The Mystery of the Failing Jobs: Insights from Operational Data from Two University-Wide Computing Systems

Rakesh Kumar¹, Saurabh Jha², Ashraf Mahgoub¹, Rajesh Kalyanam¹, Stephen L Harrell¹, Xiaohui Carol Song¹, Zbigniew Kalbarczyk², William T Kramer², Ravishankar K Iyer², Saurabh Bagchi⁰

¹: Purdue University, 2: University of Illinois at Urbana-Champaign

Supported by NSF

Overview

• Introduction
• System and Data Details
• Job Characteristics
• Analyses
  – Job Categories Based on Exit Statuses
  – Effect of Resource Usage on Job Failures
  – Predicting Job Failures and a Better Checkpointing Method
• Open Challenges
• Conclusion
Introduction

- Job failure leads to resource wastage and user dissatisfaction
- University computing clusters are uniquely challenging:
  - Heterogeneity of jobs: Compute-Intensive, Memory-Intensive, IO-Intensive
  - Varied expertise level of the users
  - Relatively smaller size of the system administration staff
- The most comprehensive dataset publicly analyzed to date in terms of variety of data sources
  - Accounting logs, resource utilization stats, failure reports
  - System-A: Less expensive HW, 617 users, ~3M jobs
  - System-B: More expensive HW, 467 users, ~2M jobs
- New insights and old insights in new environments
  - Recommendations to reduce job failure/resource wastage for both system user and system admin
  - Build an actionable failure prediction model based on resource usages

System and Data Details

- System A
  - 580 nodes, Intel Xeon E5-2670 processors, 64 GB/node, 100 MB/s local IO BW, 23 GB/s network IO BW
- System B
  - 26,868 nodes, AMD 6276 Interlagos processors, 64 GB/node, 1.1 TB/s network IO BW
- Data
  - Accounting logs
  - Resource Utilization Stats
    - 5-minute granularity for System A and 1-minute granularity for System B
  - Node Failure Reports
### Summary of Data Analyzed

<table>
<thead>
<tr>
<th>Computing Cluster</th>
<th>System A</th>
<th>System B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>Mar 2015-Jun 2017</td>
<td>Feb-June 2017</td>
</tr>
<tr>
<td># jobs</td>
<td>2,908k</td>
<td>2,219k</td>
</tr>
</tbody>
</table>

#### Shared

- **single**: 1,125k (38.7%, 15.8%) - 1,153k (39.7%, 17.7%)
- **multi**: 28k (1.0%, 1.9%) -
- **total**: 1,153k (39.7%, 17.7%) -

#### Non-shared

- **single**: 1,348k (46.3%, 18.4%) - 1,640k (73.9%, 9.4%)
- **multi**: 407k (14.0%, 63.9%) - 580k (26.1%, 94.6%)
- **total**: 1,755k (60.3%, 82.3%) - 2,219k (100%)

- **All production jobs**
- **Node-seconds** = #nodes x execution time
- **The percentages in parenthesis refers to the raw counts and node-seconds**
- **Sharing allows multiple jobs to run on the same node**
  - **System A**: 39.7% by count and 17.7% by node-seconds

### Job Characteristics

- **Job size**
  - **Single node jobs by count**
    - **System A**: 85%, **System B**: 74%
  - **Single node jobs by node-seconds**
    - **System A**: 34%, **System B**: 5%
Job Characteristics

- **Job node-seconds**
  - System A and System B: 50% of the jobs run for less than $\sim 10^3$ node-seconds
  - System B: Jobs run up to $\sim 10^9$ node-seconds

![Graph showing PDF and CDF for System A and System B](image)

Job Categories Based on Exit Statuses

Table: Job categories based on exit codes. Percentages in brackets are based on the total node-seconds

<table>
<thead>
<tr>
<th>Category</th>
<th>Environment &amp; Job Type</th>
<th>System A</th>
<th>System B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>single</td>
<td>multi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared</td>
<td>multi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>overall</td>
<td>overall</td>
</tr>
<tr>
<td>Success</td>
<td></td>
<td>67.6%</td>
<td>68.8%</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td>6.5%</td>
<td>6.5%</td>
</tr>
<tr>
<td>User</td>
<td></td>
<td>1.2%</td>
<td>1.2%</td>
</tr>
<tr>
<td>User/System</td>
<td></td>
<td>0.4%</td>
<td>0.4%</td>
</tr>
<tr>
<td>Walltime</td>
<td></td>
<td>4.5%</td>
<td>4.5%</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1.25k</td>
<td>1.348k</td>
</tr>
</tbody>
</table>

- **Failure categories** - System, User, User/System
- **System related failures** - System A: 5.3%, System B: 0.3%
- **Success category**
  - Multi-node - System A: 61.8% (non-shared), System B: 64.0% (non-shared)
  - Single-node - System A: 93.1% (shared) vs 87.6% (non-shared), System B: 91.6%
    - Sharing does not negatively impact the jobs failure probability.
- **Walltime category by node-seconds** - System A: 33.4%, System B: 43.3%
Effect of Resource Usage on Job Failures

