CS 350: Assignment #5

Due on Thursday, March 17, 2016

Bestavros 3:00 pm

Ben Pusey

1. $\lambda = 100$ rps

$T_S = 0 - 19$ ms uniformly distributed

bound on F-Scans unfairness

$\frac{1}{2}\lambda T_q = \frac{1}{2} q$ by littles Law

To solve for q we must use M/G/1 formula

$$q = \frac{\rho^2 A}{1-\rho} + \rho \qquad A = \frac{1}{2}\left[1 + \left(\frac{\sigma T_S}{T_S}\right)^2\right]$$

$$\rho = \frac{\lambda}{\mu} \qquad \mu = \frac{1}{T_S}$$

$$T_S = \frac{0+19}{2} = 9.5 \text{ ms} = .0095 \text{s}$$

$$\sigma T_S = \sqrt{\frac{1}{12}(b-a)^2} = \sqrt{\frac{1}{12}(19)^2} = 5.5 \text{ ms} = .0055 \text{s}$$

$$A = \frac{1}{2}\left[1 + \left(\frac{.0055}{.0095}\right)^2\right] = .67$$

$$\mu = \frac{1}{.0095} = 105$$

$$\rho = \frac{100}{105} = .95$$

$$q = \frac{.95^2(.67)}{1-.95} + .95 = 12.96$$

So the bound is $(\frac{1}{2})12.96$ or 6.5

Problem 2:

FCFS

Low Lambda

```
==========================================================
CLOCK: 100004.58204220934
Rho: 0.0098266704691827
IO System Time 0.012825090401281155
CPU System Time 0.3120715908599175
IO Tasks through system: 6029
CPU Tasks through system: 2995
==========================================================
```

Normal Lambda

```
==========================================================
CLOCK: 100000.00494249418
Rho: 0.9611812499997576
IO System Time 7.156759764158426
CPU System Time 7.436791653536298
IO Tasks through system: 600654
CPU Tasks through system: 300588
==========================================================
```

Slowdown I/O = $\frac{7.16}{.013} = 550$

Slowdown CPU = $\frac{7.44}{.31} = 24$

The slowdown for I/O is much longer than the slowdown for cpu because I/O jobs have to wait for the much longer cpu jobs ahead of them, and their optimal system times are very low.

SRTN

Low Lambdas

```
==========================================================
CLOCK: 100022.04952596084
Rho: 0.009745373282928895
IO System Time 0.010058497904441464
CPU System Time 0.30702980961107906
IO Tasks through system: 6025
CPU Tasks through system: 2993
==========================================================
```

Normal Lambda

```
==========================================================
CLOCK: 100000.23194325212
Rho: 0.9589099005456027
IO System Time 0.014010580748338622
CPU System Time 1.44634390924814
IO Tasks through system: 600748
CPU Tasks through system: 298971
==========================================================
```

Slowdown I/O = $\frac{.014}{.01} = 1.4$

Slowdown CPU = $\frac{1.45}{.3} = 4.8$

The slowdown for I/O is lower than the slowdown for CPU because I/O tasks have priority over CPU tasks since their service time is lower.

RR

Low Lambdas

```
==========================================================
CLOCK: 100020.82290120877
Rho: 0.009837938942820677
IO System Time 0.01049754071171709
CPU System Time 0.30351451323802425
IO Tasks through system: 5953
CPU Tasks through system: 3070
==========================================================
```

Normal Lambda

```
==========================================================
CLOCK: 100000.02893491843
Rho: 0.9615470237624546
IO System Time 1.9791537542270279
CPU System Time 7.383351984222936
IO Tasks through system: 600051
CPU Tasks through system: 300386
==========================================================
```

Slowdown I/O = $\frac{1.98}{.01} = 198$

Slowdown CPU = $\frac{7.4}{.3} = 24.66$

The lower slowdown of I/O in comparison to the I/Os Fifo slowdown makes sense as the time quantum is .1 so a I/O task only has to wait .1 seconds for CPU tasks ahead of them as opposed to their full service time.

HSN

Low Lambdas

```
==========================================================
CLOCK: 100002.867397461
Rho: 0.009363340133960237
IO System Time 0.010516938701498349
CPU System Time 0.2962037126131248
IO Tasks through system: 6114
CPU Tasks through system: 2971
==========================================================
```

Normal Lambda

```
==========================================================
CLOCK: 100000.07831014277
Rho: 0.96003786142339
IO System Time 0.1670700196483842
CPU System Time 3.9945701548944794
IO Tasks through system: 600314
CPU Tasks through system: 300119
==========================================================
```

Slowdown I/O = $\frac{.17}{.01} = 17$

Slowdown CPU = $\frac{4}{.3} = 13.33$

The slowdown for I/O and CPU are similar because HSN attempts to minimize the slowdown of the tasks in the system by servicing tasks with higher slowdowns.

IO Slowdown Rankings

1. SRTN slowdown = 1.4
2. HSN slowdown = 17
3. RR slowdown = 198
4. FCFS slowdown = 550

3.

a. The most obvious example would be if Q1 always has requests in it then requests in Q2 will never be serviced.

b. n-batch scan

c. A larger N makes the cache more efficient

d. A larger N is less fair though.

e.

- N = 100 The load on the system is low so we don't care about fairness. Setting N to 100 will also increase our systems performance
- N=10 The load is moderate so we should be more concerned about fairness. N=10 provides a good balance on efficiency and fairness.
- N=1 high load so fairness is a concern, caching provides little benefit here because there is little locality
- N=10 because the load is extremely high we should be concerned with fairness, however extremely high locality means caching will be effective. 10 provides a good balance between the two.

4.

a. yes starvation is possible. Class C jobs are the most susceptible because class b jobs are scheduled more frequently then class C jobs and have a higher priority.

b.

A event	1/minute	highest priority	5 seconds
B event	1/second	medium priority	.5 seconds
C event	1/5 minutes	lowest priority	20 seconds

A worst case 5 seconds, it has the highest priority

B worst case 5.5 seconds, an A event arrives during the B events execution.

C worst case 50 seconds. I used geometric series to think about this

event C takes 20 seconds to do work which will generate 20 B events, Those 20 B events take 10s to service so they will generate an additional 10 B events those 10 B events take 5s to service so they will generate 5 more B events. This is a geometric series so the 20 seconds of work on A create 20 seconds of work on B events. If an A event arrives its 5 seconds of work will generate 5 seconds of work on B events. Total is $20+20+5+5 = 50$s.

E4

c. $2x + \left(\left\lceil \frac{2x}{60} \right\rceil 5\right) 2 = 300$

$x = 125$ so 125 is the maximum amount of cpu time for C.

d. if a Job C arrives just before a Job A the Job A will have to wait for Job C to finish using The resource R before it can begin to be serviced. Therefore the higher priority Job A must wait for the lower Priority Job C.

e. A Job C arrives before Job A. Then The C Job is interrupted by Job Bs causing Job C to take 40 seconds to complete Job A then Runs for a total of 45 seconds.

f. The above is an example of priority inversion even though Job A has higher priority then Job B Job B's are serviced before Job A. A higher priority Job is preempted by a lower priority one.

g. Priority inheritance attempts to solve the problem of priority inversion. The priority of a low priority task is increased to the highest priority task waiting for the shared resourse being consumed by the low priority task.

h. Job Bs can no longer interrupt Job Cs using the resource so As worst case service time is 25 s