I/O Performance Analysis Framework on Measurement Data from Scientific Clusters

Michelle Koo[1], Wucherl Yoo (Advisor)[2], Alex Sim (Advisor)[2]
[1]University of California Berkeley, [2]Lawrence Berkeley National Laboratory

RESEARCH GOAL

To develop an I/O performance analysis framework to identify performance characteristics in scientific applications by analyzing measurement data collected on scientific clusters.

BACKGROUND INFO

- Project is motivated by observations that I/O performance analyses can be conducted from monitored performance measurement data from scientific clusters
- Large science projects rely on thousands of CPUs to produce/analyze petabytes of data
- Understanding and debugging performance issues are challenging because:
  - Current data accesses may compete with each other for shared data storage and networking resources
  - Unexpected delays possibly due to: complexity of storage and memory hierarchies on hardware, or temperature-based throttling mechanisms in modern CPUs
- Developed I/O analysis framework tool used to study I/O performance behavior
- Experiments ran on a cluster with several hundred machines with two 8-core CPUs: Intel Xeon E5-2670 and 64 GB memory

SOFTWARE

- iPython notebook & iPython widgets: to create the interactive analysis tool
- Python's Matplotlib: to create visualization tools of the data stored in the Resilient Distributed Datasets (RDDs)
- Apache Spark: to query, compute, and sort measurement data into different RDDs
  - Spark allows framework to quickly integrate new performance info gathered in the database
  - Computations are distributed to multiple nodes and executed in parallel

APPLICATION: PALOMAR TRANSIENT FACTORY

- Palomar Transient Factory (PTF) - a fully-automated, wide-field survey of the sky to detect transient objects
- Data recorded from cameras are transferred to NERSC Edison where it undergoes a real-time reduction pipeline
- PTF analysis pipeline consists of 38 checkpoints
- Application logs contain timestamps of the start & end times of each checkpoint that were analyzed to determine concealed I/O bottlenecks in the PTF analysis pipeline
- PTF was executed on compute nodes each with two 12-core CPUs: Intel Xeon E5-2670 and 64 GB memory
- 1.6 TB of system logs collected & processed by the developed framework tool from database on NERSC Edison

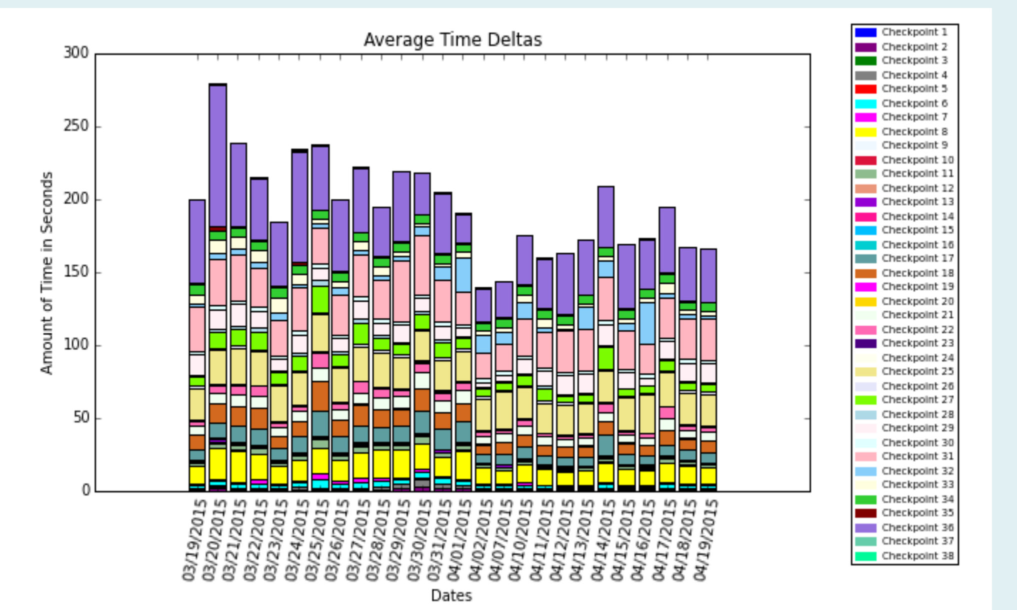

Figure 1: Shows the average amount of time in seconds that each operation takes from March 19, 2015 to April 19, 2015.

