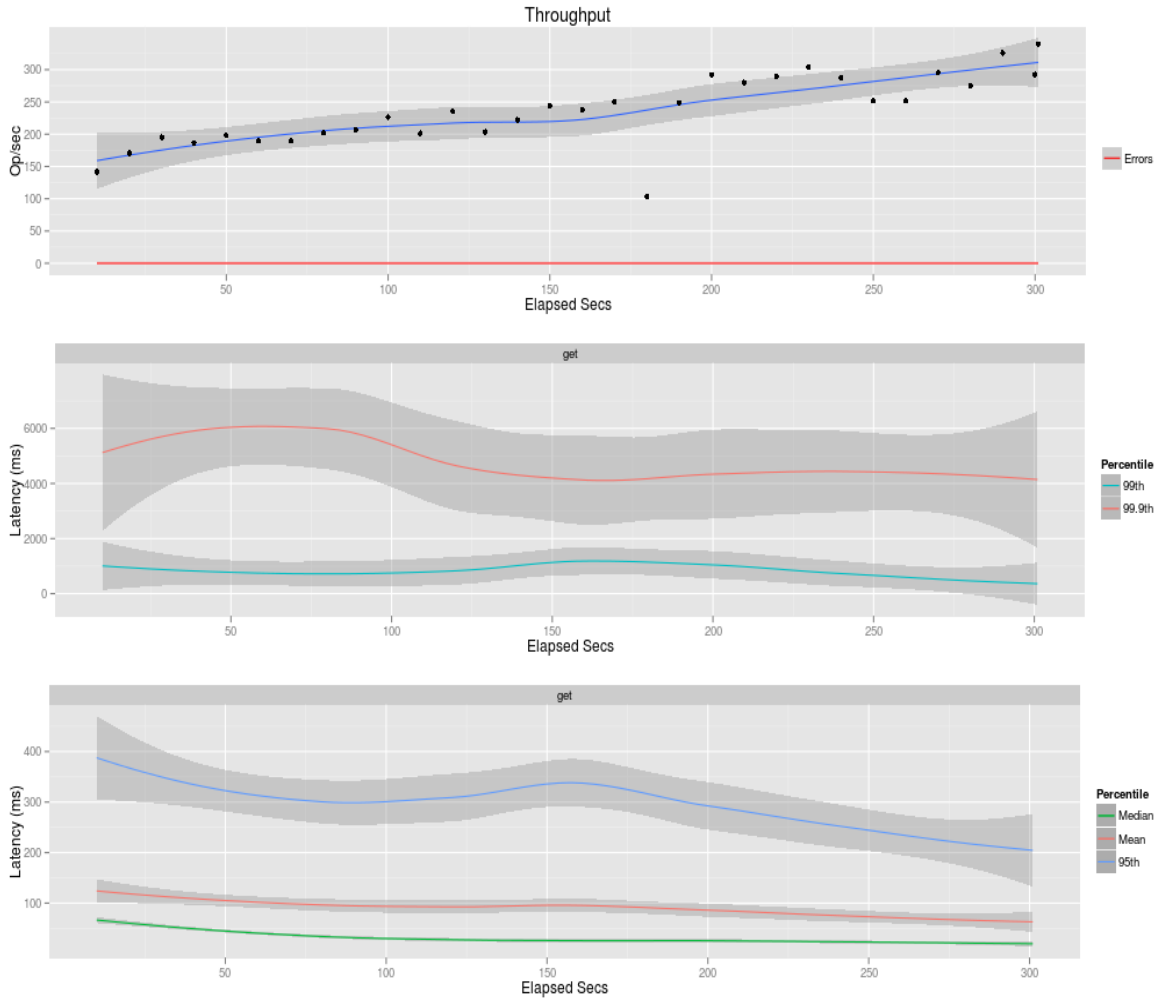
3.1 Leveldb random read

Random reads, 20 threads



Performance drop from 170-210s, that might be worth investigating further.

3.2 Leveldb Pareto distribution reads



Steady performance increase during the test, probably due to increasing cache-hit rates, when the caches learn which data is accessed most frequently.