Rapid Information Factory

Applying Lean Six Sigma to Parallel Processsing 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 the data into information. Our research covers the structure of this processing framework and the use of a three node Beowulf cluster [6] that combines into the formation of the Rapid Information Factory. The improve the performance of the factory we apply same Lean Six Sigma [4] rules that applies to normal manufacturing factories with proven success.

2. Research Question

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

3. Background

The following is the backgound research for this presentation.

3.1 Lean work cells

The Lean work cells [2] is an optimised processing unit that performs work with singlular work outcome. The factory is built up using various manufacturing processes joint to form work cells that delivers the works on a just-in-time processing principal [3]. The lean work cells directly translate into a FastFlow Farm [1] that enables the formation of a optimised processing unit.

4. Lean Waste

A waste is any step or action in a process that is not required to complete a process (i.e. Non Value-Adding) successfully. When Waste is eliminate, only the steps that are required (i.e. Value-Adding) to deliver a satisfactory product or service to the customer remain in the process.

4.1 Eight Lean Wastes

The lean wastes are categorise into eight wastes type. These are the 8 Wastes:

1. Defects : Products or services that are out of specification that require resources to correct.
2. Overproduction : Producing too much of a product before it is ready to be sold;
3. Waiting for the previous step in the process to complete.
4. Non-Utilized Talent Employees that are not effectively engaged in the process.
5. Transportation Transporting items or information that is not required to perform the process from one location to another.
6. Inventory or information that is sitting idle (not being processed).
7. Motion People, information or equipment making unnecessary motion due to workspace layout, ergonomic issues or searching for misplaced items.
8. Extra Processing Performing any activity that is not necessary to produce a functioning product or service.

5. 5S

The 5S is a workplace organization technique composed for five primary phases: Sort, Set In Order, Shine, Standardize, and Sustain.

Sort Keep only necessary items in the workplace.

Review tools, parts, and instructions; Keep only what is essential; Eliminate anything that is non-essential.

Examples: Obsolete or expired procedures, damaged or expired inventory, defunct or old equipment

Set In Order Arrange items to promote efficient work flow.

Arrange items in a logical order; Indicate places for each item clearly; Keep each item close to where it will be used.

Reduces: excess movement, excess transportation, over processing, over production, excess inventory, excess delays, defects

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

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

Figure 1. Eight Lean Wastes

Figure 2. 5S - Sort, Set In Order, Shine, Standardize, and Sustain.

Shine Clean the work area so it is neat and tidy.

Make cleaning a part of daily work; Assign areas of responsibility; Return all items or files to their assigned place.

Examples: Dirty tools and equipment, spills and leaks, clutter and mess

Standardize Set standards for a consistently organized workplace.

Create standards for Sort, Set In Order, and Shine; Make standards easy to understand with Visual Controls; Assign and educate on individual responsibilities.

Examples: Work instructions, hazard warnings, equipment/tool labels, process diagrams

Sustain Maintain and review standards.

Measure and monitor process; Address root causes and avoid reversion to the old ways; Promote individual feedback and response for improvement.

5.1 Poka-Yoke

Poka-yoke is a Japanese term that means mistake-proofing. A poka-yoke is any mechanism in a Lean Manufacturing process that helps an equipment operator avoid (yokeru) mistakes (poka). Its purpose is to eliminate product defects by preventing, correcting, or alert to errors as they occur.

5.2 Takt Time

The Takt Time is the rate at which a finished product needs to be completed in order to meet customer demand.

5.3 Cycle Time

The time it takes to do one repetition of any particular task typically measured from Start to Start the starting point of one products processing in a specified machine or operation until the start of another similar products processing in the same machine or process.

