Homework 2 - Solutions

CS 414, Spring 2009
Instructor: Klara Nahrstedt

Note: Homework is an individual effort, i.e., no working in groups. Consider the homework as a preparation for your midterm. The deadline for HW2 is **Wednesday, May 6, midnight**. You can email your solutions to klara@cs.uiuc.edu in **pdf** format or you can slide the homework solutions in **paper form** under the door of the office 3104 Siebel Center. ***No handwritten homework!!!***

Problem 1: (20 Points)

1.1 **(10 Points)** Consider two MJPEG compressed video streams and one audio stream (stored on a single multimedia disk) with the following characteristics:

- Video v_1 (Playing Video): 20 frames per second, each compressed frame on average is of size 16 Kbytes, processing time for reading the frame from a disk is estimated at 10 milliseconds (ms) and the processing time for displaying the frame is estimated at 10 ms.
- Video v_2 (Recording Video): 10 frames per second, each compressed frame on average is of size 32 Kbytes, processing time for reading the frame from video/compression card is 20 ms, processing time for storing the frame to the disk is estimated at 25 ms.
- Audio a_1 (Playing Audio): 5 samples per second, each sample is 8 Kbytes, processing time for reading a sample from the disk is 5 ms, and processing time for writing it to the CD-quality audio speaker device is 4 ms.

Determine which preemptive multimedia scheduling algorithms are suitable for scheduling the tasks on these three streams (compare at least two multimedia suitable scheduling algorithms). Points will be given for

(1) **(2 Points)** specification of admission tests for these two algorithms,

Solution: RMS admission test

$\sum ei/pi \le \ln 2 = 0.69314$

$10/50 + 10/50 + 20/100 + 25/100 + 5/200 + 4/200 = 179/200 = 0.895 > 0.69314$

So the tasks for streams v1, v2 and a1 are not admissible together via RMS scheduling since the admission test is not satisfied;

EDF admission test: $\sum ei/pi \le 1$; $0.895 < 1$; hence tasks for streams v1, v2, a1 are admissible together via EDF scheduling since the admission test is satisfied

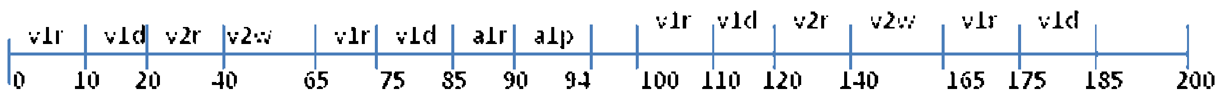
(2) **(6 Points)** specification of schedules for these algorithms (if they exist),

Solution: EDF Schedule:

Let us assume v1r (read frame from disk), v1d (write frame to display)

Let us assume v2r (read frame from video card), v2w (write frame to disk);
Let us assume a1r (read sample from disk), a2p (write sample to speakers)

The EDF schedule for the common hyper-period is v1r, v1d, v2r, v2w, v1r, v1d, a1r, a1p, v1r, v1d, v2r, v2w, v1r, v1d;



(3) **(2 Points)** determining the number of context switches in this example within the common hyper-period, i.e., the first 200 ms).

Solution: Number of context switches for EDF is 13 by the end of the hyper-period of 200 ms.

1.2. **(6 Points)** Let us assume that we are profiling execution times for reading and writing tasks of audio/video frames from/to the disk. Let us consider that the video streams and audio stream from Part 1.1. are stored using continuous placement. Let us assume that the disk head is positioned at the beginning of each stream and each stream is profiled separately. Let us assume that the disk transfer rate is 1,000,000 bytes per second and the disk block is 4 Kbytes (4096 Bytes). Note that based on the assumptions, the block access time in this example is dominated by the disk transfer time (disk rotation and seek time are negligible). Verify if the estimation of the processing times used in the Part 1.1. is realistic for

- Reading video frames of video v_1 (estimated value is 10 ms) from the disk
- Reading audio frames of audio a_1 (estimated value is 5 ms) from the disk, and
- Writing video frames of video v_2 (estimated value is 25 ms) to the disk.

