3/2/2011

Advanced Topics in Distributed Systems

EXAM Problem A

The servers in data center network are connected using a 1gb/s link to a Top of Rack switch. Each ToR switch has L 1Gb/s uplinks, each connecting to one of the L aggregate switches. Each aggregate switch is connected to all the ToR switches.

1. If all switches have 48 ports, and L is 24, what is the maximum number of servers this topology can support? (hint: draw the topology to help reasoning about its properties) (1p)

Raspuns: Avem 24 de switch-uri la nivelul cel mai de sus; acestea au fiecare 48 de porturi, deci pot fi conectate fiecare la 48 ToR switches; deci avem 48 rack-uri. In fiecare rack putem avem maxim 24 servere (pentru ca celelalte 24 de porturi ale ToR-ului sunt conectate in sus). Deci in total avem 24 * 48 servere = 1152

2. Assume 48 port switches and L=24. Each server is identified by rack ID and rack position as (rackID,rackPos). Assume there are T racks in total, and consider the following traffic matrix: server (i, j) sends to server ((i+1) % T, j); such that every server has one outgoing TCP connection and one incoming TCP connection. What is the maximum throughput each connection can achieve? (2p)

Raspuns: 1Gb/s ; topologia este full bisection

3. How should connections be placed on paths to achieve this throughput? Give an example of placement. (2p)

Raspuns: desenati doua rack-uri asociati fiecare conexiune TCP cu o cale libera din topologie.

An application runs in the data center described above, with 48 port switches. The application needs to replicate large files (10GB each) on a configurable number of servers R.

4. Say R is two. Which servers should store file replicas to ensure that any single failure in the system does not affect the availability of the files? 1.5(p)

Raspuns: in servere din rack-uri diferite pentru a evita probleme atunci un ToR switch moare.

5. How should the replicas be written to the servers to minimize the time needed to store the files? Say a file needs to be replicated on ALL the servers in the data center. How long will this take, asumming L=24? (2p)

Raspuns: daca folosim daisy chaining fisierul poate fi scris pe toate serverel la viteza de 1Gb/s; timpul este de 80Gb / 1Gb (s) = 80s.

6. What is the minimum L that also minimizes the file store time for a file stored on ALL the servers? (1.5p)

Raspuns: L = 1 este de ajuns, pentru ca ne trebuie un singur transfer inter-rack de la fiecare rack la rack-ul urmator; toate celelalte replici se scriu pe urmatorul server in rack

3/2/2011

Advanced Topics in Distributed Systems

EXAM Problem B

The execution trace included below shows Apache HADOOP, an implementation of Map Reduce, running on 12 servers connected to a single switch. Each server has a single hard disk drive, and is configured to run concurrently at most three maps and two reduce tasks. The input data is stored on the same nodes using HDFS (a variant of GFS) at replication level 2. The input data is a set of web pages crawled from the UK domain; its size is approximately 50GB (unreplicated).

Hadoop runs the WordCount app that computes for every word in the input files, the total number of occurrences in all the files. The output of the job is a list of (word,count) pairs that is written to HDFS.

The output shows CPU usage (every point is an average over 1s), disk usage (averaged over 1s) in blocks/s (1 block=1024B, IN=read, OUT=write), and network usage (averaged over 1s). Each map reads a single block (128MB). The system is configured to use 48 reduce tasks.

Map progress is given by the amount of input data read out of the total data. Reduce progress has three factors: 1/3 corresponds to the shuffle phase, 1/3 to the sorting of the keys, and 1/3 to running reduce and writing the outputs.

Questions:

Describe what happens during the Map phase. (4p)

1. Why are there phases in CPU usage?

Raspuns: valuri de taskuri map sunt pornite si se termina in acelasi timp

2. Why are the phases less distinct towards the end of the map phase?

Raspuns: taskuri map se desincronizeaza catre sfarsitul job-ului din cauza contentiei pentru I/O, etc.

