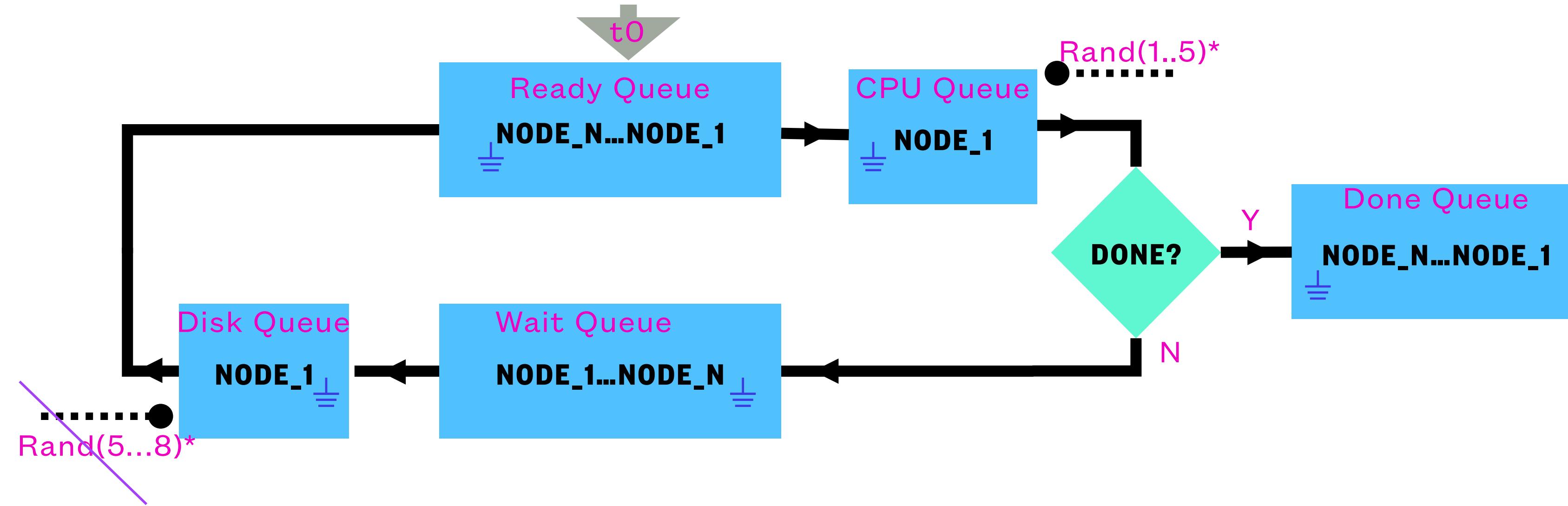




cosc 514 Group Project

SOPHIE JOHNSON-SHAPOVAL, BASIL HUFFMAN, FEKADU URGA. APRIL 29, 2021

Design Overview



Initially, processes are placed in Ready Queue and moved through the pipeline with the constraint of 1 CPU and 1 DISK reading in lines in a file until Done Done when read in all 100 lines for all 10 processes
 Each file consists of 100 lines and 10 char per line
 File in a line consists of {token(pid) followed by lineNumber } x repeat
 CPU time used steady_clock()
 The five queues are implemented as linked lists (append @ end; remove @ front)
 A delay in the CPU stage (rand() injection) was implemented

PCB:

- Pid
- State
- PC
- Ptr to proc. mem. head
- Ptr to next PCB

States:

- 1-in Read
- 2-in CPU
- 3-in Wait
- 4-in Disk
- 5-in Done

Memory Hash Table:

- Pid-1 is the index in table
- Contains Ptr's to proc. mem. heads
- 1 Frame=8 Pages

Proc. Memory Node:

- Pid
- Frame Number
- Contents of the read in frames
- Start/End Lines in File
- CPU time
- Ptr to next Node