RESULTS

- Top 3 most significant checkpoints:
  checkpoints 25, 31, and 36 (Figure 1)
- Average percentage calculations over 64 days:
  - Checkpoint 25 ~ 11.16%
  - Checkpoint 31 ~ 14.79%
  - **Checkpoint 36 ~ 23.72%**
- Checkpoint 36 was the largest bottleneck uncovered in the PTF analysis pipeline (Figures 2, 3, and 4)
- Checkpoint 36 involves the Transients in the Local Universe query which consists of a table lookup
- PTF field name 2049 was found to have the longest average amount of execution time (Figures 5 and 6)
- Time execution build up in Figure 3 reveals a possible system anomaly

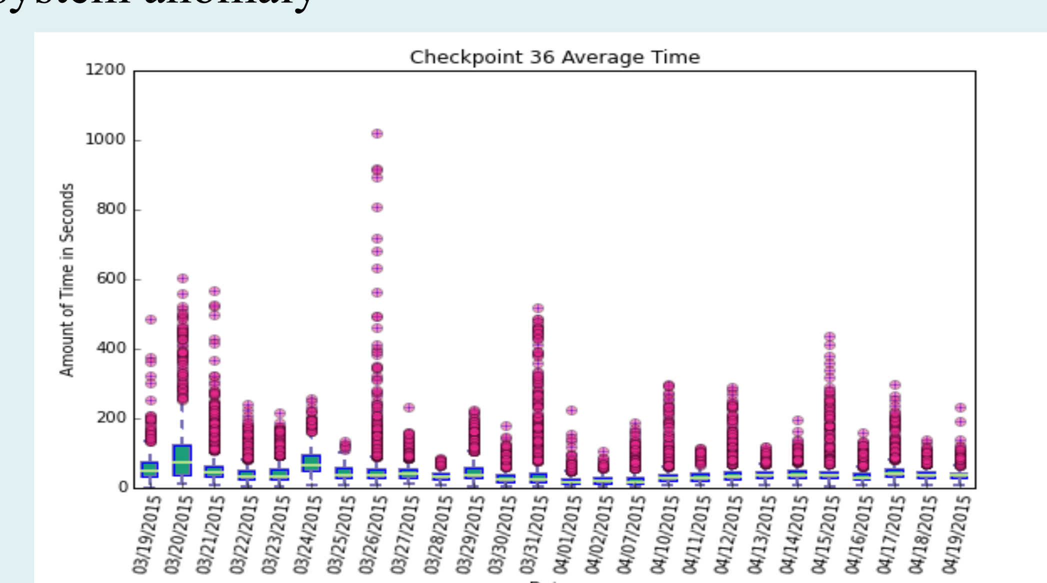

Figure 2: Shows the amount of time in seconds each timestamp of checkpoint 36 takes per day.

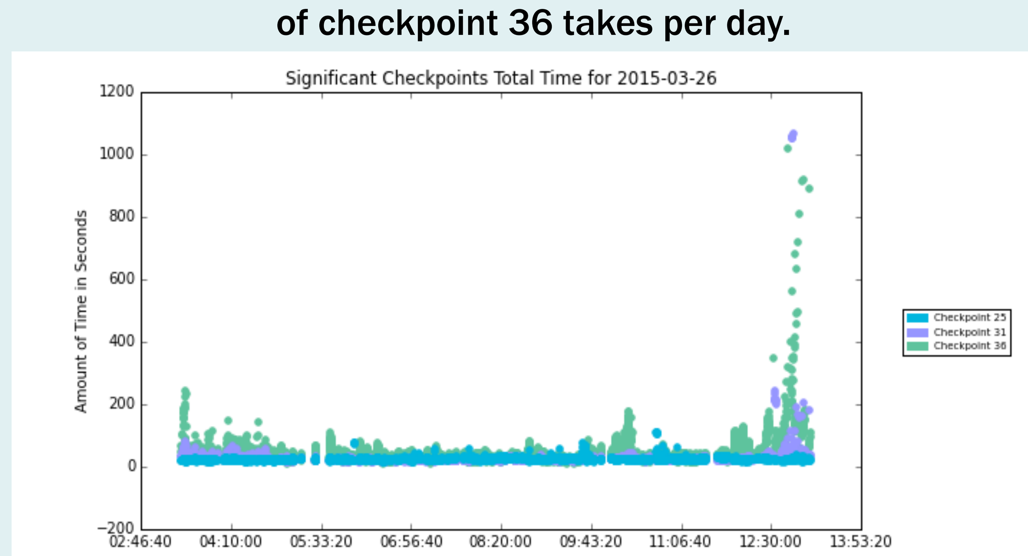

Figure 3: Shows all of the timestamps and the total execution time of Checkpoints 25, 31, and 36 throughout March 26, 2015.

