Andrew J. Dupree and Tony Vivoli
Dr. Kozyrakis
EE282: Computer Systems Architecture
6/5/2013

<u>Homework 3</u>

**Problem 1**

*A)*

| | |
|---|---|
| BarossoA TCO: | 13.879285 million per month |
| Hamilton TCO: | 3.5306750 million per month. |

The BarossoA datacenter has a significantly higher TCO than that of the Hamilton facility. The difference seems to come largely from the server cost.

| | |
|---|---|
| BarossoaA c_server: | 9.96429294 million/month |
| Hamilton c_server: | 1.99810277 million/month |

Given that the two datacenters have roughly the same number of racks, BarossoA is paying significantly more per server per month. This facility should consider purchasing less expensive server hardware (if it would fit their needs), or trying to use their servers for longer.

*B)*

The comparison is still largely valid, as the TCOs between the data centers change very little. In both data centers power scales neatly with utilization, the power consumption of the 50% utilization cases being 53% of their previous power consumptions. In the original utilization case cost of power was a larger percentage of the TCO in the Hamilton facility over the BarossoA facility, so it could be said that the Hamilton facility benefits more from this utilization reduction.

*C)*

Theoretically, costs will increase if we replace our existing servers with higher capacity versions. The servers will be more expensive themselves, power consumption will increase, cooling costs will increase, and the cost of maintenance will increase as well with more hard drive failures. Overall, TCO will rise.

This is born out in simulation. The original TCO (with correct server cost/power) is:

| | |
|---|---|
| tco: | 3716203.28 $/month |
| c_infrastructure: | 818569.68 $/month 22% |
| c_server: | 2091806.89 $/month 56% |
| c_power: | 510745.45 $/month 14% |

The Hamilton TCO with higher capacity servers is

| | |
|---|---|
| tco: | 4143279.84 $/month |
| c_infrastructure: | 950240.05 $/month 23% |
| c_server: | 2298507.18 $/month 55% |
| c_power: | 599451.36 $/month 14% |

*D)*
Considering that RAM is relatively inexpensive and consumes less power than disk, this should increase performance with a relatively modest increase in TCO.

| | |
|---|---|
| tco: | 4229617.28 $/month |
| c_infrastructure: | 972185.11 $/month 23% |
| c_server: | 2348115.25 $/month 56% |
| c_power: | 614235.68 $/month 15% |

*E)*
With these changes made, we now have the above TCO of 4.229 million/month, and the following performance numbers:

| | |
|---|---|
| Performance: | 45978 |
| PerformanceOverTCO: | 10.87 x1000 |
| PerformanceOverWatt: | 3.11 x1000 |

This question is difficult to answer accurately due to inaccuracies in the simulator. The MTTR and staff salary options had no effect on TCO and performance whatsoever, though we would expect them to. The PUE and server lifetime parameters function correctly. As we expect from Amdahl's law and analyzing the percentage contributions to TCO in part D, increasing the server lifetime brings down the cost most significantly. Increasing the lifetime from three years to the maximum reasonable lifetime of five years brings the TCO down to 3.338 million. At this point we could bring the PUE down from 1.4 to a state-of-the-art 1.15, but given that power is already such a small contributer to our TCO we only see a small improvement, a drop to 3.232 million.

Realistically speaking at this point we would have to reduce the cost of our server equipment to bring the price down reasonably. This would involve reverting back to either our higher latency or lower capacity server configurations.

**Problem 2**

*(a)*
Since the difference between a tier 1 and tier 2 data center is the duplication of the power and cooling infrastructure, to upgrade to tier 2 you need to pay the power and cooling infrastructure cost twice. This is an additional $765,369 added to the total cost of tier 1, $3,799,911, giving a total cost for a tier 2 data center of $4,565,280.

*(b)*
The table below can be used to determine whether or not paying extra for higher availability is justified depending on the application. For example, for brokerage applications it is very expensive if the servers go down, so it will likely be worth spending more money on higher reliability. However, at the opposite extreme, for ATM service fees it is relatively inexpensive for the servers to be down, and the money lost due to the servers being down is likely much less than what it would cost to ensure higher uptime.

