

Rapid Information Factory

Applying Lean Six Sigma to Parallel Processing Framework

Andreas Francois Vermeulen

University of St Andrew and University of Dundee
afvermeulen@dundee.ac.uk

Dr Vladimir Janjic

University of St Andrew

Prof Janet Hughes

University of Dundee

jhughes@computing.dundee.ac.uk
vj32@st-andrews.ac.uk

Abstract

This is the text of the abstract.

Categories and Subject Descriptors CR-number [subcategory]: third-level

General Terms Parallel applications and frameworks, lean six sigma

Keywords rapid information factory, rif, framework, lean six sigma, heterogeneous computing, parallel, beowulf, cluster, master-slave, pipe-line.

1. Introduction

The Rapid Information Factory is a data processing architecture that enables the enhanced processing of data by using a structured and highly optimised parallel processing framework. The core of the framework is a processing pipeline with feedback to enhance the processing of the data into information. Our research covers the structure of this processing framework and the use of a three node Beowulf cluster [2] that combines into the formation of the Rapid Information Factory. The improve the performance of the factory we apply same Lean Six Sigma [1] rules that applies to normal manufacturing factories with proven success.

2. Research Question

"Does a Rapid Information Factory improve processing of data into information when Lean Six Sigma improvements normally applied to manufacturing factories is used to guide improvements?"

3. Rapid Information Factory

The factory is a singular processing solution that processes all data into information using a XML based rules. The solution consists of an interaction between three dimensional frameworks:

3.1 Customer Framework

This is the view the single view of the information to the customer. It is structured to show the information in the view the customer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CONF 'yy, Month d-d, 20yy, City, ST, Country.

Copyright © 2016 ACM 978-1-xxxx-xxxx-n/yy/mm...\$15.00.

<http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn>

specifies in the functional requirements. As for this specific research this is not expanded or discussed further.

3.2 Project Framework

The factory evolves over time as it is developed and then redeveloped to handle extra enhancements or new data sources. The project is driven by a agile project methodology consisting a backlog and a cycle of five day sprints covering a period of six months. As for this specific research this is not expanded or discussed further.

3.3 Rapid Information Factory Framework

The framework is an ontology scripting the processes in Extensible Markup Language (XML) to define the set of rules for encoding the process and the interactions between process. The framework consists of a five high-level layers:

3.3.1 Business Layer

The Business Layer contains the Business specific framework configurations that covers either functional requirements or non-functional requirements. The layer consists of two groupings:

1. Functional
2. Non-functional

As for this specific research this is not expanded or discussed further.

3.3.2 Utility Layer

The Utility Layer consists of sets of utilities for the overall factory. The layer consists of two groupings of utilities:

1. Maintenance utilities
2. General utilities

As for this specific research this is not expanded or discussed further.

3.3.3 Audit, Balance and Control Layer

The Audit, Balance and Control Layer (ABC) is the layer that supports the factory while it is running. This layer handles any active processing allocated to the Beowulf engine. The layer consists of three groupings.

1. Audit
2. Balance
3. Control

3.3.4 Operational Management Layer

The Operational Management Layer supports setup of the individual job definition and interaction between jobs, job parameters,

scheduling, monitoring, communication and alerting within the factory. The layer is the central management engine of the factory. The layer consists of five groupings:

1. Jobs
2. Schedule
3. Monitor
4. Communication
5. Alerts

3.3.5 Functional Layer

The functional layer store the scripts that describes every functional process of the complete factory. The layer consists of six groupings of jobs:

1. Retrieve
2. Assess
3. Process
4. Transform
5. Organise
6. Report

3.4 Audit, Balance and Control Layer

The Audit, Balance and Control Layer (ABC) is the layer that supports the factory while it is running. This layer handles any active processing allowcated to the Beowulf engine. The layer consists of three groupings.

3.4.1 Audit

3.4.2 Balance

3.4.3 Control

3.5 Operational Management Layer

The Operational Management Layer supports setup of the individual job definition and interaction between jobs, job parameters, scheduling, monitoring, communication and alerting within the factory. The layer is the central management engine of the factory. The layer consists of five groupings:

3.5.1 Jobs

3.5.2 Schedule

3.5.3 Monitor

3.5.4 Communication

3.5.5 Alerts

3.6 Functional Layer

The functional layer store the scripts that describes every functional process of the complete factory. The layer consists of six groupings of jobs:

3.6.1 Retrieve

3.6.2 Assess

3.6.3 Process

3.6.4 Transform

3.6.5 Organise

3.6.6 Report

A. Performance Improvement Results

The Performance Improvement Results is as follows: Applying 5S - Sort improvement process to Retrieve Jobs. Applying 5S - Set-in-

Order improvement process to Retrieve Jobs. Applying 5S - Shine improvement process to Retrieve Jobs. Applying 5S - Standardise Sort improvement process to Retrieve Jobs. Applying 5S - Sustain improvement process to Retrieve Jobs. Applying Lean Waste - Transport to Retrieve Jobs. Applying Lean Waste - Inventory to Retrieve Jobs. Applying Lean Waste - Motion to Retrieve Jobs. Applying Lean Waste - Waiting to Retrieve Jobs. Applying Lean Waste - Overproduction to Retrieve Jobs. Applying Lean Waste - Over-processing to Retrieve Jobs. Applying Lean Waste - Defects to Retrieve Jobs.

Acknowledgments

Thank you to Prof Mark Whitehorn and Prof Janet Hughes for her guidance into the Operational Research, Business Intelligence, Data Science and Data Engineering concepts utilised in this research.

References

- [1] M. George, D. Rowlands, M. Price, and J. Maxey. The lean six sigma pocket toolbox. 2005.
- [2] V. E. Sonzogni, A. M. Yommi, N. M. Nigro, and M. A. Storti. A parallel finite element program on a beowulf cluster. *Advances in Engineering Software*, 33(7):427–443, 2002.