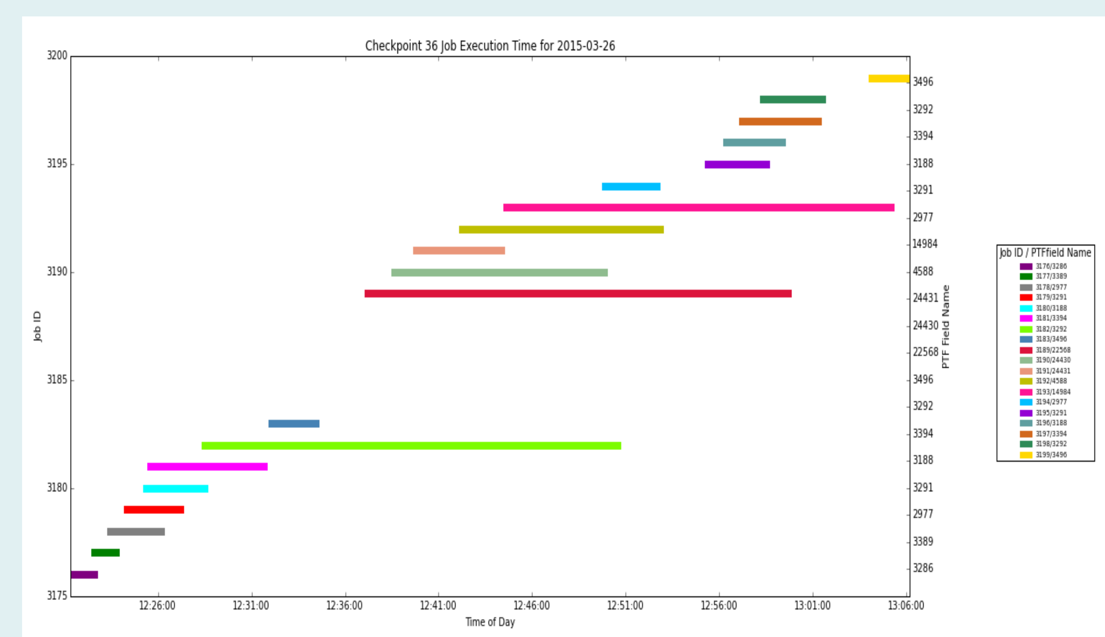

Figure 4: Illustrates the time execution of all jobs and their corresponding PTF field names on March 26, 2015.

