USING SIMPLE PID CONTROLLERS TO PREVENT AND MITIGATE
FAULTS IN SCIENTIFIC WORKFLOWS

Rafael Ferreira da Silva¹, Rosa Filgueira², Ewa Deelman¹, Erola Pairo-Castineira³, Ian Michael Overton⁴, Malcolm Atkinson⁵

¹USC Information Sciences Institute
²British Geological Survey, Lyell Centre
³MRC Institute of Genetics and Molecular Medicine, University of Edinburgh
⁴Usher Institute of Population Health Sciences and Informatics, University of Edinburgh
⁵School of Informatics, University of Edinburgh

11th Workflows in Support of Large-Scale Science (WORKS’16)
Salt Lake City, UT – November 14th, 2016
WHY SCIENTIFIC WORKFLOWS?

**Automation**
Enables parallel, distributed computations
Automatically executes data transfers

**Recover & Debug**
Handles failures with to provide reliability
Keeps track of data and files

**Reproducibility**
Reusable, aids reproducibility
Records how data was produced (provenance)

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson
*Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows*
A science-gateway workload archive to study pilot jobs, user activity, bag of tasks, task sub-steps, and workflow executions

Rafael Ferreia da Silva and Tristan Gistard
University of Lyon, CNRS, INSERM, CREATEIS, Villeurbanne, France
(rafael.silva.gistard)@univ-lyon.fr

Abstract: Archives of distributed workflows acquired at different infrastructural and application levels include logs. Science gateways provide consistent access points to the

MOTIVATION

Grid computing
180k failed tasks out of 340k

Characterizing a High Throughput Computing Workload:
The Compact Muon Solenoid (CMS) Experiment at LHC
Rafael Ferreira da Silva1,2, Mate Rygajd3, Gideon Jones3, Igor Shligil3, Ewan Deelmaan4, James Lottal4, Frank Würthwein4, and Miran Livio5
1 University of Southern California, Information Sciences Institute, Marina Del Rey, CA, USA
2 University of California at San Diego, Department of Physics, La Jolla, CA, USA
3 University of Wisconsin Madison, Madison, WI, USA
4 Justus-Liebig-Universität, Giessen, Germany
5 The Ohio State University, Columbus, USA

Abstract: High-throughput computing (HTC) has aided the scientific community in the analysis of vast amounts of data and computational jobs is distributed environments. To manage these large workloads, several systems have been developed to efficiently allocate and provide access to distributed resources. Many of these systems rely on job characterization estimates (e.g., job

CMS (Aug 2014)
385k failed tasks out of 790k

Consecutive Job Submission Behavior
at Mira Supercomputer
Stephan Schlagkamp1, Rafael Ferreira da Silva1, William Allcock2
1 RoboS Research Institute, T.T. Bomholt University, Dortmund, Germany
2 University of Southern California, Information Sciences Institute, Marina Del Rey, CA, USA

Abstract: Understanding user behavior is crucial for the evaluation of submitting and allocating performances to HTC environments. This paper aims to further understand the dynamics of job behaviors between parallel job characteristics. We analyze</p>

The Failure Trace Archive
26 datasets from 2006-2014

The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems
Derrick Kondo1, Bahram Javadi2, Alexandre Iosup3, Dick Epsen4
1 INRIA, France, 2TU Delft, The Netherlands

Abstract: With the increasing functionality and complexity of distributed systems, resource failures are unavoidable. While numerous models and algorithms for dealing with failures exist, the lack of public trace data sets and tools have prevented meaningful comparisons. To facilitate the de-
SOME APPROACHES TO HANDLE FAULTS

**Typical Approaches**
- Task Retries
- Task Resubmission
- Task Clustering
- Checkpointing
- Provenance
  ...

**Statistical and Machine Learning**
- Linear Regression
- Neural Networks
- Classification Algorithms
- Tree-based Methods
- Support Vector Machines
  ...

**Analytical Solutions**
- Failure Modeling
- Markov Chains
- Principal Component Analysis
- Histograms
  ...

**Others**
- Exception Handling
- Game Theory
  ...

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson

*Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows*
Most of them make strong assumptions about resource and application characteristics. Accurate estimates of such requirements are still a steep challenge.

Some approaches may overload the execution platform.

Most of the systems do not prevent faults, but mitigate them.

Some approaches are tied to a small set of applications.

We seek for an approach to predict, prevent, and mitigate failures in end-to-end workflow executions across distributed systems under online and unknown conditions.

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson
Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows
**PID CONTROLLERS**

**Proportional-Integral-Derivative Controller**

Control loop mechanism

Widely used in industrial control systems
- Temperature
- Pressure
- Flow rate
- etc.

*PID controller aims at detecting the possibility of a fault far enough in advance so that an action can be performed to prevent it from happening*
**PROCESS VARIABLES**

**Proportional-Integral-Derivative Controller**

\[ u(t) = K_p e(t) + K_i \int_0^t e(t) \, dt + K_d \frac{de(t)}{dt} \]

- **Proportional** (Present error)
- **Integral** (Accumulation of past errors)
- **Derivative** (Prediction of future errors based on current rate of change)

**Variables and Constants**

- \( K_p \): Proportional gain constant
- \( K_i \): Integral gain constant
- \( K_d \): Derivative gain constant
- \( e \): Error defined as the difference between the setpoint and the process variable value

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson

*Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows*
A run of scientific workflows that manipulate large data sets may lead the system to an out of disk space fault.