Code / Implementation

- Structures: Memory node (used in memory hash table) and PCB node (used in the five queue lists)
- Functions: add and delete to/from PCB list; print a PCB node; print a memory node; size of a PBC list; initialize hash table; create files; read n lines from line x to line y; print state table;
- Main(): init the hash table; create the 10 PCBs and assign them to ready queue; start the clock and enter the pipeline (a while loop that ends when the done list is of size 10). At the beginning of each pipeline stage loop, grab a timestamp and move nodes among the queues; integrate print statements; when done, print the contents of the memory nodes.
- Code revision and improvements include:
 - refactored queues to a double ended queue to allow for O(1) insertion/deletion on both ends;
 - refactored hash table using STL;
 - refactored such that structs are responsible for their own output;
 - refactored character logic to casting;
 - removed superfluous code;
 - linted code and refactored into proper style for time savings.

Pipeline without delay in CPU

ClockStamp	Ready	CPU	Wait	Disk	Done
0	1->2->3->4->5->6->7->8->9->10	E	E	E	E
9.833e-06	2->3->4->5->6->7->8->9->10	1	E	E	E
1.6667e-05	3->4->5->6->7->8->9->10	2	1	E	E
2.6375e-05	4->5->6->7->8->9->10	3	2	1	E
9.4875e-05	5->6->7->8->9->10->1	4	3	2	E
0.000144583	6->7->8->9->10->1->2	5	4	3	E
0.000193	7->8->9->10->1->2->3	6	5	4	E
0.000241792	8->9->10->1->2->3->4	7	6	5	E
0.000291667	9->10->1->2->3->4->5	8	7	6	E
0.000339708	10->1->2->3->4->5->6	9	8	7	E
0.000385708	1->2->3->4->5->6->7	10	9	8	E
0.00043225	2->3->4->5->6->7->8	1	10	9	E
0.000478458	3->4->5->6->7->8->9	2	1	10	E
0.000524667	4->5->6->7->8->9->10	3	2	1	E
0.000569042	5->6->7->8->9->10->1	4	3	2	E
0.000613292	6->7->8->9->10->1->2	5	4	3	E
0.00336496	7->8->9->10->1->2->3	6	5	4	E
0.00345525	8->9->10->1->2->3->4	7	6	5	E
... image continues.					
0.0088771	2->3->4->5->6->7->8->9	1	10	9	E
0.00894646	3->4->5->6->7->8->9	2	E	10	1
0.00899283	4->5->6->7->8->9->10	3	E	E	1->2
0.00899842	5->6->7->8->9->10	4	E	E	1->2->3
0.009004	6->7->8->9->10	5	E	E	1->2->3->4
0.00900921	7->8->9->10	6	E	E	1->2->3->4->5
0.00901437	8->9->10	7	E	E	1->2->3->4->5->6
0.00901992	9->10	8	E	E	1->2->3->4->5->6->7
0.00902546	10	9	E	E	1->2->3->4->5->6->7->8
0.00903108	E	10	E	E	1->2->3->4->5->6->7->8->9
0.00903675	E	E	E	E	1->2->3->4->5->6->7->8->9->10

