Space-based faults

Daniel Krolopp | Aaron Neustedter

D

Motivation

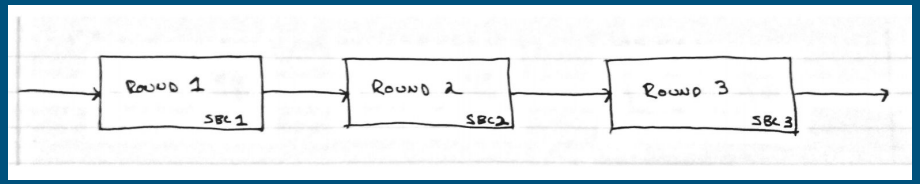
- Renewed global interest in space exploration
  - NASA Artemis
  - SpaceX Starship
  - Blue Origin Blue Moon

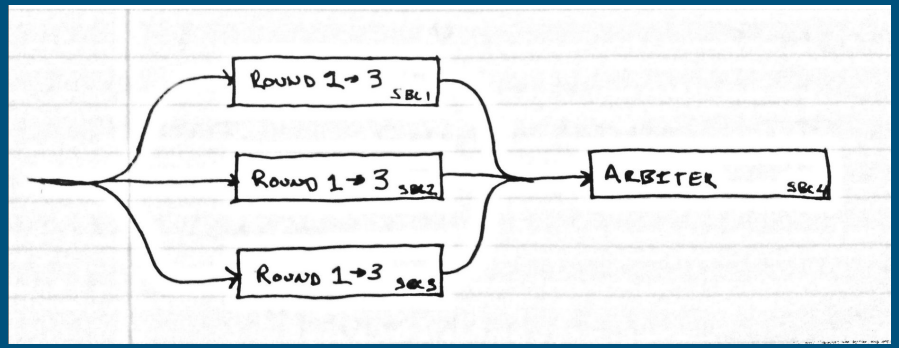
A

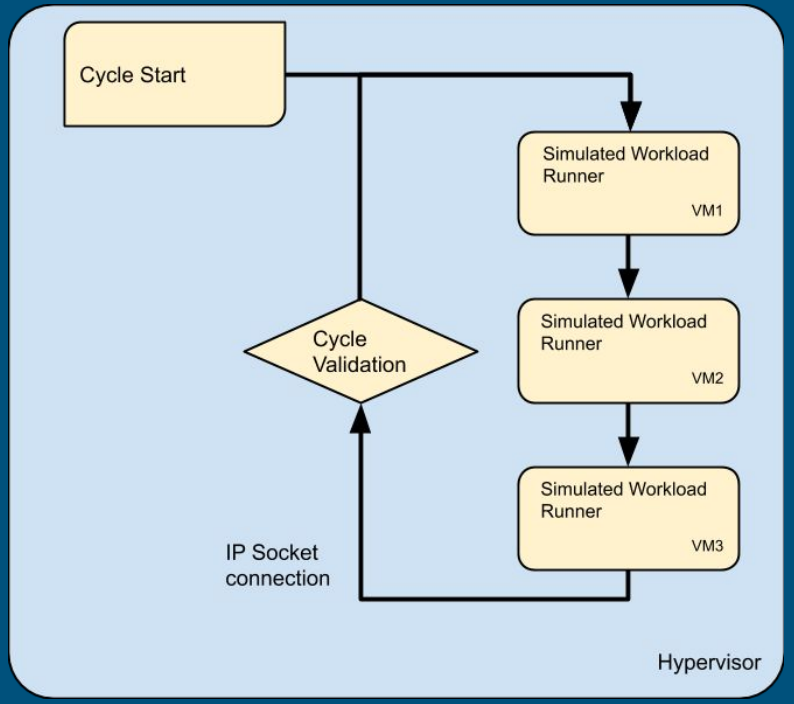
Test Plan

- Create a ecosystem of machines working together
- Inject errors into machines and observe error propagation
- Create different combinations of machines
- Evaluate the robustness of each combination, propagation of errors, etc

