CS-553 Cloud Computing
Homework#02

Rohit Madhukar Lakde A20434998

1. Processors (21 points):

a. Today's commodity processors have 1 to 32 cores, with some more exotic processors boasting 72-cores, and specialized GPUs having 5000+ CUDA-cores. About how many cores/threads are expected to be in future commodity processors in the next five years?

Ans: We need to understand **Moore's law** for this question which says after every 18 months number of transistors on an integrated circuits doubles and cost of that reduces to half. But today we have reached at the extreme end of Moore's law and it is because of below reasons:

- **Electrical Leakage:** Transistors are getting smaller and more energy efficient, however they are getting smaller as small as 10nm and the channels which are carrying current cannot always contains it, which results in heat and that can damage transistors and its performance.
- **Increased cooling cost:** As the number of transistors increases heat generated by them also increases which results in increased cost of cooling, so to minimize the cost we need to minimize the number of transistors.

As there is no other way to improve the improve the performance of processors, Researchers are concentrating on putting more and more cores on same chip. But according to **Amdahl's law** if we are doing a task on parallel platform then only part of task can be optimized with a greater number of cores. Below are few reasons for that.

- **Memory Management:** As CPU speed increases exponentially with multiple cores but accessing memory could not match with processor speed which can cause bottleneck so there in no gain in increasing core count after some limit.
  e.g. With given configuration performance of 16 core = performance of dual core
- **Programs with frequent memory access:** Programs can be parallelized, but data needs to be moved from memory to different cores so having 2-4 cores is good than having 40-50 cores which might take more time to access memory.

Considering above points and end of Moore's law we can conclude that we would not see any progress processor and threads can be increased to some extent. But there is need of new technology like Graphene Processor, Photon transistors to improve performance of processors further.

b. How are these future processors going to look or be designed differently than today's processors?

Ans: Below are the below techniques on which engineers are working on

1. **Graphene Processor:** Graphene transistors can conduct an electricity much more easily and efficiently; this kind of transistors can increase the clock speed by factor of 10 or 100. Which means way lower power and far low heat.
2. **3D chips:** It is a one of the best solutions for a manufacturer which are looking for performance increase with reducing the size of processor. One chip is placed on another with the help of wafer bonding technique. Carbon nanotubes are used to place memory on top of processor which increases the performance by 1000 times.
3. **Photon Transistors:** In this type beams of lights are used to perform on off operations, Light is emitted on the optical thing present at the input and output varies according to the conditions. Intensity of light is measured to perform operations in this type of transistors.

c. What are the big challenges they need to overcome?

Ans: To build new processors below problems need to overcome

1. **Challenges with graphene processor:** Graphene is very efficient material to conduct charges and it is very fast in pushing the charges, but problem is that it cannot retain the charge. It is necessary to retain the charges in binary system so that your code cannot close the instances they are running.
2. **Challenges with 3D chips:** Integrating heterogeneous components with chips, making it secure and trustworthy, Bandwidth, power and signal integrity requirement for all the interconnects, testing each component individually are few challenges in 3D chips.
3. **Challenges with photon transistors:** Main problem with photon transistors is that we cannot make photons to interact with each other. Software's need to achieve exponential speedup to configure with photon transistor.

d. Describe what a core and hardware thread is on a modern processor, and the difference between them?

Ans: Difference between hardware thread and modern processor is as follows:

Multiple Cores	Hardware Thread
If we consider a case of 2 core system, then physically two cores are present in system.	If we consider a case where single core is present with hyperthreading then it acts like a two logical CPU's to operating system.
If a task is divided into two threads, then in two core system both can work concurrently	In hyperthreading system with one core, two cores cannot work parallelly, if one is stopped and waiting in that case only another virtual thread can run.
More efficient as compared to hardware threads	Not more efficient than multiple cores

e. What type of workloads are hardware threads trying to improve performance for?

Ans: Hardware threading is mainly used for:

1. ***Hardware:*** Hyper threads with one core share everything between them, so most benefit for this is to keep execution status to avoid context switch which saves a lot of time.
2. ***Good usage:*** When input output operation is very slow and there are number of calculations to do, in this case hyper thread system will be more efficient as hype threads share data in between them. It is good to use hyper threads when there are number of IO operations involved on single core system.
3. ***Workflows with multiple tasks:*** When we have complete workflow tasks or sequence of task and we need multiple things to work at once, in such cases hardware threading is useful.

f. Compare GPU and CPU chips in terms of their strength and weakness. In particular, discuss the tradeoffs between power efficiency, programmability and performance.