- Job failure rate is defined as the fraction of jobs that fail due to system related issues
- All analyses conducted using tail utilization values
- Hypothesis testing for all correlation studies
- Resource usage prediction models based on user profiling
  - Last: same resource usage as last finished job of a given user
  - Average: average resource usage of last ‘n’ finished job of a given user
  - Median: median resource usage of last ‘n’ finished job of a given user
  - Maximum cosine similarity: same resource usage as the most similar job based on cosine similarity

Effect of Resource Usage on Job Failures

- Memory
  - Single-node jobs: +ve correlation
  - Multi-node jobs: no correlation
  - 99th percentile value:
    - System A – 11.7 GB
    - System B – 45.6 GB

<table>
<thead>
<tr>
<th>System</th>
<th>Job Type</th>
<th>Correlation coeff., p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Non-shared single</td>
<td>0.83, 1.7e-28</td>
</tr>
<tr>
<td></td>
<td>Non-shared multi</td>
<td>0.17, 0.4</td>
</tr>
<tr>
<td></td>
<td>Shared single</td>
<td>0.84, 7.2e-32</td>
</tr>
<tr>
<td>B</td>
<td>Non-shared single</td>
<td>0.57, 3.2e-9</td>
</tr>
<tr>
<td></td>
<td>Non-shared multi</td>
<td>0.13, 0.2</td>
</tr>
</tbody>
</table>
Resource Usage Prediction by User Profiling

- Memory (System A)
  - MAPE of all predictors are less than 12% (for at least one history length)
  - Maximum Cosine Similarity (MCS) outperforms others
  - Use case: Predict memory usage in advance
    - Better scheduling for heterogeneous memory cluster
    - Better scheduling when sharing is enabled

Summary: Effect of Resource Usage on Job Failure

- Random IO access requests lead to failure even at ~1% of BW
  - Local IO (System A)
    - BW of 100MB/s while failure rate starts rising with utilization as low as 3 MB/s (shared) - 6MB/s (non-shared)
  - Remote IO (System B)
    - BW of 1.1TB/s while failures are observed with a utilization of only 46MB/s for a given job

- Contention at remote resources (outside node) dominant in non-shared environment, while the contention at local resources (at node) dominant in shared environment.
  - Use user-based resource usage prediction while making scheduling decisions
  - Use dynamic reconfiguration of applications based on current resource availability, such as reconfiguring the number of threads or network timeout.
Predicting Job Failure

- **ML model**
  - Input: current resource usages, Output: failure probability within the next monitoring window
- **Better checkpointing method**
  - Combine our ML model with the optimal periodic checkpointing method

A Better Checkpointing System

<table>
<thead>
<tr>
<th>System</th>
<th>Periodic</th>
<th>ML</th>
<th>ML + Periodic</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTBF=1e4, Ts=60sec</td>
<td>A</td>
<td>shared single</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared single</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared multi</td>
<td>0.50</td>
</tr>
<tr>
<td>B</td>
<td>non-shared single</td>
<td>0.51</td>
<td>0.54</td>
</tr>
<tr>
<td>MTBF=1e5, Ts=60sec</td>
<td>A</td>
<td>shared single</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared single</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared multi</td>
<td>0.50</td>
</tr>
<tr>
<td>B</td>
<td>non-shared single</td>
<td>0.51</td>
<td>0.54</td>
</tr>
<tr>
<td>MTBF=1e6, Ts=60sec</td>
<td>A</td>
<td>shared single</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared single</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared multi</td>
<td>0.50</td>
</tr>
<tr>
<td>B</td>
<td>non-shared single</td>
<td>0.51</td>
<td>0.54</td>
</tr>
<tr>
<td>MTBF=1e6, Ts=10sec</td>
<td>A</td>
<td>shared single</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared single</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>non-shared multi</td>
<td>0.57</td>
</tr>
<tr>
<td>B</td>
<td>non-shared single</td>
<td>0.67</td>
<td>0.70</td>
</tr>
</tbody>
</table>

- **ML + periodic checkpointing method outperforms the base optimal checkpointing method by between 12.3% (unreliable system with MTBF =1e4, Ts=60s) and 2X (reliable system with MTBF = 1e6 and Ts=60s).**
- **Savings achieved by the optimal checkpointing method in case of failure decreases as a system becomes more reliable i.e., as the MTBF increases from 1e4 to 1e6.**
Open Challenges

• Current optimal checkpointing estimation methods take only hardware reliability (such as MTBF) into account
  – This paper integrated it with job failure likelihood information
  – A better method is to consider in addition the rate of job progress
• Current contention-aware schedulers need to profile a job first to estimate job’s interference and latency-sensitivity
  – Major limitation for clusters where majority of jobs are short running
  – Use user history-based resource usage predictions to profile a job profile

Conclusion

• The most comprehensive dataset publicly analyzed to date in terms of variety of data sources
• Publicly released the dataset on which the analyses are based
• Important insights into how the clusters behave and implications for how they can be managed more effectively.
Thank You!