3. What is likely the performance bottleneck during the map phase?

Raspuns: CPU este folosit 100%

Describe what happens in the reduce phase: (3p)

4. Why does the reduce phase begin before the map ends? Is this behaviour correct?

Raspuns: partea de shuffle incepe imediat ce unii mapperi au terminat; reducerii ruleaza in paralel cu mapperii. Comportamentul este corect.

5. What is likely the bottleneck in the reduce phase? Note: the disks on the servers were profiled separately and it was found that the average

sustained write rate is 500Mb/s. What other experiments would you run to accurately determine the bottleneck?

Raspuns: disk-ul este bottleneck-ul.

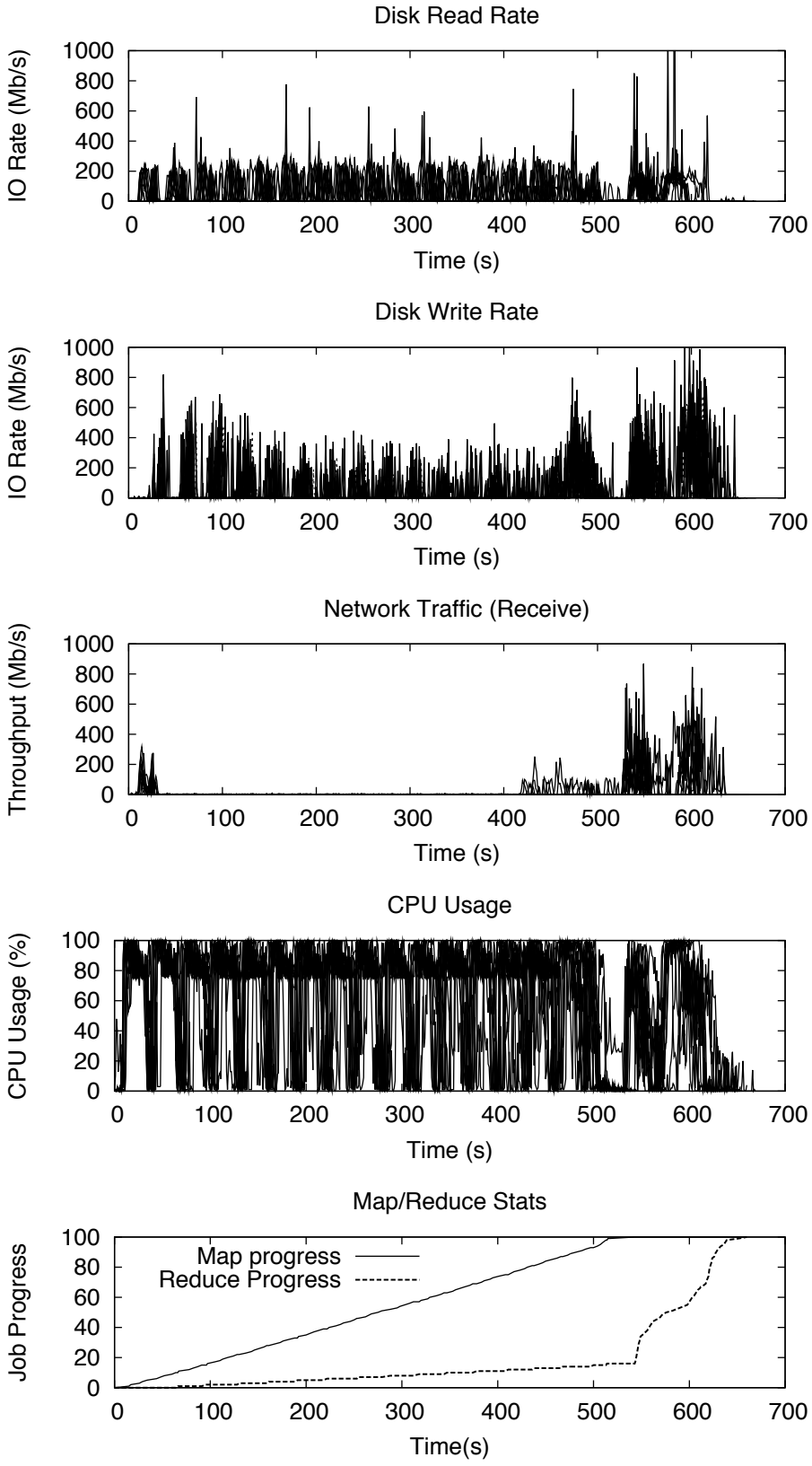
Improving performance (both software and hardware solutions can be considered). (3p)