Note that realistic value means in this case if your computed disk access time value using the given disk characteristics is less or equal to the estimated processing time value.

Solution: Since disk rotation and seek time are negligible, we will only consider as part of the reading/writing time from/to disk. Since the data are stored in continuous placement, the index management is minimal. Hence, the time to read/write one disk block of 4KB is equal to 4096/1,000,000 = 0.004096s, which is 4.096ms.

Now, in case of v1r we have to read frame size of 16KB, hence time to read the frame is 4*4.096 = 16.386ms. This means that the estimated time of 10ms is not realistic since computed disk access time 16.386ms is larger than the estimated processing time value of 10ms.

In case of v2r, we have to write a frame of 32 KB, hence the time to write the frame is 8*4.096 = 32.768ms. Again the estimated time of 25ms is not realistic since computed disk access time 32.768ms is larger than the estimated processing time value of 25ms.

In case of a1r, we have to read sample of 8KB, hence the time to read the sample is 2*4.096 = 8.192ms. Again, the estimated time of 5ms is not realistic since the computed disk access time of 8.192ms is larger than the estimated processing time value of 5ms.

1.3 **(4 Points)** Let us assume that the actual processing time for the display task of video v_1 fluctuates during the run time as follows: (measurement time, processing time):=
(t0, 10 ms), (t1, 8 ms), (t2, 12 ms), (t3, 10ms), (t4, 16ms), (t5, 20ms), (t6, 16ms), (t7, 20ms), (t8, 18ms), where *t1= t0+50ms; t4=t0+4*50ms; t8 = t0+8*50ms*. If we assume that the scheduling framework makes a reservation for the display task at time *t0* for 10ms, how would you change the reservation at time *t4* and *t8*? If you use the exponential average adaptation policy (with $\alpha = 0.3$), then determine the changed reservation value at time *t4*. If you use the statistical adaptation strategy (with $f = 0.4$, $ws = 5$), then determine the changed reservation value at time *t8*. Explain your steps clearly.

Solution: With the initial reservation, one needs to deploy some reservation adaptation policy. One could adjust the reservation every 50 ms as the processing times change for the display task, or one can use the exponential average adaptation policy/the statistical adaptation strategy after 'k' steps. It means, to do adaptation of reservation after each step might be too expensive, so one can specify a threshold 'k' and after 'k' times overruns happen, adaptation of reservation should be invoked.

Exponential average adaptation policy:

t1 = α*t0 + (1-α)*t1 ; t0, t1 – measurements, **t1** – new value
t1 = 0.3*10 + 0.7* 8 = 8.6ms

t2 = α***t1 +** (1-α) *t2 ; **t1** – estimated/guaranteed value; t2 – measurement; **t2** – new value
t2 = 0.3*8.6 + 0.7*12 = 10.98ms

t3 = 0.3*10.98 + 0.7*10 = 10.294ms

At time **t4** = 0.3*10.294 + 0.7*16 = 14.288ms we would need to adjust the reservation from 10ms to 14.288ms since with the reservation of 10ms for the display task this task would overrun.

Statistical Adaptation: with f=0.4 (40% can be the others within the window of 5 elements), which is two elements out of 5 (window size).

Measurements at t2 and t4 are above the other measurements in the window of 5 elements, so measurement at t2& t4 would be adjusted to reserved time of 10ms, and hence with statistical adaptation at time t4 the reservation would not change.

If the measurements stay as they are (unchanged reservation) and we take the window of t4, t5, t6, t7, t8 time measurements, then two highest values are at t5&t7 time points which would be 20ms. These two measurements would be adjusted to 18ms. t8 would stay at 18ms and the initial reservation would adjust from 10ms to 18ms.

If one slides the window, i.e., take windows (t0, t1, t2, t3, t4), (t1, t2, t3, t4, t5), (t2, t3, t4, t5, t6), (t3, t4, t5, t6, t7), (t4, t5, t6, t7, t8) and follows the earlier adjustments, then we have (t0, 10ms), (t1,8ms), (t2, 10ms), (t3,10ms), (t4,10ms), (t5,10ms), (t6, 10ms), (t7, 10ms), (t8, 10ms) at the time t8. It means all the high measurements would be adjusted to 10ms of the reserved time.