In summary, if it is not very expensive when servers are down, a data center designer will likely choose not to spend extra infrastructure costs to ensure higher availability. But if it is very expensive when the servers are down, it may be more cost effective to spend extra money to ensure higher availability.

**Problem 3**

Note: Wolframalpha was used for all of the calculus/summations/algebraic simplifications

*(a)*
-First, solve for the probability that a single node will not fail in some time interval of length t (note: it is assumed that this time interval t is extremely small, and at the end of the problem we will take the limit as t approaches zero to switch from a discrete time interval to a continuous model)
-For a processor node:

$$X(k) = \frac{e^{-nt}\,(n\,t)^k}{k!}$$

-For a disk node:

$$X(k) = \frac{e^{-mt}\,(m\,t)^k}{k!}$$

-We are interested in the probability that each node does not fail in a time interval, which is the k = 0 case. Both equations become the following for k = 0:

$X(0) = e^{-nt}$, $X(0) = e^{-mt}$

-The probability that the entire system does not crash in a time interval t is the probability that all

N nodes do not crash times the probability that all M disks dont crash, which is all M times N individual probabilities multiplied together. This gives:

$$X(0) = e^{-m M t - n N t}$$

-We want the probability that the system will crash at time interval i. This means that for i-1 intervals the system did not crash, and on the ith interval it did crash. This yields the equation:

$$X(i) = \left(e^{t\,(-(m M + n N))}\right)^{-1+i} \left(1 - e^{t\,(-(m M + n N))}\right)$$

-The expected value for which interval the system will crash is found with the following summation : (note, the equation is multiplied by t to convert from expected number of intervals until failure to expected number of seconds to failure)

$$\sum_{i=1}^{\infty} i\, t\, (1 - \exp(-(t\,(m M + n N))))\, \exp(-((t\,(m M + n N))\,(i - 1)))$$

This summations yields:

$$\frac{t\, e^{t\,(m M + n N)}}{e^{t\,(m M + n N)} - 1}$$

as the expected time until the system crashes.

-This, however, is still a function of how the time interval t. In a real system, there are not discrete chunks in which a processor can fail. Instead it is a continuous function. To get a continuous model for failure, just take the limit as t approaches 0:

$$\lim_{t \to 0} \frac{t\, e^{t\,(m M + n N)}}{e^{t\,(m M + n N)} - 1} = \frac{1}{m M + n N}$$

-**<u>Above is the MTTF for the system</u>**

-Availability = MTTF / (MTTF + MTTR), so the availability is

$$\frac{1}{R\,(m M + n N) + 1}$$

*(b)*

Plugging in the given numbers and solving for R, you get R = 9.0918

*(c)*

Using the equation for MTTF in part A and the numbers in part B, the MTTF **is 25.25** hours.

**Problem 4**

| RAID Level | Small Read | Small Write | Big Read | Big Write | Space |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 1/2 | 1/2 | 1/2 | 1/2 |
| 3 | 1/2 | 1/2 | 1 | 1 | (G - 1) / G |
| 5 | 2 | 1/2 | 2 | 2 | (G - 1) / G |
| 6 | 1/2 | 1/2 | 1 | 1 | (G - 2) / G |

(b)

1) True. RAID 6 can tolerate two disk failures, so a single disk failure will not crash the system. Note that if there is no redundancy in the raid controller, if the raid controller fails, the system will crash (and my friends have had RAID controllers fail, so this probably shouldn't be overlooked!).

2) False. Performance-wise, RAID 4 is indeed strictly inferior to RAID 5, but the RAID controller is more complex and therefore more expensive for RAID 5.

3) False. RAID 2 doesn't rely on the failed disk to self-diagnose unlike RAID 5. (This is assuming RAID 2 is the memory-style ECC described in the book...)

4) True. Since RAID 1 can tolerate one disk loss, as long as a second disk does not fail while another is recovering, the crashed disk can successfully be recovered.