MongoDB performance evaluation on different instance types

1. Experimental Setup

We use three different types of instance here to make comparisons,

Instance types:

t2.micro: 1vCPU, 1GB memory

t2.small: 1vCPU, 2GB memory

t3.micro: 2vCPU, 1GB memory

We mainly use YCSB to test the MongoDB performance. However, at first we need to install MongoDB and support java environment.

Install MongoDB

Download

$ wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.6.12.tgz

Unzip

$ tar -zxvf mongodb-linux-x86_64-3.6.12.tgz

$ cd mongodb-linux-x86_64-3.6.12

Install JDK

$ sudo yum install java-devel

(The version of jdk should be higher than 1.8.0)

Install Maven

$ sudo yum install maven

Install YCSB

Download YCSB MongoDB

$ wget https://github.com/brianfrankcooper/YCSB/releases/download/0.15.0/ycsb-

mongodb-binding-0.15.0.tar.gz

Unzip

$ tar -zxvf ycsb-mongodb-binding-0.15.0.tar.gz

We set up the operarioncount and recordcount to be 100k in workloads directory and use the following commands to test.

$ bin/ycsb.sh load basic -P workloads/workloada(or workloadb)

$ bin/ycsb.sh run basic -P workloads/workloada(or workloadb)

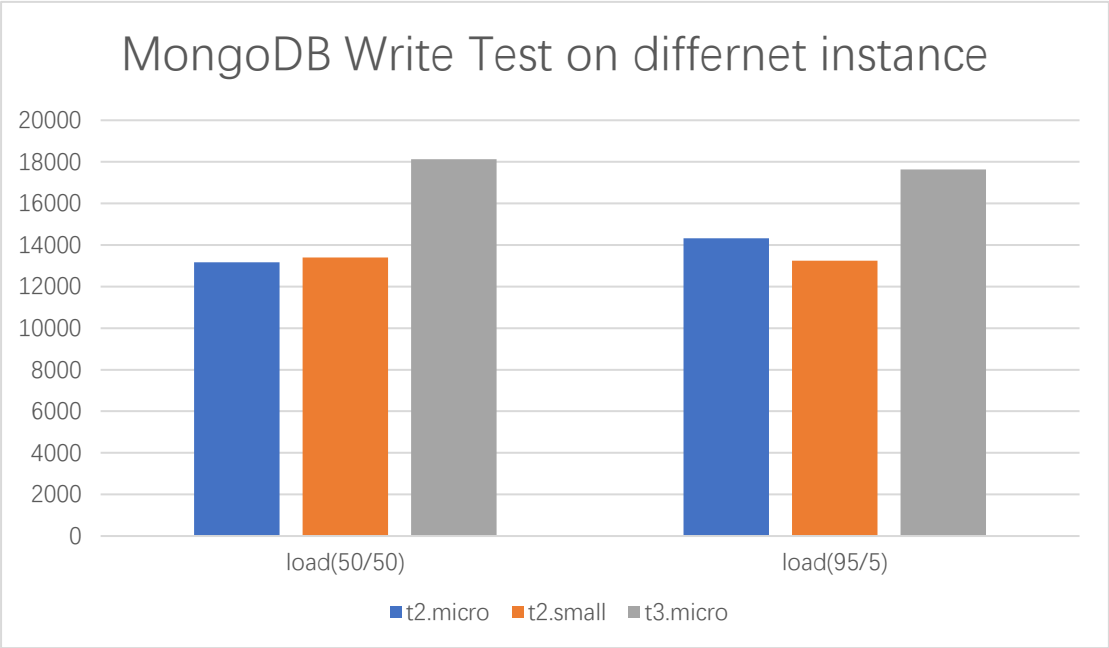
t2.micro(1vCPU 1G)	load	run
workloada(50/50)	13161	20283
workloadb(95/5)	14316	26178

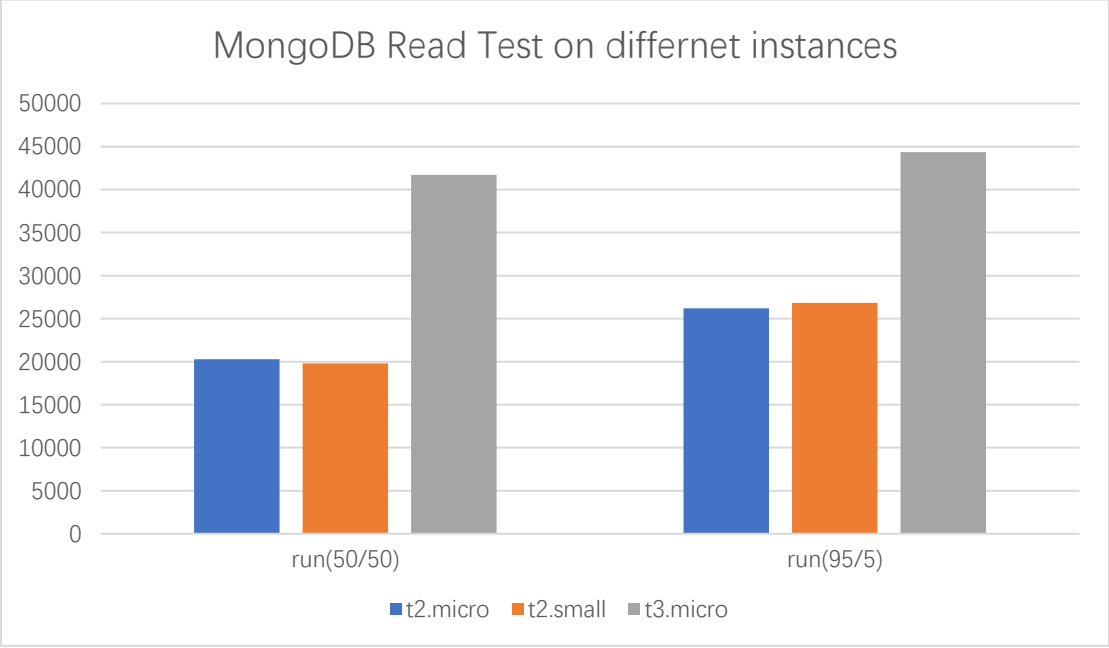
t2.small(1vCPU 2G)	load	run
workloada(50/50)	13397	19805
workloadb(95/5)	13250	26831

t3.micro(1vCPU 2G)	load	run
workloada(50/50)	18131	41701
workloadb(95/5)	17633	44345

workloada: 50%/50% Reads/Writes

workloadb: 95%/5% Reads/Writes





2. Discussion of the Result

From the result above, serveral points can be summarized:

1. The write speed of MangoDB on the three different types of instances is almost the same.

And the t3.micro performs a little better. We have known about the difference between the three instances. Thus, we could conclude that the more CPUs the instance has, the MongoDB could perform a little better on the write speed. At the same time, we could have that the memory size makes no contribution on the write speed.

2. As to the read speed of MangoDB on the three different types of instances, the t3.micro performs obviously much better. Meanwhile, the t2.micro and the t2.small perform the same. Thus, we could conclude that the number of CPUs has a significant effect on the read speed. The more CPUs it has, the faster the read speed is. Also, we could have that the memory size makes no contribution on the read speed.

According to the above situation, MongoDB has a relatively high requirement for the number of CPUs, which I think will cause a large load on the CPUs, so it needs good CPUs to support it. It could be a bottleneck of the system since it costs much.

Also, If your database content is small or not a massive amount, the read speed requirements are not large, then you can use less core CPU, like t2.micro. If the content is large and require the read speed to be high, it is recommended to use more CPU cores and prepare more memory space, like t3.micro or a better one.