Problem 2: (25 Points)

2.1 **(13 Points)** Let us consider a single multimedia disk with number of tracks from 1 to 100. Let us consider the following set of requests in a request queue for a multimedia disk. Note that each request is represented at the disk management level as a pair of values (deadline, track number). Let us assume that the disk scheduling policy operates (makes a decision) over the whole request queue.

Order of requests →
(1, 23), (2, 57), (1, 1), (1, 89), (2, 13), (2, 78), (1, 5), (2, 75), (3, 45), (3, 98), (2, 72), (3, 47), (2, 76), (3, 97), (3, 46), (3, 95), (3, 48), (4, 2), (4, 55), (4, 6), (3, 49), (4,3).

(a) **(3 Points)** Derive the EDF schedule for these requests.

Solution: (1,23,), (1,1), (1,89), (1,5), (2,57), (2,13), (2,78), (2,75), (2,72), (2, 76), (3,45), (3,98), (3,47), (3,97), (3,46), (3,95), (3,48), (3,49), (4,2), (4,55), (4,6), (4,3);

(b) **(3 Points)** Derive the SCAN schedule for these requests. Assume that the disk head starts at track 20 and moves upwards.

Solution: (1,23), (3,45), (3,46), (3,47), (3,48), (3,49), (4,55), (2,57), (2,72), (2,75), (2,76), (2,78), (1,89), (3,95), (3,97), (3,98), (2,13), (4,6), (1,5), (4,3), (1,1);

(c) **(4 Points)** Derive the SCAN-EDF schedule for these requests using a simple perturbation function. Assume that the disk head starts at the track 20 and moves upwards.

Solution: (1.03,23), (1.69,89), (1.95, 5), (1.99, 1), (2.12, 13), (2.56, 57), (2.71, 72), (2.74, 75), (2.75, 76), (2.77, 78), (3.17, 95), (3.19, 97), (3.2, 98), (3.51, 49), (3.53, 47), (3.54, 46), (4.1, 55), (4.94, 6), (4.97, 3), (4.98, 2);

(d) **(3 Points)** Compare all three schedules in terms of number of tracks the disk head needs to make (e.g., if a head must jump from track 23 to 57, it needs to traverse 57-23=34 tracks)

Solution:

EDF: 23-1-89-5-57-13-78-75-72-76-45-98-47-97-46-95-48-49-2-55-6-3 =>
22 + 88 + 84 + 53 + 44 + 65 + 3 + 3 + 4 + 31 + 53 + 51 + 50 + 51 + 49 + 47 + 1 + 47 + 53 + 49 + 3 = 811

SCAN: 20-23-45-46-47-48-49-55-57-72-75-76-78-89-95-97-98-13-6-5-3-1 =>
3 + 22 + 1 + 1 + 1 + 1 + 6 + 2 + 15 + 3 + 1 + 2 + 11 + 6 + 2 + 1 + 2 + 87 + 7 + 1 + 2 + 2 = 179 (SCAN goes from 100 to 1 before changing directions).

SCAN-EDF: 20-23-89-5-1-13-57-72-75-76-78-95-97-98-49-47-46-55-6-3-2 =>

3 + 66 + **11** + 95 + 4 + 12 + 44 + 15 + 3 + 1 + 2 + 17 + 2 + 1 + **2**+ 51 + 2 + 1 + 45 + 54 + 45 + 96 + 3 + 1 = 531 (I have counted from 1 to 100 tracks and assumed that SCAN goes up to 100 or 1 track before changing directions).

2.2. **(12 Points)** Let us assume you are a system administrator who needs to organize a single large multimedia disk (VOD server) for a community of college students. You should take the following assumptions into account: (a) the users will be able to retrieve videos out of the VOD disk ONLY, not store videos on the disk (only you as the admin will be able to store videos on this disk), (b) 50% of users like movie V1, 20% of users like movie V2, 10% of users like movie V3, 3% of users like movie V4, 2% of users like movie V5, 2% of users like movie V6, 1% of users like movie V7, 1% of users like movie V8, and 1% of users like movie V9, (c) V1 takes 3 Mbytes storage space, V2 takes 4 Mbytes space, V3 takes 1Mbytes space, V4 takes 2 Mbytes space, V5 takes 2 Mbytes space, V6 takes 3 Mbytes space, V7 takes 3 Mbytes space, V8 takes 1 Mbytes space, and V9 takes 1 Mbytes space.