Ans: Differences as follows:

Attribute	GPU	CPU
Long form	Graphic processing unit	Central processing unit
Power efficiency	As GPU performs more functions with the same resource as compared to CPU so GPU's are more power efficient as compared to CPU	CPU contains more powerful cores but only few, so to do same amount of work done by GPU, CPU needs more power than GPU.
Programmability	Programming in GPU is very difficult as all the programs run in parallel, we need to take care of resource locking and releasing, mutual exclusion cases. CUDA is used for GPU programming	Programming in CPU is simple, there is no need to consider locking mechanism for the resources.
Performance	GPU is consisting of thousands of threads so it can accelerate software by 100x over CPU, it is good for parallel work	CPU is composed of only few cores so it can handle only few threads at a time, so it is good for sequential work.

g. Why do we not have processors running at 100GHz today (as might have been predicted in 2000)?

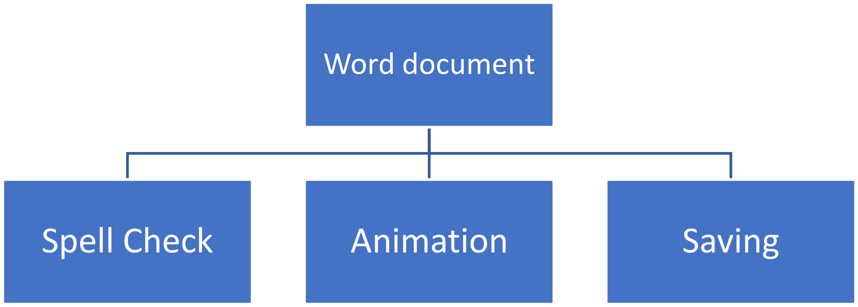
Ans: Answer of this question is lies in Moore's law which states that after every 20 months number of transistors doubles and its price reduces to half. Till now we followed the Moore's law and distance between two transistors today is around 5nm. We have at max i9 processors which has frequency range of 3.3 – 4.3 GHz, making it 1000Ghz means increasing the transistors in the same multiple which is not possible. We cannot fit that many transistors on the same chip we need to search for new techniques now.

2. Threading (21 points):

a. Why is threading useful on a single-core processor?

Let's consider an example of running a word document to understand it

1. ***Without threading***: We perform many tasks in word like spell check, animation and Save document. After updating changes to document and clicking save button our expectation is to check content for spell check and save document at the same time, but actually it wont happen concurrently so it will take a lot of time, Case will become worst when another high priority task ask for resource then context switching will happen which results in delay.
2. ***With Threading***: As given in below hierarchy running a word document is our process which we are dividing into three threads Spell check, Animation and Saving. Once we run out process all process will concurrently which will result in efficiency. Suppose there is one high priority task asking for resource which also need system call, as system call requires a lot of time, in that time CPU can execute out threads will result in efficiency.



b. Identify what a thread has of its own (not shared with other threads):

Ans: Process details like code, data and files are maintained in PCB which are shared between all the threads. Each thread has their own registers, program counter and stack which is private for each thread and they don't share it.

c. What is the advantage of OpenMP over PThreads?

Ans: Advantages of OpenMP over PThreads are as follows:

1. OpenMP is much higher level as compared to PThreads.
2. It is not required to understand where to put locks and where to release it in OpenMP, we just have to assign the task and ask to do it parallelly.
3. As Pthread is language specific which is library in C to work with concurrency, While OpenMP is doesn't limit you to specific language.
4. To achieve the same performance Pthreads require a lot of work as compared to OpenMP.
5. Pthread provides detailed control over threads which OpenMP doesn't.

d. Do more threads always mean better performance?

Ans: No. Number of threads is not directly proportional to performance after some limit, Even Amdahl's law states the same, there is some percentage which can be run in parallel and another cannot be run in parallel.

Let's consider

- Our program takes 10msec to run (without using threads)
- we have 5 cores present in our system.
- Consider thread spawning and terminating takes 2msec.