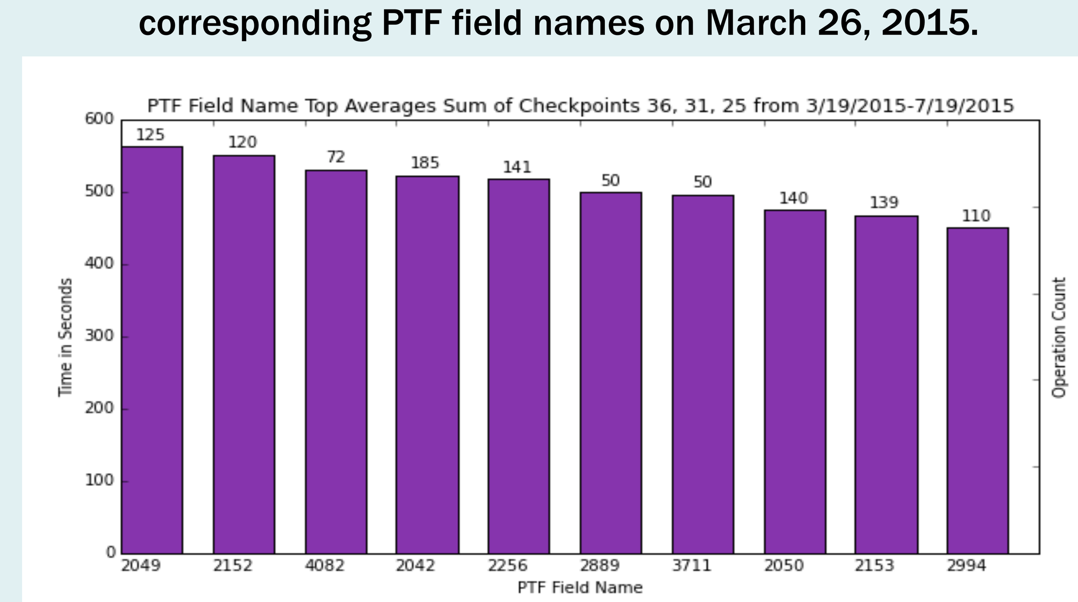

Figure 5: Shows the PTF field names of the average longest time execution of the sum of checkpoints 25, 31, and 36.

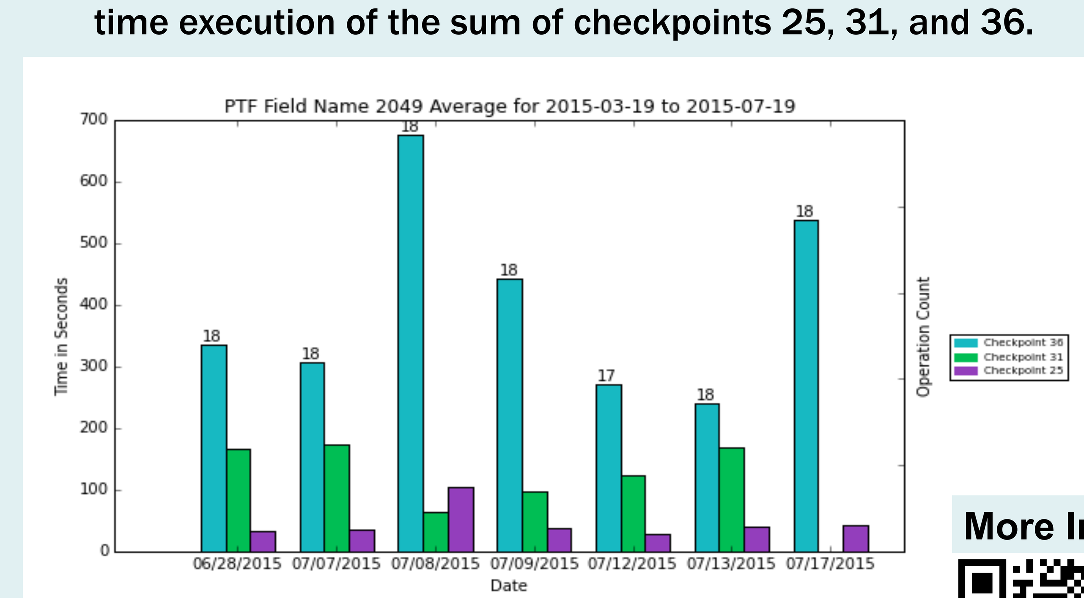

Figure 6: Shows the amount of time on average PTF field name 2049 takes per day for checkpoints 25, 31, and 36.

More Info

FRAMEWORK TOOL

- Allows users to interactively explore performance data for maximal understanding of applications

Figure 7: A screenshot of the I/O Analysis Framework that was developed to uncover I/O performance bottlenecks.

CONCLUSION

- I/O analysis framework applied to PTF analysis pipeline uncovered major bottlenecks (Checkpoint 36 & PTF field name 2049)
- PTF scientists suggest possible cause of bottlenecks:
  - TILU query accesses "local" galaxies, resulting in a high hit rate, but as PTF field counts increase, execution time slows
- Developed I/O analysis framework tool:
  - Allows users greater autonomy & maximizes understanding of underlying workflows in applications
  - Provides execution time analysis & data dependency analysis on different types of performance measurements
  - Helps identify and analyze hidden I/O performance bottlenecks
- Future work:
  - Optimize major uncovered I/O bottlenecks in PTF analysis pipeline
  - Apply I/O analysis framework tool to other scientific applications for performance analysis

I would like to thank my mentors Alex Sim and Wucherl (William) Yoo, and also John K. Wu, Peter Nugent, and Yi Cao for their support and feedback. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Berkeley UNIVERSITY OF CALIFORNIA
U.S. DEPARTMENT OF ENERGY Office of Science
BERKELEY LAB
NERSC
SDM
PALOMAR TRANSIENT FACTORY