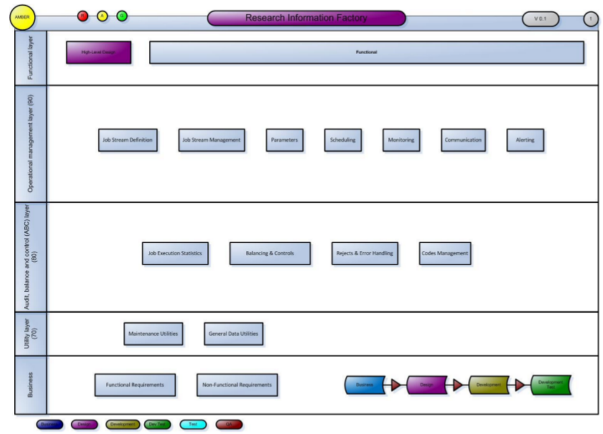

Figure 3. Rapid Information Factory Framework

6. Software - Building Block

6.1 0QM

The system uses the ZeroQM [5] libraries to communicate beteen the processes in the work cells.

ZeroMQ supplies:

- Connect your code in any language, on any platform.
- Carries messages across inproc, IPC, TCP, TIPC, multicast.
- Smart patterns like pub-sub, push-pull, and router-dealer.
- High-speed asynchronous I/O engines, in a tiny library.
- Supports multi language and multi platform.
- Build various architectures: centralised or distributed.

6.2 C++

The system is developed using C++.

7. 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:

7.1 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 spesifies in the functional requirements. As for this spesific research this is not expanded or discussed further.

7.2 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 spesific research this is not expanded or discussed further.

7.3 Rapid Information Factory Framework

The processing farmework 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:

7.3.1 Business Layer

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

1. Functional work cell
2. Non-functional work cell

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

7.3.2 Utility Layer

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

1. Maintance utility work cell
2. General utility work cell

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

7.3.3 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 allowcated to the Beowulf engine. The layer consists of three groupings.

1. Audit work cell
2. Balance work cell
3. Control work cell

7.3.4 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 work cell
2. Schedule work cell
3. Monitor work cell
4. Communication work cell
5. Alerts work cell

7.3.5 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 work cell
2. Assess work cell
3. Process work cell
4. Transform work cell
5. Organise work cell
6. Report work cell

7.4 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.

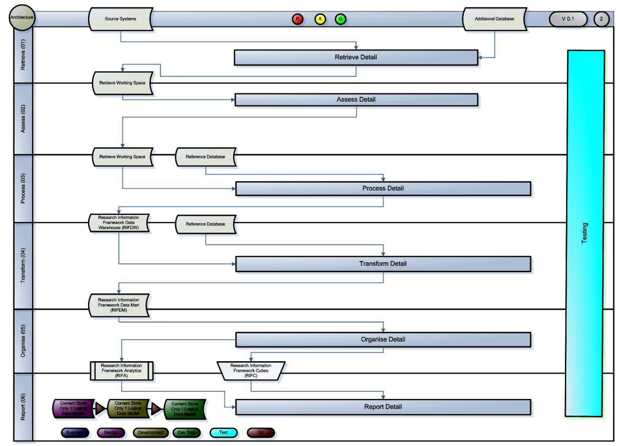

Figure 4. R-A-P-T-O-R process pipe

7.4.1 Audit work cell

7.4.2 Balance work cell

7.4.3 Control work cell

7.5 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:

7.5.1 Jobs work cell

7.5.2 Schedule work cell

7.5.3 Monitor work cell

7.5.4 Communication work cell

7.5.5 Alerts work cell

7.6 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:

7.6.1 Retrieve work cell

7.6.2 Assess work cell

7.6.3 Process work cell

7.6.4 Transform work cell

7.6.5 Organise work cell

7.6.6 Report work cell

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. Aldinucci, M. Torquati, and M. Meneghin. Fastflow: Efficient parallel streaming applications on multi-core. *arXiv preprint arXiv:0909.1187*, 2009.

[2] J. T. Black and S. L. Hunter. *Lean manufacturing systems and cell design*. Society of Manufacturing Engineers, 2003.

[3] T. Cheng and S. Podolsky. *Just-in-time manufacturing: an introduction*. Springer Science & Business Media, 1996.

[4] M. George, D. Rowlands, M. Price, and J. Maxey. The lean six sigma pocket toolbook. 2005.

[5] P. Hintjens. Ømq-the guide. *Online: http://zguide. zeromq. org/page: all, Accessed on*, 23, 2011.

[6] T. L. Sterling. *Beowulf cluster computing with Linux*. MIT press, 2002.