Design the multimedia disk management and specify

(1) **(2 Points)** What are the storage parameter decisions (block size, etc.) for these videos on the multimedia disk?

Solution: The storage parameters to consider are block size, placement of files, disk layout.

In our problem description, since all the movies V1,…, V9 are multiples of 1Mbyte, the block size should be 1 MByte. Since we are going to read the files only (no writes), the placement of blocks should be continuous. If we use traditional disk layout, we should use the organ placement of movie files on a single disk. If we use the ZRB disk layout, one should store the most popular movies on the outer tracks, i.e., V1, then V2, etc should start to be placed on the outer tracks.

(2) **(2 Points)** What is the file placement strategy for these videos on the multimedia disk to get the optimal file access time for each video? Identify the strategy by name and show your work.

Popularity sequence: V1 (50%), V2 (20%), V3 (10%), V4 (3%), V5 (2%), V6 (2%), V7 (1%), V8 (1%), V9 (1%) :

In case of the traditional disk layout, we need to use the organ algorithm, it means put the most popular video in the center of the disk and then less popular movies will be placed closer and closer to the edges of the disk. This means V1 will be in center, then V2 will be on the right side of V1 and V3 will be on the left side, etc. So the placement will be

V9, V7, V5, V3, V1, V2, V4, V6, V8

In case of ZRB disk layout, we place the most popular file on the outer tracks of the disk and continue to the inner of the disk in a spiral manner: V1, V2, V3, V4, V5, V6, V7, V8, V9.

(3) **(4 Points)** What services must the disk management deploy to ensure guaranteed and timely access to the files/blocks? Name at least two services that the multimedia disk

management must have. Clearly specify what the services are and what their function is, and give example(s) of condition(s) that these services may need to satisfy.

One service is disk scheduling of multimedia blocks such as SCAN-EDF, i.e., the blocks access should be time sensitive to achieve guarantees. Especially of importance will be the perturbation function to refine the timestamp based on the disk head position. Solution in 2.1.c shows one example of disk scheduling according to SCAN-EDF.

Second service is admission service, especially for the constraint placement one possible admission equation would be as follows:

$$T_{play} \geq \frac{M(\text{sectors}) + G(\text{sectors})}{r_{dt}(\text{sectors}/s)}$$

T is the time to play, 'r' is the rate of sectors per second and M/G represent sectors of data and gaps.

(4) **(4 Points)** Is the popularity of the videos described above distributed according to the Zipf's law? Justify clearly (do not specify just yes or no).

Solution:
C/1+C/2+C/3+C/4+C/5+C/6+C/7+C/8+C/9 = 1 , then C*(2520 + 1260 + 840 + 630 + 504 + 420 + 360 + 315 + 280)/2520 = C*(6129/2520), then C = 2520/6129 = 0.41116

Then the popularity to pick movie V1 is 0.41116/1 = 41%, popularity/probability to pick V2 as 2^{nd} ranked movie is 0.41116/2 = 0.20558, i.e., 20%, V3's probability to pick V3 as 3^{rd} ranked movie is 0.41116/3 = 0.137, i.e., 13.7%, V4's probability for 4^{th} ranked movie is 0.41116/4 = 0.10279, i.e., 10.2%, V5's probability for 5^{th} ranked movie is 0.41116/5 = 0.082 , i.e., 8.2%, V6: 0.41116/6 = 0.0685, i.e., 6.85%; V7: 0.0587, i.e., 5.87%; V8: 5.1395%, V9: 4.56%

As one can see, only some of the given popularity sequence: V1 (50%), V2 (20%), V3 (10%), V4 (3%), V5 (2%), V6 (2%), V7 (1%), V8 (1%), V9 (1%) correspond to the Zipf distribution. Only V2 is aligned with the Zipf distribution. All other popularity numbers are not aligned with Zipf distribution.