**PID Controller**

- **P**: the error between the setpoint, and the actual used disk space
- **I**: cumulative value of the proportional responses
- **D**: the difference between the current and the previous disk overflow (or underutilization) error values

**Actions**

- $u(t) < 0$: data cleanup is used to remove unused data; or tasks are preempted
- $u(t) > 0$: the number of concurrent task executions may be increased
The performance of memory-intensive operations are often limited by the memory capacity of the resource where the application is being executed.

**PID Controller**

**P:** error between the *setpoint* value, and the actual memory usage

**I:** cumulative value of previous memory usage errors

**D:** difference between the current and the previous memory overflow (or underutilization) error values

**Actions**

- $u(t) < 0$: tasks are preempted to prevent the system to run out of memory
- $u(t) > 0$: the WMS may spawn additional tasks for concurrent execution
1000 Genome Sequencing Analysis Workflow

Identifies mutational overlaps using data from the 1000 genomes project

22 Individual tasks, 7 Population tasks, 22 Sifting tasks, 154 Pair Overlap Mutations tasks, and 154 Frequency Overlap Mutations tasks (Total 359 tasks)

The workflow consumes/produces over 4.4TB of data, and requires over 24TB of memory

https://github.com/pegasus-isi/1000genome-workflow
EXPERIMENT SETUP

Workflow Management System
PID Control Loop

Compute Node 1
Compute Node 2

Shared memory

Memory
Memory

Disk

capacity 500GB

Shared filesystem

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson

Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows
we arbitrarily define our *setpoint* as 80% of the maximum total capacity (for both storage and memory usage, and a *steady-state error of 5%*.

We assume $K_p = K_i = K_d = 1$ (no tuning).

### Reference Workflow Execution

Computed offline under *known conditions*

**Averaged Makespan:** ~106h

(standard deviation < 5%)

Execution are performed under *online* and *unknown conditions*.

the decision on the number of tasks to be *scheduled or preempted* is computed as the *min* between the response value of the unique disk usage PID controller, and the memory PID controller per resource.
OVERALL MAKESPAN EVALUATION

Average workflow makespan for different configurations of the controllers

**Proportional**

- \( K_p = 1, \ K_i = K_d = 0 \)
- Makespan: 138.76h
- Slowdown: 1.30

**Proportional-Integral**

- \( K_p = K_i = 1, \ K_d = 0 \)
- Makespan: 126.69h
- Slowdown: 1.19

**Proportional-Integral-Derivative**

- \( K_p = K_i = K_d = 1 \)
- Makespan: 114.96h
- Slowdown: 1.08

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson
*Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows*
EXPERIMENTS: DATA FOOTPRINT

PROPORTIONAL

INTEGRAL

DERIVATIVE

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson

Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows
This process occurs at about 4h, and performs more than 6,000 preemptions.
EXPERIMENTS: MEMORY USAGE

PROPORTIONAL

INTEGRAL

DERIVATIVE

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson
Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows
only a few tasks (on average less than 5) are preempted due to memory overflow
OVERALL RESULTS

<table>
<thead>
<tr>
<th>Controller</th>
<th># Tasks Preempted</th>
<th># Cleanup Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>7225</td>
<td>0</td>
</tr>
<tr>
<td>PI</td>
<td>168</td>
<td>48</td>
</tr>
<tr>
<td>PID</td>
<td>73</td>
<td>4</td>
</tr>
</tbody>
</table>

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Païro-Castineira, I. M. Overton, M. Atkinson

*Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows*
**Execution Environment**

The goal of tuning a PID loop is to make it stable, responsive, and to minimize overshooting.

**Ziegler-Nichols Method**

1. Turn the PID controller into a P controller by setting $K_i = K_d = 0$. Initially, $K_p$ is also set to zero.

2. Increase $K_p$ until there are sustained oscillations in the signal. This $K_p$ value is the ultimate gain, $K_u$.

3. Measure the ultimate (or critical) period $T_u$ of the sustained oscillations.

---

### Tuned gain parameters

<table>
<thead>
<tr>
<th>Controller</th>
<th>$K_u$</th>
<th>$T_u$</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Footprint</td>
<td>0.58</td>
<td>3.18</td>
<td>0.35</td>
<td>0.22</td>
<td>0.14</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>0.53</td>
<td>12.8</td>
<td>0.32</td>
<td>0.05</td>
<td>0.51</td>
</tr>
</tbody>
</table>

---

Ziegler-Nichols tuning, using the oscillation method.

---

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson

*Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows*
The key factor of its success is due to the specialization of the controllers to a single application.

 Avg. Makespan: 107.37h  
 Avg. Slowdown: 1.01  
 Preempted Tasks: 18  
 Cleanup Tasks: 1
SUMMARY

Conclusions
Experimental results show that faults are detected and prevented before their occur, leading workflow execution to its completion with acceptable performance.

**PID controllers should be used sparingly, and metrics (and actions) should be defined in a way that they do not lead the system to an inconsistent state.**

Future Research Directions
We will investigate the simultaneous use of multiple control loops at the application and infrastructure levels, to determine to which extent this approach may negatively impact the system.

R. Ferreira da Silva, R. Filgueira, E. Deelman, E. Pairo-Castineira, I. M. Overton, M. Atkinson
Using Simple PID Controllers to Prevent and Mitigate Faults in Scientific Workflows
USING SIMPLE PID CONTROLLERS TO PREVENT AND MITIGATE FAULTS IN SCIENTIFIC WORKFLOWS

Thank You

Questions?

Rafael Ferreira da Silva, Ph.D.
Research Assistant Professor
Department of Computer Science
University of Southern California
rafsilva@isi.edu – http://rafaelsilva.com

http://pegasus.isi.edu