System Overview







A

Test Stand Architecture

Test Stand Data Flow

System Details

- `xinu00.cs.purdue.edu`
- `cs-console` managing 185 Beaglebone Black ARM SBCs

```
# aneusted @ xinu17 in ~/ftd/SpaceWhale/Xinu-code-BeagleBoneBlack/compile on git:master x [22:46:19]
$ cs-status | head -n 5 && echo '...' && cs-status | tail -n 30
xinuserver:
    beagle101       cortex          user=                   time=
    beagle102       cortex          user= hmonga            time= 484:41:13
    beagle103       cortex          user= dhar1             time= 151:52:29
    beagle104       cortex          user= cenrigh           time= 199:59:34
...
    beagle155       cortex          user=                   time=
    beagle156       cortex          user=                   time=
    beagle157       cortex          user=                   time=
    beagle158       cortex          user=                   time=
    beagle159       cortex          user=                   time=
    beagle160       cortex          user= dkrolopp          time= 00:00:07
    beagle161       cortex          user= dkrolopp          time= 01:10:45
    beagle162       cortex          user= dkrolopp          time= 01:10:26
    beagle163       cortex          user=                   time=
    beagle164       cortex          user=                   time=
```

A

Machine Details

- BeagleBone Black SBCs running modified XINU operating system (ARM)
- Responsible for computing simulated workload and passing result to next SBC

A

Why Xinu?

- A small, embedded system much like those found on spacecraft
- Small memory footprint
- Easy fault instrumentation
- Familiarity

```
The text segment ranges from 0x81000000 to 0x810160a4 (Total size: 22569)
The data segment ranges from 0x81017000 to 0x81017868 (Total size: 538)
The BSS  segment ranges from 0x81018000 to 0x810383e4 (Total size: 33017)
```

A

Simulated Workload Details

- Our simulated workload repeatedly sums numbers from an array in memory
  - A 4KB "workspace" of ints along with kernel memory serve as a target for faults
  - Virtually every fault in our workspace will cascade into a failure
  - Each round of summations is called an "iteration"
- Faults are instrumented programmatically from within the application
  - In our previous presentation, faults had a probability ***p*** of flipping a single bit after every iteration through our algorithm
  - With current design, a background process randomly schedules memory corruption events according to a uniform distribution

D

Faulty run results

```
finished test...
Unha~dlad exceptio.. Lilk g|#ister: 0x||#

panic: **** ECEPPION ****
```

```
Unhandhad e8seption. \iji!Regaster:`0h!x

panic: **** EXcEH\ION ***.
```

```
finished test...
Unhandled exception. Link Register: 0x81007928

panic: **** EXCEPTION ****
```

```
Received! Processing 100 iterations with input 1411548400...
Processing complete. Final result: -1895994112.
Wait time: 52611, compute time: 101807
TEXT  flips: 65
DATA  flips: 1
BSS   flips: 85
OTHER flips: 1
finished test...
```

D

```
Xinu for bbb -- version #44  (aneusted)  Thu Dec 12 23:12:16 EST 2019

Ethernet Link is Up. Speed is 100Mbps
Link is Full Duplex
MAC Address is: 6C:EC:EB:AC:58:EA
 519731552 bytes of free memory.  Free list:
           [0x810526A0 to 0x9FFF9FFF]
     89900 bytes of Xinu code.
           [0x81000000 to 0x81015F2B]
    136132 bytes of data.
           [0x81016000 to 0x810373C3]

Obtained IP address  128.10.137.160   (0x800a89a0)

The text segment ranges from 0x81000000 to 0x81015f2c (Total size: 22475)
The data segment ranges from 0x81016000 to 0x81016868 (Total size: 538)
The BSS  segment ranges from 0x81017000 to 0x810373c4 (Total size: 33009)
starting test...
Processing 100 iterations with input 314...
Processing complete. Sending result: -133840484...
Result sent!
Wait time: 33, compute time: 25483
TEXT  flips: 0
DATA  flips: 0
BSS   flips: 2
OTHER flips: 0
finished test...


(command-mode) _
```

```
Xinu for bbb -- version #45  (aneusted)  Thu Dec 12 23:18:25 EST 2019

Ethernet Link is Up. Speed is 100Mbps
Link is Full Duplex
MAC Address is: 6C:EC:EB:BA:1A:B7
 519731552 bytes of free memory.  Free list:
           [0x810526A0 to 0x9FFF9FFF]
     89896 bytes of Xinu code.
           [0x81000000 to 0x81015F27]
    136132 bytes of data.
           [0x81016000 to 0x810373C3]

Obtained IP address  128.10.137.161   (0x800a89a1)

The text segment ranges from 0x81000000 to 0x81015f28 (Total size: 22474)
The data segment ranges from 0x81016000 to 0x81016868 (Total size: 538)
The BSS  segment ranges from 0x81017000 to 0x810373c4 (Total size: 33009)
starting test...
Waiting on input...
Received! -133840484 Processing 100 iterations with input -133840484...
Processing complete. Sending result: 1411548400...
Result sent!
Wait time: 26543, compute time: 25548
TEXT  flips: 23
DATA  flips: 0
BSS   flips: 30
OTHER flips: 0
finished test...
```

