Mining Supercomputer Logs to Identify Correlation

ADITYA PAKKI* and HARSHITHA PARNANDI*
[1] Group 28 - University of Utah, School of Computing.

The Intrepid Supercomputer.
Courtesy by https://www.alcf.anl.gov/

Problem:

- Simulations on next generation Supercomputers and Exascale machines (2019-2022 time frame) expected to fail every 1-2 hours.
- Data written to disks is enormous and single component failures are expensive.
- Fault detection and recovery cheaper than standard checkpoint and recovery.
- Goal is to be proactive than reactive.

Data:

- Error and job logs of Intrepid machine 01/05/09 to 08/31/09.
- Free flowing text with data stored in below format, we parsed data.

```
9
10   26124008 KERN_080A  KERNEL          _bgp_unit_ddr       _bgp_err_ddr_SSE_count              WARN     2009-01-05-00.35.53.
868598 -                    0             - ANL-R02-R03-2048                R03-M1-N15-J04
        44V3575YL12M7270WJD x'02402C600839D80F0C0E576748A4' DDR controller 0, chipselect 0 single symbol error count 9
11
12   26124021 KERN_080A  KERNEL          _bgp_unit_ddr       _bgp_err_ddr_SSE_count              WARN     2009-01-05-00.41.28.
753741 -                    0             - ANL-R45-M0-S12                  R45-M0-N09-J13
        44V3575YL12M7346J9N x'02405004BF70F5180F0667848AA5' DDR controller 0, chipselect 0 single symbol error count 48222
```

Data Pre-processing:

- Load data to MySQL tables, identify 'FATAL' error types and its subtypes.
- Template the error messages to narrow down the possible events, see below figure.

Examples of messages template to a fixed pattern.

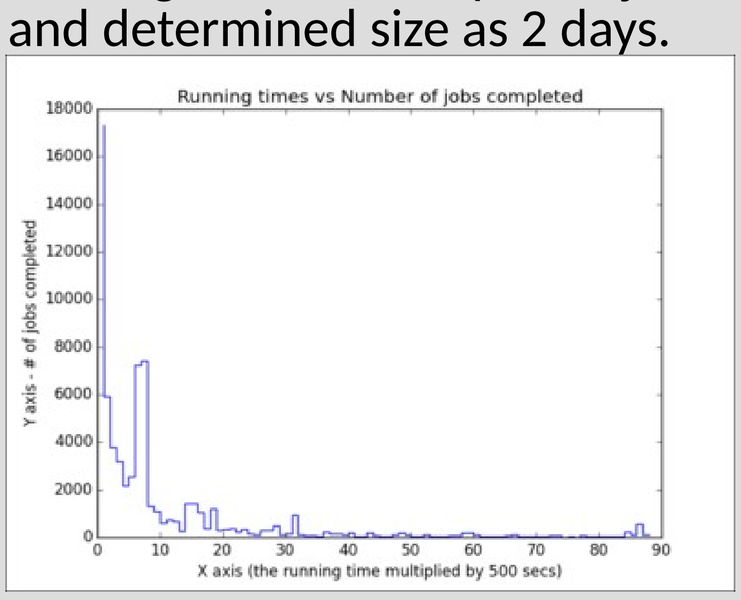
- DDR controller 0, chipselect 0 single symbol error count
- DDR controller 1, chipselect 1 single symbol error count

will be replaced by "DDR controller **d+**, chipselect **d+** single symbol error count"

- BPC pin JK126, transfer 1, bit 106, BPC module pin V03, compute trace MEMORY1 DATA 79, DRAM chip U13, DRAM pin B9.

will be replaced by "BPC pin **V***, transfer **d+**, bit **d+**, BPC module pin **V***, compute trace DATA **d+**, DRAM chip **V***, DRAM pin **V***".

- Identify a window size to perform event correlation. Measured running times of completed jobs and determined size as 2 days.