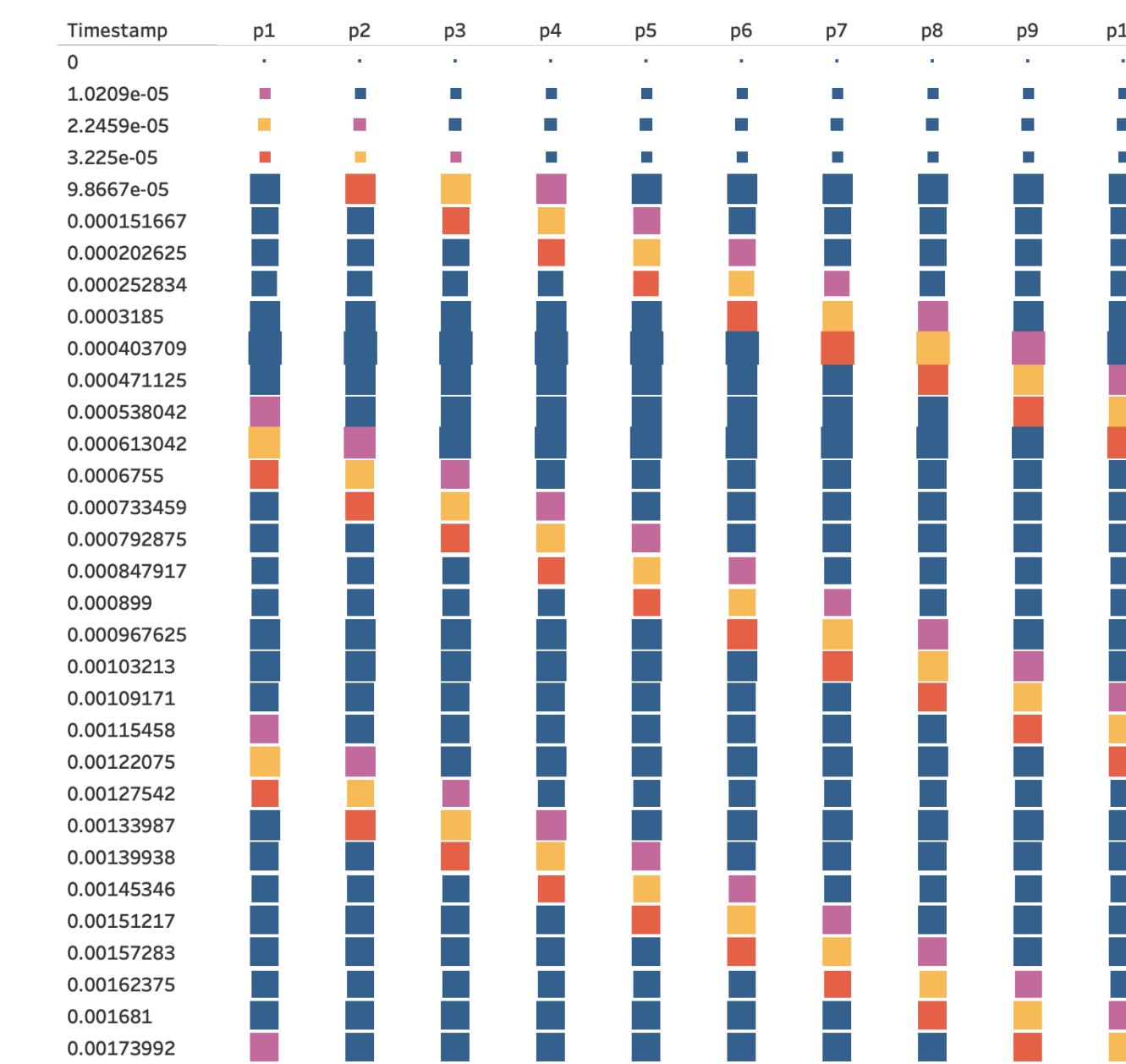
Pipeline Run without a delay in the CPU stage

```

pid: 1 pageNum 2 contains A17A17A17AA18A18AA19A19AA20A20AA21A21AA22A22AA23A23AA24A24A24A in lines 17-24 read at 0.00374662
pid: 1 pageNum 3 contains A25A25A25AA26A26AA27A27AA28A28AA29A29AA30A30AA31A31AA32A32A32A in lines 25-32 read at 0.00421196
pid: 1 pageNum 4 contains A33A33A33AA34A34A34AA35A35AA36A36AA37A37AA38A38AA39A39AA40A40A40A in lines 33-40 read at 0.00467421
pid: 1 pageNum 5 contains A41A41A41AA42A42AA43A43AA44A44AA45A45AA46A46AA47A47AA48A48A48A in lines 41-48 read at 0.00522242
pid: 1 pageNum 6 contains A49A49A49AA50A50AA51A51AA52A52AA53A53AA54A54AA55A55AA56A56A in lines 49-56 read at 0.00568937
pid: 1 pageNum 7 contains A57A57A57AA58A58AA59A59AA60A60AA61A61AA62A62AA63A63AA64A64A64A in lines 57-64 read at 0.00614983