Let's switch to four threads now five threads will run in parallel to complete the program, so time taken will be **4msec.** (10/5 + 2)

Not let's use 100 threads which will run in parallel to complete the program. As we have five cores all 100 threads cannot run in parallel, even if we skip program execution time then overall time = **40msec** (thread spawning and terminating). We have not considered Context switching, program execution time, systems or IO calls here which might add same amount of time again. So, it's good to use five threads instead of 100 threads.

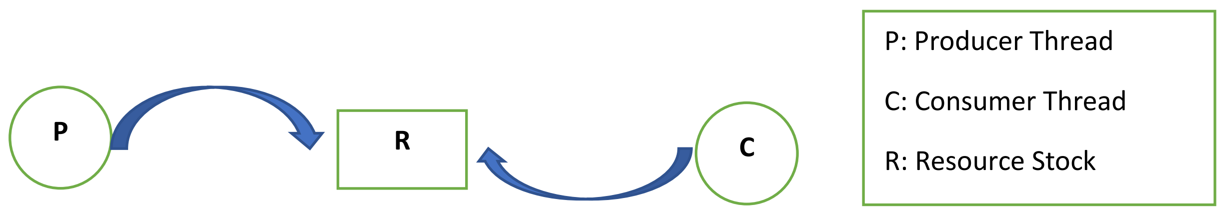
e. Is super-linear speedup possible? Explain why or why not.

Ans: Suppose our program takes x sec to execute without any parallelism on single core system, now if we use 4 cores to execute the same program with parallelism then it should take greater than x/4 sec to execute considering memory transfer, context switching etc. We call it super linear speedup when it takes less than x/4 sec to execute. It is possible due to because of Cache effect and Memory hit pattern.

- ***Cache Effect:*** Lets consider we have 4 core machines, each has 1MB cache memory, program execution needs 3MB of data. When we consider above scenario for single core system then obviously it will a lot of time to execute given program as majority of the time will be doing data movement between memory cache and RAM. In case of 4 core system, it will only have to move data between CPU and cache. In this way we can achieve super linear speedup.
- ***Memory hit pattern:*** It is a memory management technique in which the memory which was hit maximum time will be placed in more accessible place like cache. If many threads are accessing the same memory, then it will increase its hit and will be placed in cache which might cause super linear speedup.

f. Why are locks needed in a multi-threaded program?

Ans:



Let's consider an example to understand the question. Suppose we have two threads producer and consumer which are running concurrently and one resource stock. Producer gets the resource stock and produces output in it, after that producer releases the resource. Then consumer asks for the resource and stock and consumes the output. Now consider the case where consumption rate of consumer is very high compared to producer's production rate, in that case consumer will consume all the resources present in stock and will lock the stock resource, As the resource is locked by consumer thread producer cannot get that resource to produce so it's a kind of deadlock. Given problem can be solved by locks, when producer wants to produce something it will lock the resource, so that at the same time another thread cannot use the same resource, once it releases the resource consumer can lock it to consume the output.

g. Would it make sense to limit the number of threads in a server process?

Ans: It is mainly because of two reasons:

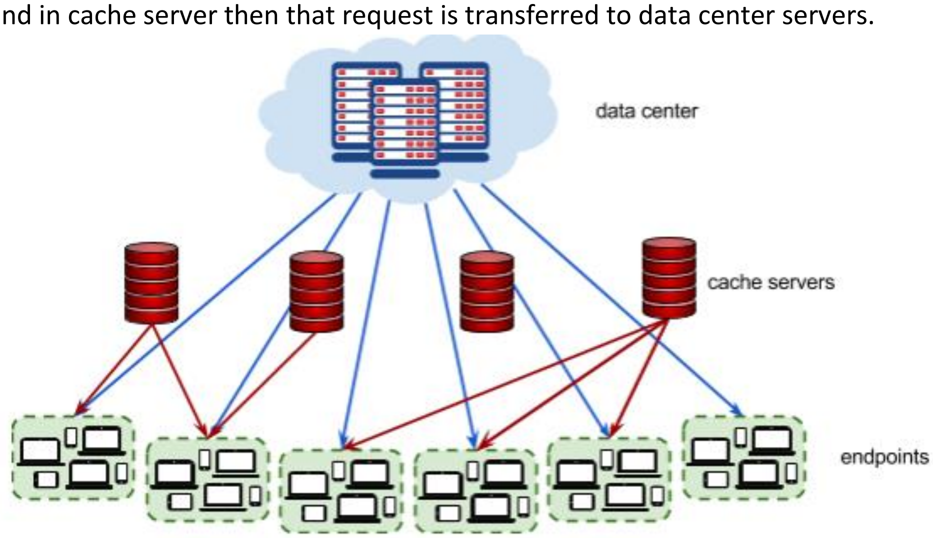
1. Threads have their own memory that is register, program counter and stack which is not shared with any other thread and processes. Increasing several threads unnecessary will take a too much of memory which might degrade the system performance.
2. To an operating system behavior of threads is always confused, when it comes to virtual memory it is always difficult to maintain relative stable set, which might cause memory of IO errors due to because of reference error, so having many threads means probability of such errors is very high.
3. If our task is small and we are sharing given task between number of threads, then majority of the time will be spent on spawning and terminating the threads.