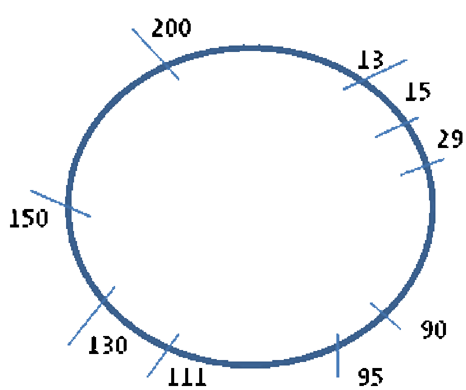
Problem 3. (25 Points)

3.1 **(4 Points)** Name four differences between Gnutella and Chord P2P systems.

Solution:

- Gnutella is an unstructured p2p system.
- Chord defines a distributed hash table.
- Peers and files are identified by keys in chord.
- Gnutella searches by flooding and Chord searches with a more efficient protocol using keys.

3.2 **(4 Points)** Assume a Chord ring where ids are truncated to 8 bits (i.e., m=8). There are currently 10 nodes in the ring with the following ids: 13, 15, 29, 90, 95, 111, 130, 133, 150, 200. Draw the resulting Chord ring (approximate scale).



3.3. **(6 Points)** Consider the Chord ring in Problem 3.2. Consider the node 95. Write down the pointers it currently has: successor, predecessor, finger table.

Solution: Successor: 111; Predecessor: 90,

Finger table:

0	111

1	111
2	111
3	111
4	111
5	130
6	200
7	13

3.4.(**3 Points**) Consider the Chord ring in Problem 3.2. If the node 95 wants to introduce a new file with id 31, what would be the first hop this introduction of the new file would take? Which node would finally store the file?

Solution: First hop: node 13
File with id 31 will be stored at node 90.

3.5 (**3 Points**) Consider the Chord ring in Problem 3.2. If the node 130 leaves, what would be the new finger table at node 95?

Solution: The fifth entry will change to point to the node with id 133.

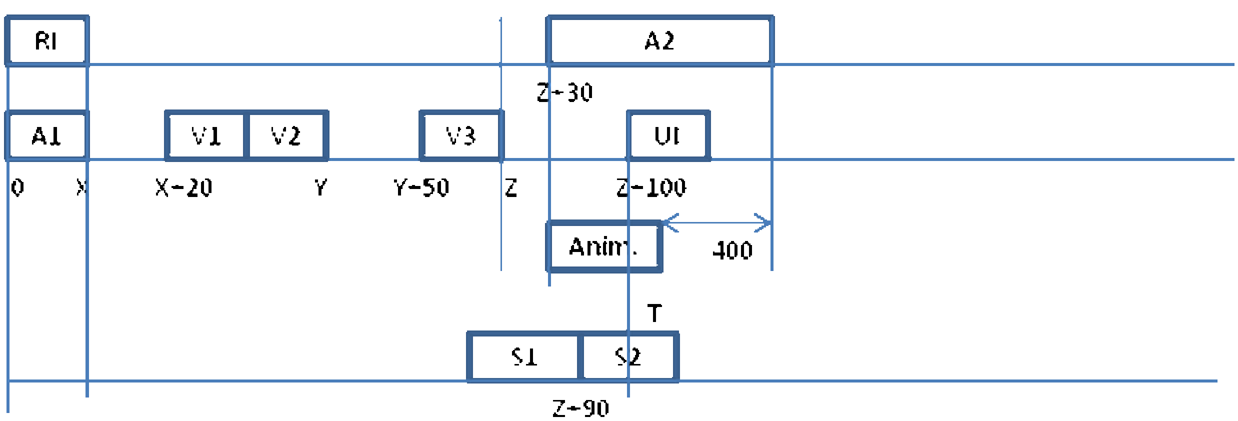
Problem 4. (20 Points)