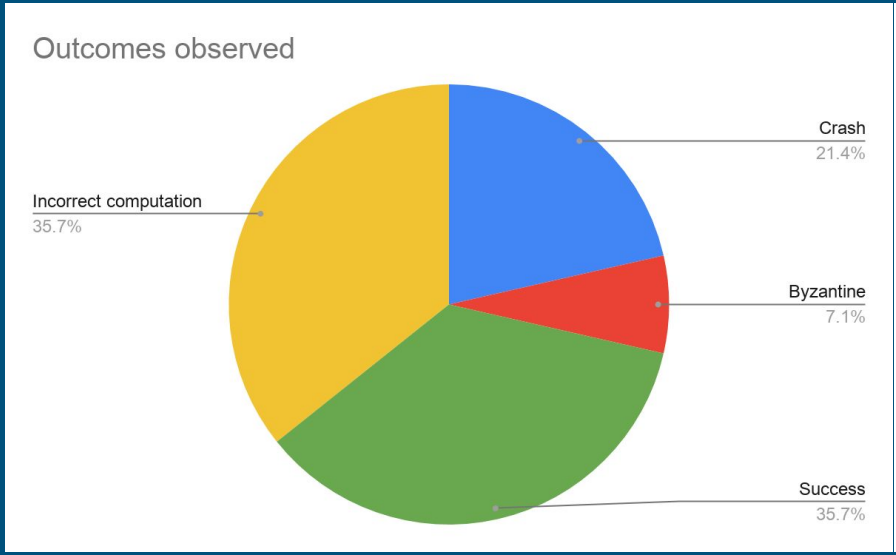
```
Xinu for bbb -- version #46  (aneusted)  Thu Dec 12 23:18:44 EST 2019

Ethernet Link is Up. Speed is 100Mbps
Link is Full Duplex
MAC Address is: 6C:EC:EB:AB:07:8B
 519731552 bytes of free memory.  Free list:
           [0x810526A0 to 0x9FFF9FFF]
     89896 bytes of Xinu code.
           [0x81000000 to 0x81015F27]
    136132 bytes of data.
           [0x81016000 to 0x810373C3]

Obtained IP address  128.10.137.162   (0x800a89a2)

The text segment ranges from 0x81000000 to 0x81015f28 (Total size: 22474)
The data segment ranges from 0x81016000 to 0x81016868 (Total size: 538)
The BSS  segment ranges from 0x81017000 to 0x810373c4 (Total size: 33009)
starting test...
Waiting on input...
Received! Processing 100 iterations with input 1411548400...
Processing complete. Final result: -1895994112.
Wait time: 52611, compute time: 101807
TEXT  flips: 65
DATA  flips: 1
BSS   flips: 85
OTHER flips: 1
finished test...
```

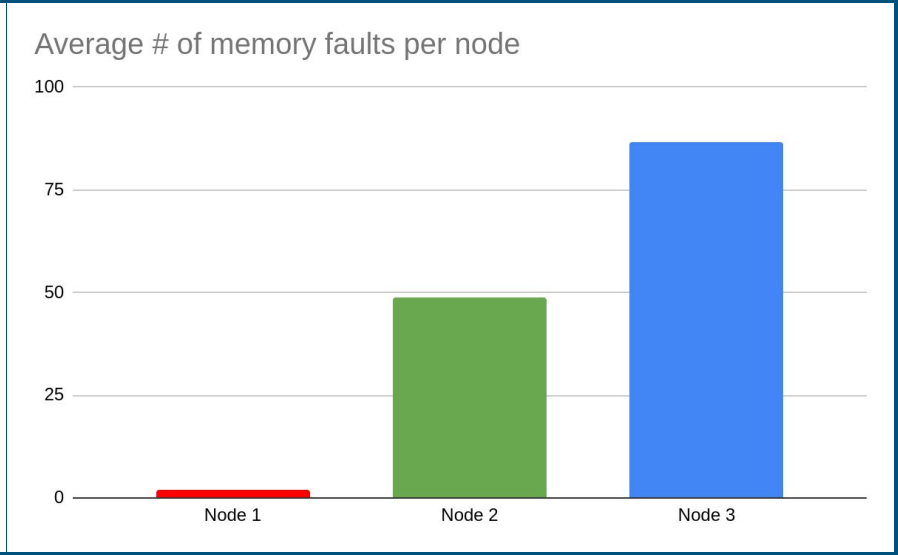
Data Collected

- Instrumented which node failures occurred, general cause, and errors injected in that node