3. Network (11 points):

a. A user is in front of a browser and types in www.google.com and hits the enter key. Think of all the protocols that are used in retrieving and rendering the Google logo and the empty search box. Describe the entire sequence of operations, commands, and protocols that are utilized to enable the above operation.

Ans: Below are the activities happen in background when we enter given URL in browser and hit enter button.

1. It checks the IP of given URL in browser cache memory.
2. If IP address is not found in cache memory, then it checks it in OS cache.
3. If it is not found in OS cache, then it goes to DNS (domain name system) to find out IP address of given URL.
4. It gives error "DNS address not found" error if DNS entry is not present for given URL.
5. Google search server structure is as shown in following diagram, where records or searches which are trending are placed into cache server to reduce the overall latency. If information is not found in cache server then that request is transferred to data center servers.



6. DNS will return IP address of more than one Google servers to browser.
7. Browser will pick first IP address and try to build **TCP** connection with the server. This connection is established with TCS three-way handshake method which is like:
    i. Browser will send SYN package to server to ask if it is open for new connection.
    ii. If server is open for new connection then it will send sync acknowledge package in the return, if it is not ready then it will pick new IP returned by DNS and try to establish connection with it.
    iii. After receiving ACK package, browser will send acknowledge package to server, and finally connection is established.
8. Browser will send HTTP request to the server to get HTML data. First it will send the POST request to server(as www.google.com is converted into https://www.google.com) which will send few attribute values about our browser to server.
9. Once server receives the HTTP request, it passes it to request handler to generate the response.
10. Request handler then assembles the content in the form of HTML, JSON and XML.

11. Finally, server sends the HTTP response back to the browser along with the HTML content.
12. HTML content is then loaded into browser and shown to user.

4. Power (12 points):

a. Why power consumption is critical to datacenter operations?

Ans: There are mainly two reasons for that which are as follows:

1. ***To reduce cost:*** 20% of the total cost required to build on data center is spent on power and cooling infrastructure. It's a very big amount so it is very necessary to reduce overall power consumption and cost of cooling at minimal level.
2. ***Prevent hardware components from damage:*** Even if we are not considering the cost, suppose we are using a lot of power to run our datacenter. We know power is directly proportional to work done and work done is proportional to heat produced. As we raise our power supply heat generated in system increases which results in system damage.

b. What is dynamic voltage frequency scaling (DVFS) technique?

Ans: It's a power saving technique in computer architecture by adjusting the value of frequency or voltage of CPU or attached peripherals. It consists of two techniques Dynamic voltage scaling and Dynamic frequency scaling.

1. ***Dynamic voltage scaling:*** In this case voltage is adjusted to reduce the power consumption, there are two types of DVS knows as 'undervolting' and 'overvolting'. Increasing in voltage is known as overvolting and decrease in voltage is known as undervolting. Undervolting is to conserve the power which is mainly used in devices having batteries like laptops, mobiles. Overvolting is for increasing computer performance.
2. ***Dynamic frequency scaling:*** It's a technique where frequency of the computer is adjusted to reduce power consumption, it is also known as CPU Throttling. It helps to preserve the batteries which reduces the cooling cost. Software's put overclocked frequencies into dynamic frequency scaling algorithms to avoid chip degradation.

c. If you were to build a large $1B data center, which would require $50M/year in power costs to run the data center and $50M/year in power costs to cool the data center with traditional A/C and fans. Name 2 things that the data center designer could do to significantly reduce the cost of cooling the data center?