4.1 Consider the following Time-Interval Specification:

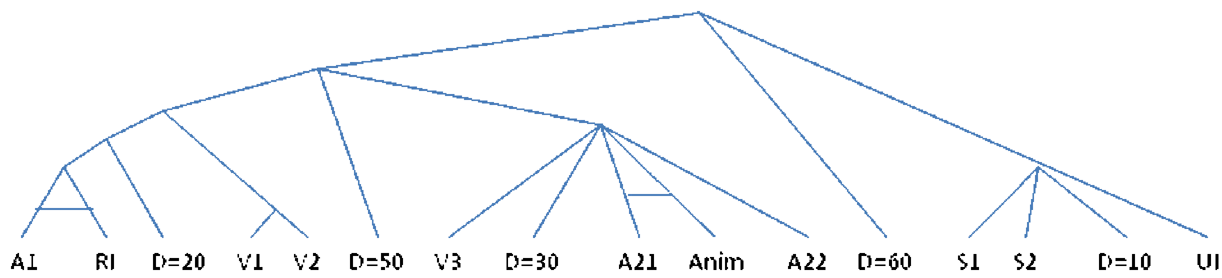
Audio1 while(0,0) RecordedInteractions
Audio1 before(20) Video 1;
Video1 before(0) Video2;
Video2 before(50) Video3;
Video3 before(100) UserInteraction;
Video3 before(30) Audio2;
Audio2 while(0,400) Animation;
Slide1 before(10) UserInteraction;
Slide1 before(0) Slide2;

Transform the interval-based specification into

(a) **(4 Points)** Time-axis specification



(b) **(4 Points)** Basic Hierarchical specification



(c) **(6 Points)** Compare all three specifications (Interval-based, Time-axis and Hierarchical Specifications) with respect to this example in terms of their strengths and weaknesses. Point out two strength and two weaknesses for each specification.

4.2 **(6 Points)** Consider the Video-on-Demand (VOD) Service, where the VOD client is watching a movie, streamed from the VOD server with audio stream (*Audio1*), and video stream (*Video1*), at the time point *T* (from the start of the movie *T* time units elapsed). After the time point *T* the client is receiving *Audio1*, but it is not receiving the corresponding video (*Video1*), i.e., somewhere along the end-to-end path blocking of the video stream happened. At the client side, the VOD system faces so called **gap problem**. Provide two different solutions to solve the gap problem. Describe clearly what action, algorithm, and/or protocol you would execute to minimize or remove the gap problem.

Gap problem occurs if starvation is present, i.e., video frames are not coming.

One solution could be to repeat the existing video until the new video frames arrive. In this case, if the threshold of the buffering indicates starvation of video, the protocol should send a feedback to the VOD server and ask for increase of frame rate for video.

Second solution is to have a strong admission control to allocate sufficient resources that the gap problem does not occur, or has a low probability to occur. In case of gap, one could show an advertising video clip to fill the gap, and then when video frames arrive, continue with the playback. Using other locally stored video clip is usually used at the start time when selected video frames need to be prefetched.

Problem 5. (10 Points)

5.1 **(3 Points)** List three differences between IP multicast and Overlay/P2P multicast.

- IP multicast is the layer 3 protocol, P2P overlay multicast is layer 5 protocol;
- IP multicast is more efficient implementation of multicast than P2P overlay multicast because there is less background overhead.
- IP Multicast is harder to deploy than P2P overlay multicast because one must make changes in routers which requires third party admin changes.

5.2 **(3 Points)** What's the main advantage of streaming using a multiple tree approach instead of a single tree approach?

Solution: Special coding techniques can be used, like MDC. This will allow one description to be propagated per tree. When there are tree failures, the multimedia stream will gracefully degrade (from the client perspective) instead of losing parts of the stream.
Another acceptable answer is: Multiple trees make better use of the peer's bandwidth, because in a single tree the leaves only receive data but do not forward the stream.

5.3 **(4 Points)** Mention four differences between the mesh and tree based approach to streaming.

- Trees require finding a parent, in the mesh approach you can potentially connect to any peer.
- In a mesh swarming is used to propagate the data.
- Mesh uses packet scheduling.
- Mesh has two propagation phases, but trees data just flows down the tree.