System Design for Google Maps

Note: This is just a brainstorming article.
It could be a complete mismatch with what actually goes on in Google Maps.
This only shows how one can logically reason and try to draw some estimates about a big cloud service.

Google maps shows images of earth and allows zooming in and out of these areas.
So a good starting point is to know how much area does Google maps cover and how much storage would it need to cover all the area.
Earth has an area of around 500 million km^2.

Removing 70% water and other non-interesting area, we are left with 10% area that should be actually mapped ~ 50 million km^2= 5x10^7 km^2

Lets say we have one image for each 10mx10m block, and size of each such image is 1 MB
Assuming 2 road-names and 1 building-name per image: 300 bytes per image
We can ignore such contextual information in regards to the actual image size of 1 MB
We will have 100x100 such blocks per km^2, so 10,000 MB of images per km^2
For 5x10^7 km^2, we will need 5 * 10^{11} MB space which is ~ 10^{12} MB = 1000 PB

We also need to store zoom levels.
Assume a factor of 3.3 when zooming up or down, we need to calculate image-data at following scales:

```
a.  1000 PB/3.3 = 330 PB

b.  330 PB/3.3  = 100 PB

c.  100 PB/3.3  =  33 PB

d.  33 PB/3.3   =  10 PB
```

Adding these, we get an upper bound of **1500 PB**
If you think this is crazy and probably there is a mistake in the calculations, then you may be right but 100s of petabytes of data is not uncommon these days.
For example, Yahoo! revealed in 2014 that it stores around 500 PB of data in over 40,000 servers!
Also read this interesting article about how a startup Backblaze bought hard-drives when their customers were consuming 50 TBs each day.

So we assume that 1500 PB of data is not a bizarre number and we can go ahead with our analysis.

Number of storage nodes required

Assuming a single machine could manage around 15 TB of space, we would need 1500 PB/ 15 TB = 100,000 database shards.
We would need some replicas with each shard to handle read traffic and to add fail-tolerance.
Assuming 1 master 2 replicas per shard system, we would need 300,000 database nodes, each holding 15 TB of space.
BTW, 15 TB of space per machine may not be too costly these days.
Also see how Backblaze got some really cheap storage.

Managing 300,000 database nodes could become a nightmare.
So instead of just relying on horizontal scaling, we could use some vertical scaling as well.
Let us see theoretically how much memory is addressable by a 64-bit machine.

2^{32} = 4 GB ($4*10^9$ bytes), therefore
$2^{64} = (4*10^9)^2 = 16*10^{18}$bytes = $16*10^6$ TBs = 16000 PBs = 16 EBs (Exabytes)

Thus a 64 bit system can theoretically address 16 EBs of space and can theoretically host the entire 1500 PBs of data in a single machine!
But in practice, systems with such huge memory become exorbitantly expensive.
So we don't know how to avoid this nightmare of 300,000 servers and to continue our analysis, we would assume that so many servers are actually being used.

Traffic analysis for Google Maps

1. 500 million monthly google map users
2. Assuming each user requires 30 minutes of navigation guidance per month, we require 1.5 billion minutes
3. How many images are required per minute depends on the speed of travel, which we assume to be an average of 30 km/hour
4. 30km/hour will need 30MB of data per hour (because we had 1 MB per km^2) which is 0.5 MB / minute
5. So for 1.5 billion minutes, we need $1.5 \times 10^9 \times 0.5$ = 0.75 biliion MB in the whole month = 750 TB traffic/month
6. A month has roughly 2.5 million seconds, so we need to support 750 TB/$2.5 *10^6$ = 750 MB / 2.5 = 300 MB / second traffic

7. Number of requests: As seen in #4 above, 30 requests per hour will be there from each user. This translates to:
   30 x 500 million / 2.5 *10^6 = 1500 x 10^6 / 2.5 x 10^6 = 300 / 0.5 = 600 requests per second

8. So traffic itself doesn't seem too much of a problem here. Main bottleneck seems to be the data itself.
   Our system should be able to quickly locate the image position and return the same within a few milliseconds.

Maps are from satellites, but where does real-time traffic come from?

Governmental transportation departments install solar-powered traffic sensors on major roadways to gather planning statistics, improve accident response times and increase traffic flow. Google can partner with these departments to share the costs of sensors while getting a share of the traffic information.

For smaller and remote areas, devices running the maps application provide the actual speed the device is traveling to measure speed.

More at this link