Ans: There are two techniques which we can use to reduce the cost of cooling of data center.

1. Immersion cooling: It is a technique in which hardware components are directly immersed into non-conducting liquid. It reduces the overall size of datacenter by 10 times, talking about power consumption cost it saves around 95% of the cost, so it's very efficient technique to save power. In above case we can save **$47.5M** per year.
2. By building datacenter at the top of hill or nearby river so that we can pass cold water through tubes in hardware will also save cost of cooling, this technique is known as liquid cooling.

d. Is there any way to reduce the cost of cooling in (C)? If yes, how low could the costs go? Explain why or why not?

Ans: Overall 5% of the worlds total electricity is required for datacenters out of which 38% of the energy usage is for air cooling electronics. There are few ways in which we can reduce the cost of cooling which are as follows:

1. ***Immersion cooling:*** It is a technique in which hardware components are directly immersed into non-conducting liquid. As compared to previous techniques instead of spreading out the components to cool, it packs all the components to put into liquid, so it saves a lot of space. Talking about the energy saving, it reduces the energy cost up to 95%. That's why it is one of the efficient cooling techniques.
2. ***Central air conditioning system:*** In small data centers air conditioners are used to keep your data centers at low temperature. It results in wastage of lot of energy and air.
3. ***Liquid cooling:*** It is technique in which cool water is passed into the hardware components through tubes, it is a better cooling technique than using fans or central AC's.

5. Storage (15 points):

a. If a manufacturer claims that their HDD can deliver sub-millisecond latency on average, can this be true? Justify your answer?

Ans: In some case it is true if we consider the cache implementation, but it will only work for the short period of time but if we have to search it on entire drive then cache will be useless.

Let's consider HDD with 6000 rpm

1. Rotations per minute are 6000
2. Which is 100 rotations per second
3. One rotation time is 1/100 sec = 10msec

Even if we consider high end machines having 18000 rpm then we get average latency of 1.5msec which is far greater than sub millisecond.

To complete one rotation, it will take 10msec, so average latency of the disc is 5msec, but in given example it is saying average latency of sub millisecond which is not possible.

b. Explain why flash memory SSD can deliver better performance for some applications than HDD.

Ans: It is due to because of following reasons:

1. All the data present in flash memory SSD can be accessed at the same time there is no need to locate data first and then read as compared to HDD.
2. Flash memory SSD is usually faster which has speed of 200 to 600 MB/s where as speed of HDD is around 20 to 60 MB/s.
3. No mechanical parts like motor, moving hand are present so it requires less power which generates less amount of heat so more efficient.

c. What types of workloads benefit the most from SSD storage?

Ans: Below type of workloads benefit the most from SSD.

1. Any program which requires to read or write memory from different location in SSD of the drive will get performance boost using SSD.
2. Programs which are present entirely in SSD and which needs data from SSD will have benefit from SSD storage.
3. If we put frequently used programs like Microsoft office, browser, outlook in hard drive then it will show performance improvement

d. If a manufacturer claims they have built a storage system that can deliver 1 Terabit/second of persistent storage per node, would you believe them? Justify your answer to why this is possible, or not. Make sure to use specific examples of types of hardware and expected performance.

Ans: When it comes to solid state drives IOPS of 50K/sec – 500K/sec can be achieved with the help of SATA and SAS interfaces and throughput range is around 500MB/sec. Throughput of 4000 MB/sec can be achieved with the help of NVMe(Non-volatile memory express) interface. I didn't hear anything like 1 Terabit/second in reality. Even we cannot consider HDD, as performance of HDD is weaker than SSD. Some improvement can be achieved when we use multiple heads in HDD but 1Terabit/sec is not possible.

e. In this problem you are to compare reading a file using a single-threaded file server with a multithreaded file server. It takes 16 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming the data are in the block cache. If a disk operation is needed (assume a spinning disk drive with 1 head), as is the case one-fourth of the time, an additional 32 msec is required. What is the throughput (requests/sec) if a multi-threaded server is required with 4-cores and 4-threads, rounded to the nearest whole number?