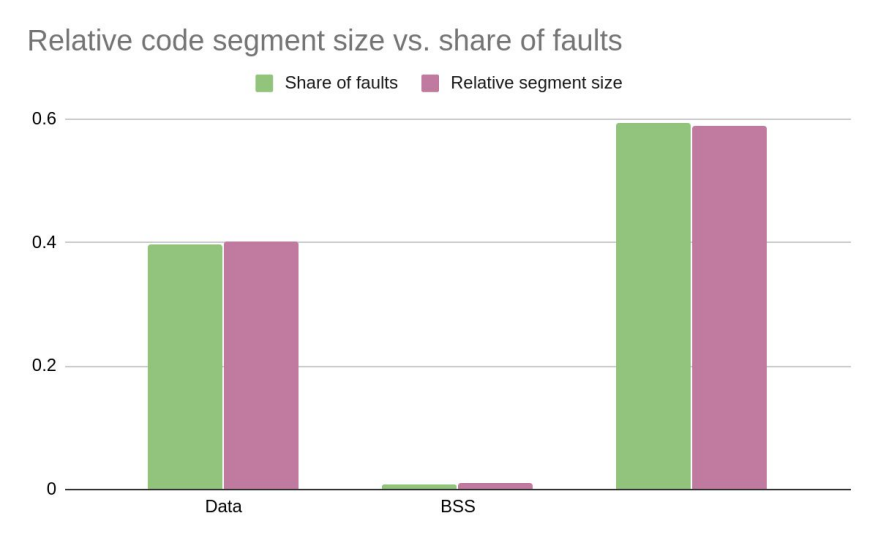
O	P	Q	R	S	T	U	V	W	X	Y	Z
# TEXT bit flips (2)	# DATA bit flips (2)	# BSS bit flips (2)	# OTHER bit flips (2)	Total (2)	# TEXT bit flips (3)	# DATA bit flips (3)	# BSS bit flips (3)	# OTHER bit flips (3)	Total (3)	Outcome	Notes
0	0	0	0	0	0	0	0	0	0	Success	
0	0	2	0	2	-	-	-	-	0	Failure	Crash in node 2
17	1	35	0	53	-	-	-	-	0	Failure	Crash in node 3
21	0	30	1	52	48	1	55	1	105	Failure	Sent correct message 2 -> 3 but arrived differe
20	0	30	0	50	27	0	49	0	76	Success	
20	0	32	1	53	27	1	53	0	81	Success	
24	2	31	2	59	27	2	43	2	74	Success	
18	0	30	1	49	28	1	43	2	74	Failure	Incorrect computation happened in node 2
27	0	28	1	56	28	1	51	0	80	Success	
23	0	30	0	53	65	1	85	1	152	Failure	Node 4 took 4x as long as expected
15	0	33	2	50	31	1	53	0	85	Failure	Incorrect computation happened in node 3
22	0	26	2	50	32	0	42	3	77	Success	
20	0	20	1	41	26	0	42	1	69	Failure	Incorrect computation happened in node 2
25	0	32	0	57	-	-	-	-	0	Failure	Crash in node 3
21	0	35	0	56	30		48	2	80	Failure	Incorrect computation happened in node 2
0	0	0	0	0	-	-	-	-	0	Failure	Crash in node 2 during message sending
18.200	0.200	26.267	0.733	45.400	33.545	0.800	51.273	1.091	86.709		
40.09%	0.44%	57.86%	1.62%	100.00%	38.69%	0.92%	59.13%	1.26%	100.00%		