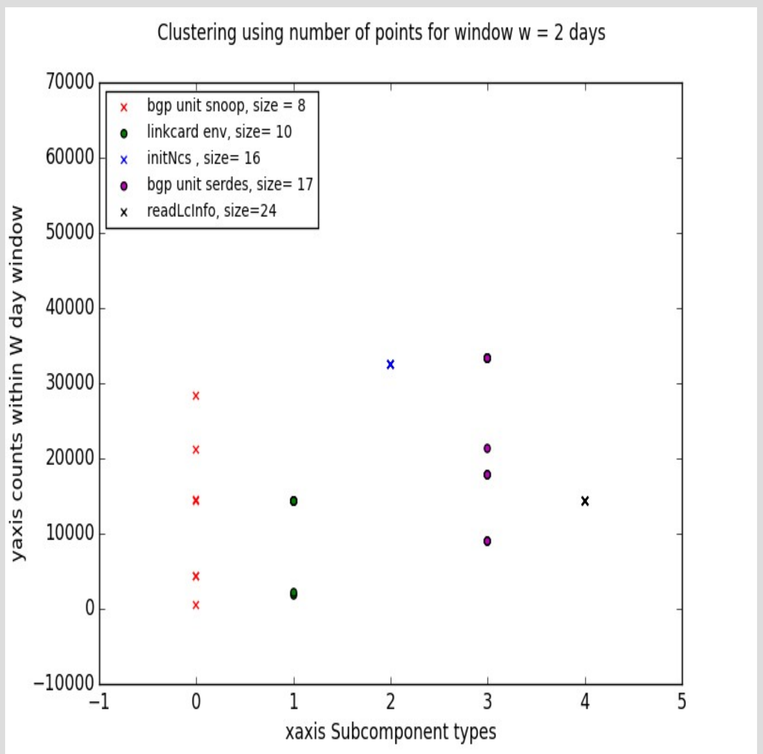


Key Ideas and Techniques:

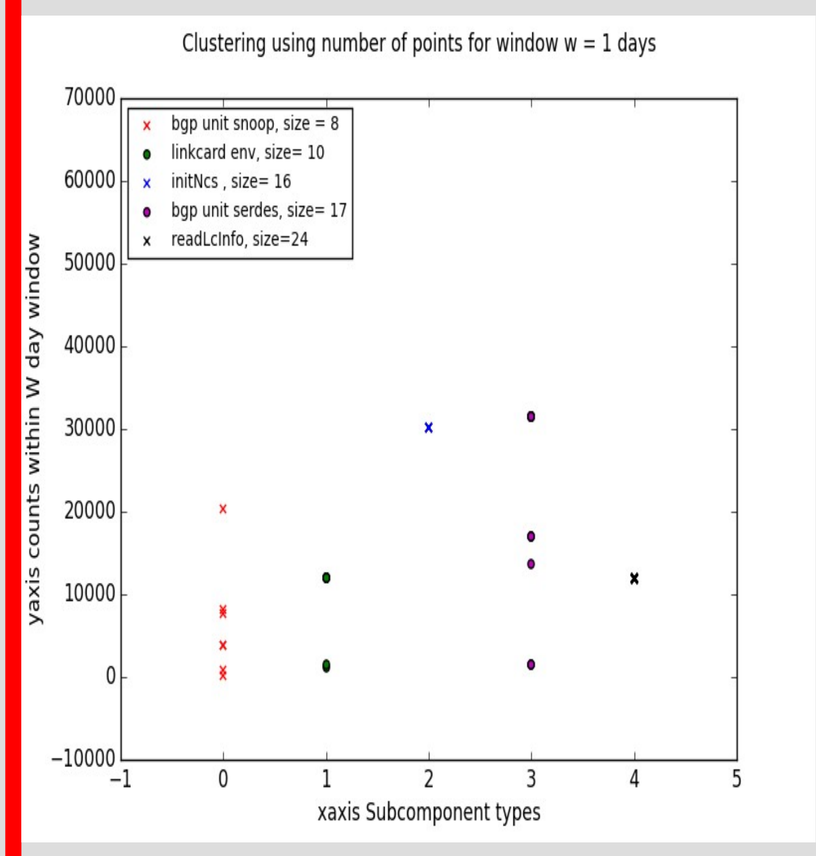
- Implemented DBScan (Density Based Spatial Clustering of Applications with Noise) and Apriori algorithm.
- Data contains noise, duplicate, periodic data. Remove them by looking back in time. Results below for 60 & 30 sec.

Sno (60 s)	Count	Severity	Sno (30 s)	Count	Severity
1	15378	INFO	1	15378	INFO
2	15372	INFO	2	15372	INFO
3	15267	INFO	3	14241	INFO
4	14850	WARN	4	14119	WARN
5	10923	INFO	5	10920	INFO

- Minimize the number of messages by running similarity rules on the data. Jaccard Similarity between messages, with threshold > 0.8 are similar.
- Using clustering straight away gave us mixed results, few fatal errors highly correlated, others varied widely.
- For 2 day window period, the below image shows at points 2 & 4 events to be highly clustered and possibly correlated.