```

Processes' progression in the pipeline at given time stamps with the duration of each slice shown as a relative size



Pipeline with delay in CPU

ClockStamp	Ready	CPU	Wait
0	1->2->3->4->5->6->7->8->9->10	E	E
1.6875e-05	2->3->4->5->6->7->8->9->10	1	E
2.8667e-05	2->3->4->5->6->7->8->9->10	1	E
3.6417e-05	3->4->5->6->7->8->9->10	2	1
4.3917e-05	3->4->5->6->7->8->9->10	2	E
0.000108709	3->4->5->6->7->8->9->10->1	2	E
0.000115417	4->5->6->7->8->9->10->1	3	2
0.000122792	4->5->6->7->8->9->10->1	3	E
0.000188209	5->6->7->8->9->10->1->2	4	3
0.000196375	5->6->7->8->9->10->1->2	4	E
0.000257792	6->7->8->9->10->1->2->3	5	4
0.000265209	6->7->8->9->10->1->2->3	5	E
0.00033	7->8->9->10->1->2->3->4	6	5
0.0003375	8->9->10->1->2->3->4	7	6
0.000394875	8->9->10->1->2->3->4->5	7	E
0.000505667	9->10->1->2->3->4->5->6	8	7
0.000513125	9->10->1->2->3->4->5->6	8	E
0.00057325	10->1->2->3->4->5->6->7	9	8
0.00058075	10->1->2->3->4->5->6->7	9	E
0.000632334	10->1->2->3->4->5->6->7->8	9	E
0.000639292	10->1->2->3->4->5->6->7->8	9	E
0.000645625	1->2->3->4->5->6->7->8	10	9
0.000652792	1->2->3->4->5->6->7->8	10	E
0.000712375	1->2->3->4->5->6->7->8->9	10	E
0.000712375	1->2->3->4->5->6->7->8->9	10	E

... image continues

0.0131254	9->10	8	E
0.0131314	9->10	8	E
0.0131372	10	9	E
0.0131445	10	9	E
0.0131506	10	9	E
0.0131565	E	10	E
0.0131634	E	E	E

Pipeline Run with a delay in the CPU stage

The contents of the memory for each process-----

```
pid: 1 PageNum 0 contains A1A1A1A1A1A2A2A2A2A3A3A3A4A4A4A4A5A5A5A6A6A6A6A7A7A7A8A8A8A8 in lines 1-8 read at
4.3917e-05 Addr:0x1058714a8=4387706024
pid: 1 PageNum 1 contains A9A9A9A9A10A10AA11A11AA12A12A12AA13A13A13AA14A14A14AA15A15AA16A16A in lines 9-16 read at
0.000827417 Addr:0x100759808=4302673928
pid: 1 PageNum 2 contains A17A17A17AA18A18AA19A19AA20A20AA21A21AA22A22AA23A23AA24A24A24A in lines 17-24 read at
0.00163492 Addr:0x105871768=4387706728
pid: 1 PageNum 3 contains A25A25A25AA26A26AA27A27AA28A28AA29A29AA30A30AA31A31AA32A32AA32A in lines 25-32 read at
0.00241867 Addr:0x1058719e8=4387707368
```

...etc

SOPHIE JOHNSON-SHAPOVAL, BASIL HUFFMAN, FEKADU URGA. APRIL 29, 2021

Disk	Done	CPU delay:
E	E	2
E	E	3
1	E	2
2	E	2
3	E	2
4	E	1
5	E	2
6	E	2
7	E	4
8	E	4
9	E	4

CPU delay: 1

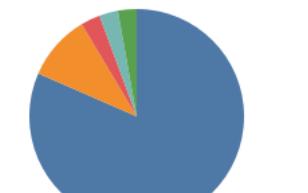
Processes' progression in the pipeline at a given time with the duration of each pipeline slice shown as relative sizes (rand() injection @ CPU state)



States:
1-in Read
2-in CPU
3-in Wait
4-in Disk
5-in Done

... image continues

Avg time per state (delayed)



Avg time per state (delayed)

State	1	2	3	4	5
	0.01456	0.00177	0.00052	0.00050	0.00050

Sum of AVG broken down by State.