Ans:

For single threaded system cache hit time = 16 msec

Cache miss time = 48 msec

Average memory access time = h*tc + (1-h) *(tc + tm)

Where h = hit ratio

Tc = cache hit time

Tm = cache miss time

=¾ * 16 + ¼ * 48 = 12 + 12 = 24 msec

So mean request takes around 24msec for completion.

Server can do 1000/24 requests per second = 41.66 approx 42 requests per second

In multithreaded system disk waiting time is dominated by thread creation locking and all so single request will take 16 msec which is like 62.5 request per second.

Multithreading with 4 cores and 4 threads: Single request will take 16/4 = 4 msec which means 250 requests per second

6. SQL vs Spark (20 points):

a. You hired by a company to help them decide what software stack and hardware they should adopt to store, process, and analyze 100TB of data. Their choices for software stack are MySQL (https://en.wikipedia.org/wiki/MySQL) and Spark (https://en.wikipedia.org/wiki/Spark_(software)). It has been determined that most queries will only touch 10% of the data using primarily a random-access pattern. The computation to be done seems to be scalable, and that the more computing resources, the faster the computation will run, as long as it can be maintained in memory. The requirement is that there should be at least 200 cores of computing running at 2GHz of faster. There are no requirements on the processors used (as long as they are x86 compatible). There should be enough memory to store 10% of the dataset in memory, and there should be enough storage to reliably store 100TB of storage. If a multi-node approach is taken, the network should be as fast as possible (e.g. 100GbE) to ensure good scalability. Assume administration cost is 10% of a full-time system administrator (at a salary of $100000/year). Assume power costs $0.10 per KWH, and that cooling costs are in-line with the power costs of powering the hardware. Use the ThinkMate website (https://www.thinkmate.com) to come up with a solution for MySQL and one for Spark in terms of costs over a 5 years period, including hardware, power, cooling, and administration. Note that your solution has to be rack mountable (you cannot use desktops or laptops).

Ans: SQL is data retrieval software in which data is maintained in different table and each table is related to another table with the help of keys which we call as foreign keys. SQL data is maintained in single node so parallelism is not possible in sql where as Spark is an analytical and data processing engine where data is maintained across the cluster because we cannot maintain such a huge data on single computer that's why it runs on a cluster or server. Below are the cost details for SQL (single machine) vs Spark (clustered machines).

<u>For SQL system:</u>

As per the given conditions configured a system on THINKMATE having below features

- One 8 socket system
- 8 intel processors having 26 core each which are equal to = 26 * 8 = 208 cores.
- 64 GB DDR4 with quantity 96 = 6 TB
- Additional memory 60 * 64 GB = 4 TB
- Total memory = 6 + 4 = 10 TB
- 13 hard discs with 7.68 TB = 13 * 7.68 = 100 TB

Overall cost estimate = $221K

Your Order

Product	Unit Price	Quantity	Subtotal
SuperServer 7089P-TR4T (My System February 24th, 11:55 pm EST) Thinkmate Config ID 334787 Supermicro SuperServer 7089P-TR4T - 7U - 16x SATA/SAS - 2x M.2 - 96x DDR4 - Quad 10-GbE - 1600W Redundant 8 x Intel® Xeon® Platinum 8164 Processor 26-core 2.00GHz 35.75MB Cache (150W) 96 x 64GB PC4-21300 2666MHz DDR4 ECC Registered Load-Reduced DIMM No SATA Disk on Module for boot 13 x 7.68TB Micron 5200 ECO Series 2.5" SATA 6.0Gb/s Solid State Drive Supermicro AOC-S3008L-L8e SAS 3.0 12Gb/s 8-Port Host Bus Adapter Supermicro SIOM 1-Gigabit Ethernet Adapter AOC-MGP-i2M (2x RJ45) 8 x Supermicro Update Manager (SUM) (OOB Management Package) No Operating System System Assembly and Testing (8-blade enclosure) 3 Year Depot Warranty (Return for Repair) Reconfigure Modify Spares Kit	$176,852.00	1 Remove	$176,852.00
Spares Kit for SuperServer 7089P-TR4T 60 x 64GB PC4-21300 2666MHz DDR4 ECC Registered Load-Reduced DIMM No SATA Disk on Module for boot 7.68TB Micron 5200 ECO Series 2.5" SATA 6.0Gb/s Solid State Drive Supermicro - FAN-0171L4 92x92x76 mm, 10.5K-10.8K RPM, Counter-rotating Exhaust Fan for SC718 X10 8-way Server Supermicro PWS-1K62A-1R - 1U 1600W Titanium Level Power Supply	$44,228.00	1 Remove	$44,228.00
		Update	$221,080.00