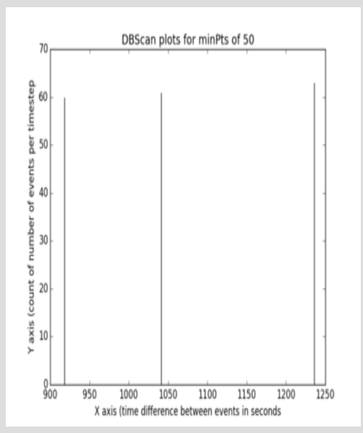


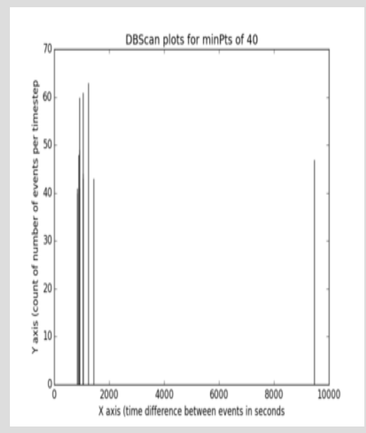
- Ssimilar results even for the events with window size as 1, others are still spread out as in window size of 2 days.



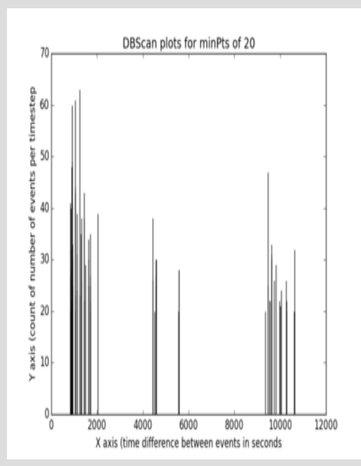
Results:

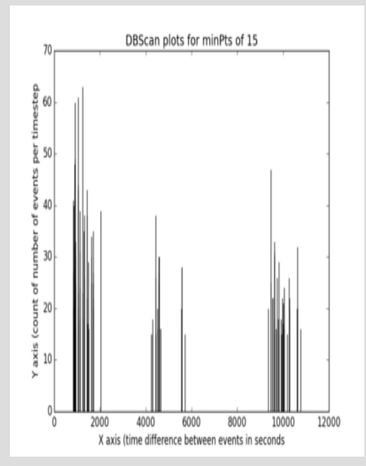
- For DBScan algorithm, window size of 2 days, sub component 1, minPts = 50, 40





- similarly for minPts = 20, 15





- For Apriori Algorithm, for the 5 sub components, the event correlation table with Pair of correlated messages and descriptions below.

subcomponent	Rules written as part of the GID
_bgp_unit_snoop	(71, 65), (71, 67)
LINKCARD_ENV	(72, 67), (70, 65), (72, 65), (71, 67), (64, 67), (70, 67), (71, 65), (70, 64)
InitNcs	(191, 189), (74, 65), (196, 72), (184, 69), (98, 64), (186, 71), (178, 97)
_bgp_unit_serdes	(69, 65), (71, 65), (69, 67)
ReadLcInfo	(70, 98), (84, 70), (180, 72), (64, 73), (73, 69), (99, 69), (75, 98), (97, 95), (98, 71)

TABLE 5
Generating correlation rules using Apriori Algorithm

Message	GID
There were d+ proxy processes that did not cleanly at the end of job V*	27
Collective network receiver link d+ has spent d+ cycles resynchronizing with the sender	28
DDR controller single symbol error V* Controller d+ chipselect d+ V*	29
DDR controller chipkill error V* Controller d+ chipselect d+ V*	30
Error d+ reading message d+ from control V*	31

Conclusions:

- Prediction can definitely be applied to'FATAL' events 2 & 4 and recovery routines can be triggered.
- The minPts parameter in DBScan showed that 50 points was good enough to identify the correlation, other experiments were unnecessary.
- DBScan performed better for the 5 sub components as it could give a more precise correlation than Apriori.

Acknowledgments: This project was performed as part of Spring 2016 CS 6140 course, with the ideas based from the course website and from the papers (1).*Event log mining tool for large scale hpc systems* and (*2) Logmaster: Mining event correlations in logs of large scale cluster systems*. We would also thank Prof. Jeff Philips and the T.A.'s for the help during the course project.