State (color) and sum of AVG (size

Challenges, Benefits and Other Thoughts

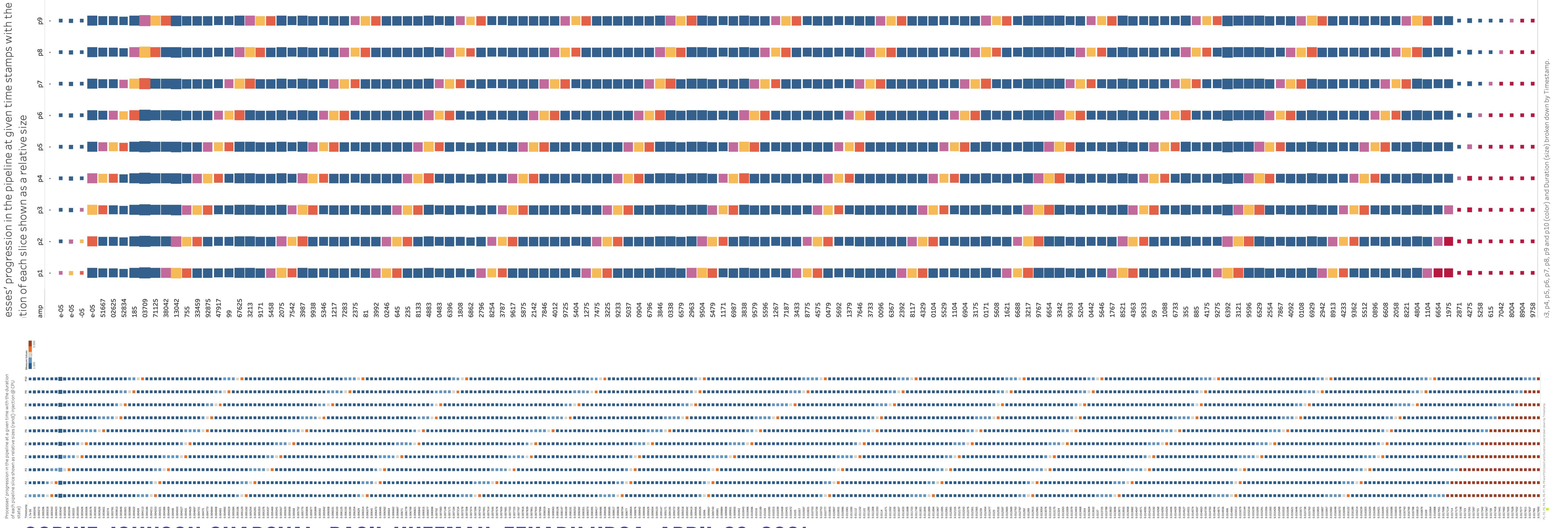
- Challenges:
 - Time constraints (every real world project has this constraint too!);
 - Need to coordinate within the team (every big real world project has this constraint too!);
 - Working students with family have less flexibility (mostly working on the project at night)
- Benefits:
 - Diverse skillset of the team members allowed us to split the work
 - Frequent communications: weekly meetings, frequent updates via email, updates on the GitHub page (https://github.com/thisisbasil/cosc514_project2) and participation in the groupchat
- Fun next steps (if had more time):
 - Implement the rand() injection in the Disk Queue and start processes spread out in time, i.e., not all at once

Code DEMO

- Live Demo

Back Up: Pipelines w/ and w/o delay in CPU (full view)

Data was post-processed in Tableau



SOPHIE JOHNSON-SHAPOVAL, BASIL HUFFMAN, FEKADU URGA. APRIL 29, 2021