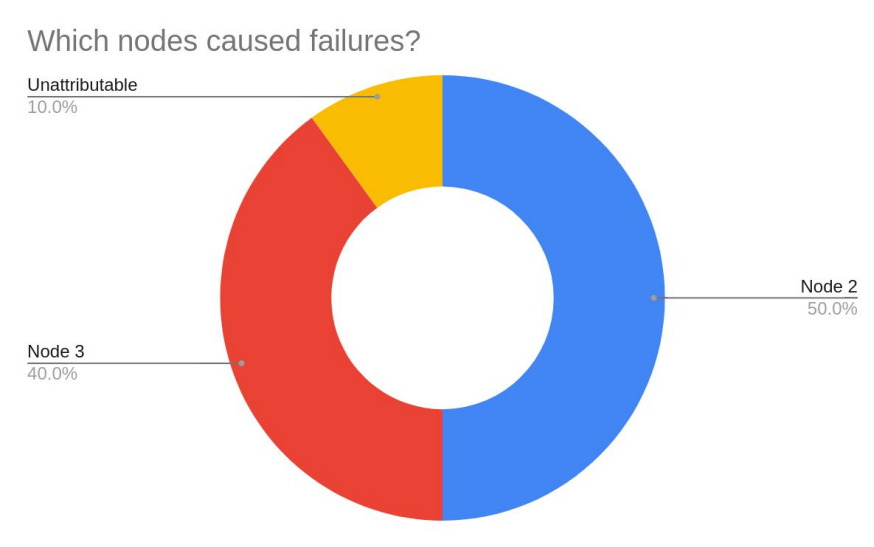
Analysis - Linear configuration



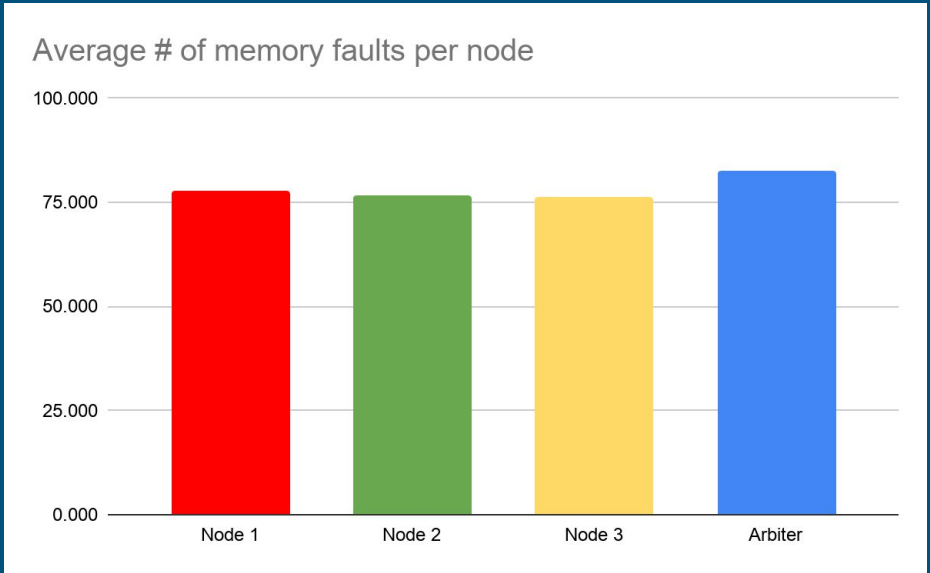


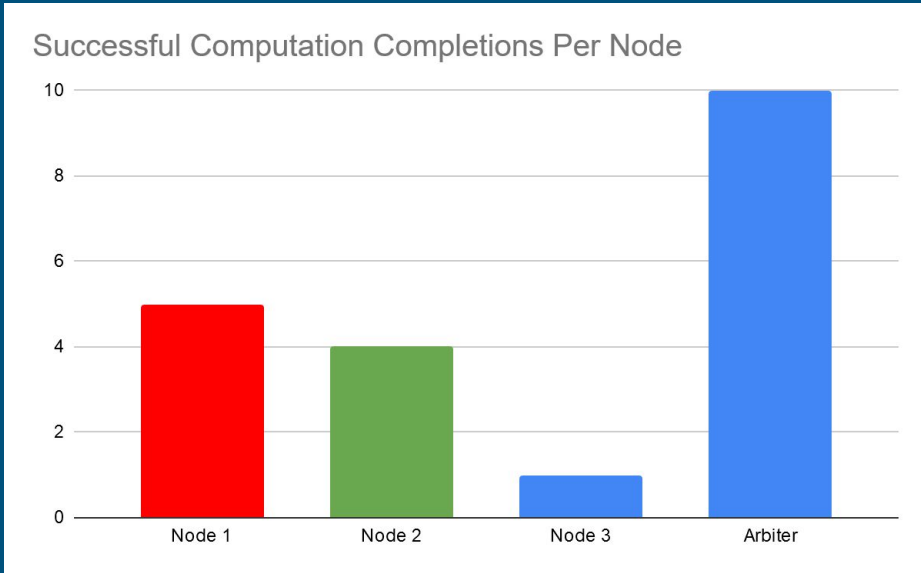
Analysis - Linear configuration



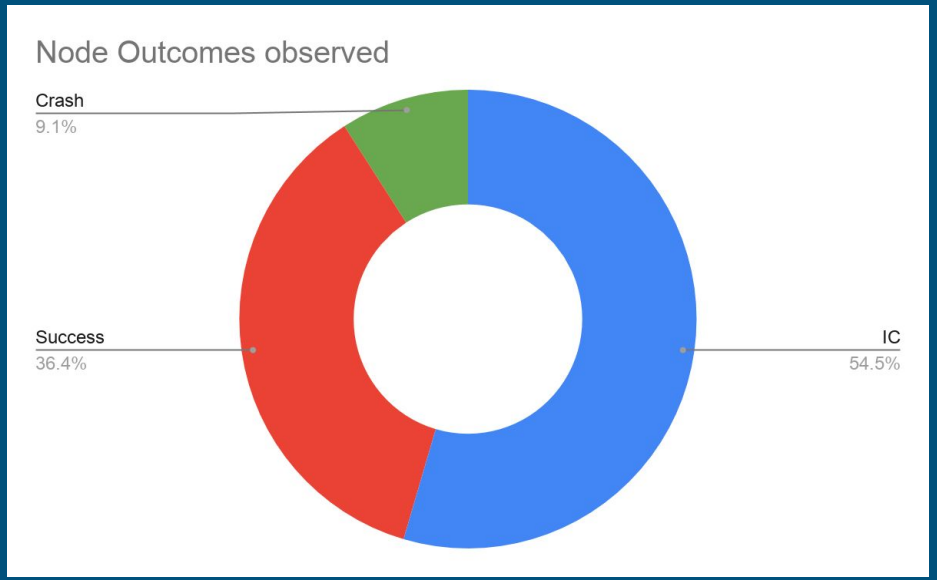


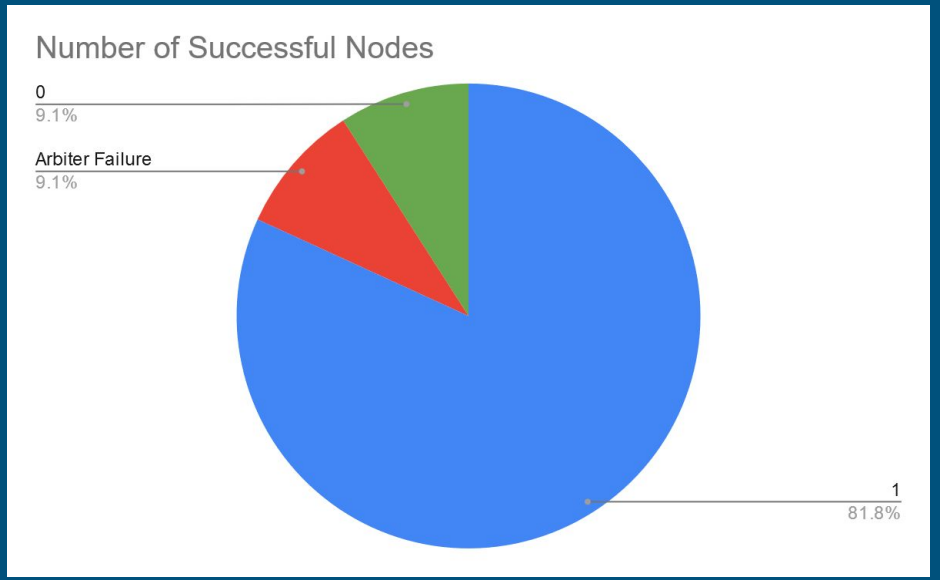
Analysis - 2-of-3 Voting scheme





Analysis - 2-of-3 Voting scheme





Analysis - 2-of-3 Voting scheme

- The voting scheme actually performs significantly worse than the linear configuration
- At least 2 voters must perform correctly. The arbiter must also perform correctly
- As Prof. Bagchi put it, you're actually weaker than your weakest link

$$\binom{3}{2}(\frac{1}{3})^2 \cdot (1-\frac{1}{3}) \cdot (\frac{1}{3}) = 0.074$$

Issues

- Some data suspiciously repeats itself
  - Maybe RAM isn't being cleared as quickly as we expect it to be
- Lack of automation lead to limited data

Future Work

- Further analysis of collected data
- Additional testbed configurations
- Additional rates of fault injection
- Improvements in scalability and automation

Conclusion

- Even using a OS with a miniscule attack surface, operation is surprisingly robust
- Use caution before blindly adding “fault tolerate schemes” to failure prone systems