Power consumption for SQL:

- Power required for 8 pin socket is(as per THINKMATE) = 8000W = 8000 * 8 hours/day
  = 64000 WH/day
  = 64 KWH/day
  =12.8 * 365 = 23360 KWH/year
  =23360 * 5 = 116800 KWH/5Years
- Power cost: 116800 * 0.10 = $11680 for 5 years.

Cooling cost for SQL (equals to power cost as given in question): $11680

Administrator cost for 5 years: $500000

Administration cost for 5 years: $50000

<u>Clustered network for SPARK:</u>

As per the given conditions configured a clustered system on THINKMATE having below features

- Rackmount server with two processors having 28 cores each = 26 * 2 *4 =208
- 2 intel processors with 28 cores in each rack
- 2.5 TB memory in each rack
- Hard disc of 25 GB in each rack
- 100 Gigabit ethernet adapter

Overall cost estimate = 28873 + 3172 = $32045

Cost for four such racks = 32045 * 4 = $128180

Let's take one high end switch to connect our systems of worth $10000

Overall cost for whole cluster = 128180 + 10000 = $138180

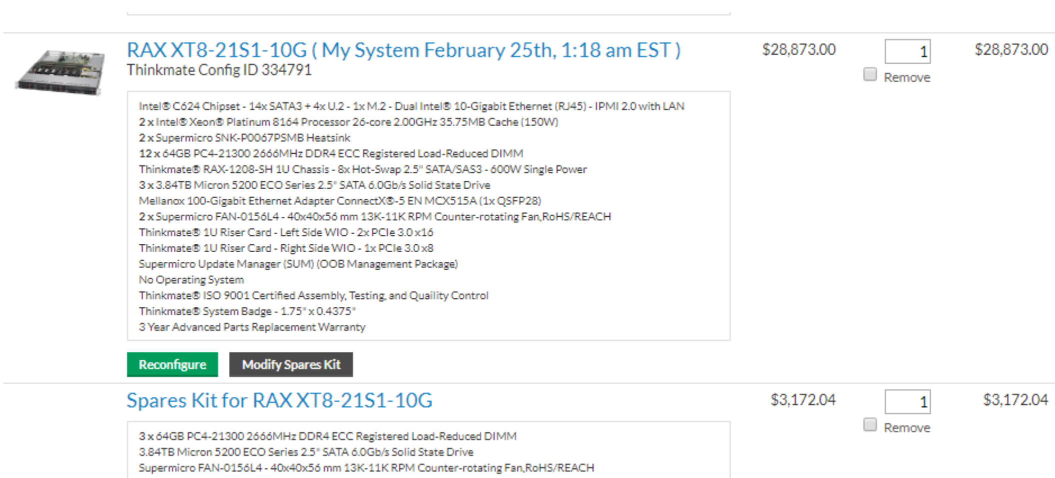
Power consumption for SPARK:

- Power required for 1 rackmount (as per THINKMATE) = 300W = 300 * 8 hours/day
  = 2400 WH/day
  = 2.4 KWH/day
  = 2.4 * 365 = 876 KWH/year
  = 876 * 5 = 4380 KWH/5Years
- Power cost: 4380 * 0.10 = $438 for 5 years.
- Power cost for 4 racks = 438 * 5 = $2190

Cooling cost for SPARK (equals to power cost as given in question): $2190

Administrator cost for 5 years: $500000

Administration cost for 5 years: $50000



Cost estimation for each system:

Attributes	SQL	SPARK
System cost	$221000	$138000
Administrator cost 5 years	$500000	$500000
Administration cost 5 years	$50000	$50000
Power cost	$11680	$2190
Cooling cost	$11680	$2190
Total	**$794360**	**$692380**

As per the estimations above I would suggest SPARK software stack is much better and cost efficient as compared to SQL.