6. What is the best way to increase performance in the map phase?

Raspuns: add cpus.

7. What is the best way to increase performance in the reduce phase?

Raspuns: faster disks – either flash disks or HDD in RAID.


Disk Read Rate
IO Rate (Mb/s)
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
Time (s)
Disk Write Rate
IO Rate (Mb/s)
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
Time (s)
Network Traffic (Receive)
Throughput (Mb/s)
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
Time (s)
CPU Usage
CPU Usage (%)
100
80
60
40
20
0
0
100
200
300
400
500
600
700
Time (s)
Map/Reduce Stats
Job Progress
Map progress
Reduce Progress
100
80
60
40
20
0
0
100
200
300
400
500
600
700
Time(s)

28/1/2011

Advanced Topics in Distributed Systems

EXAM Problem C

A distributed application runs on N servers in a data-center, with high bandwidth available between servers. The application writes objects to the system; each of these is replicated for availability. The application also reads objects constantly. A front-end server first processes requests for read/write, and directs reads and writes to the appropriate servers; it aggregates results, and finally replies to the client. The front-end knows all the N servers in the system. All N servers are similar (CPU/memory/disks).

Questions:

1. Design the system such that it maximizes write throughput (i.e. the number of object writes/s). Do not make any assumptions about the distribution of objects (i.e. it is allowed that a single object is updated all the time). Eventual consistency is sufficient.

   You need to specify how many replicas are stored per object, and which servers store the replicas. Describe how reads are served, and how to replicas are updated. Reason about the expected bottlenecks in your system. Is there anything that can be done to alleviate them? (4p)

Raspuns: Fiecare obiect va primi un ID unic. Designul va folosi minimul de doua replici ca sa asigure toleranta la defecte (fie nr de servere pe care e stocat un obiect W=2); cresterea nr de replici ar creste costurile pentru update, deci nu e oportuna. Replicile vor fi scrise pe doua servere aleatoare la fiecare updatare (asta pentru a evita bottleneck-urile la scrierea aceluiasi obiect pe acelasi server). Citirile vor consulta toate serverele (R e nr de servere contactate pentru read, R=N) afland ultima varianta a obiectului cu ID-ul dat. Se va intoarce cea mai proaspata versiune.

2. Assume you know the ratio of reads to writes, and that this ratio changes over time. Design the system to maximize throughput (i.e. the number of transactions per second). You need to specify how many replicas are stored per object, and which servers store the replicas. Describe how reads are served, and how to replicas are updated. HINT: the number of replicas might change with time. (3p)

Raspuns: se va creste W si se va reduce R astfel incat R+W>N pentru a ne asigura ca gasim ultima varianta a fiecarui obiect. (Sistemul este un quorum system). Valorile W si R se aleg astfel incat munca in sistem w * nr_writes + r * nr_reads este minima.

3. Assume the system needs strong consistency: as soon as one client sees the new version of an object, all clients should subsequently see the new version of that object. How do you design the system to achieve this goal,

and maximize system throughput? What changes compared to point (b)? (3p)

Raspuns: putem folosi Paxos la fiecare scriere pentru a asigura consistenta. Pentru a maximiza throughput-ul putem: a) reduce W si b) rula paxos pe batch-uri de update-uri, nu la fiecare update.