# Parallel patterns against an exabyte data lake using exascale heterogeneous computing


Mr Andreas Vermeulen
University of St Andrews
Saint Andrews, Fife KY16 9AJ
University of Dundee
Nethergate,Dundee DD1 4HN
a.f.vermeulen@dundee.ac.uk

Dr Vladimir Janjic
University of St Andrews
Saint Andrews, Fife KY16 9AJ
vj32@st-andrews.ac.uk

Mr Andy Cobley
University of Dundee
Nethergate, Dundee DD1 4HN
acobley@computing.dundee.ac.uk



## ABSTRACT

*An enhancement of a research information factory using exascale heterogeneous computing and parallel knowledge-extraction patterns to generate deep learning source.*



## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Theory, Framework, Application, Research, Hardware

## Keywords

*exabyte, exascale, knowledge-extraction, patterns, rapid information factory, RIF, RIFF, RIFC, heterogeneous computing, parallel patterns, mapr, cassandra, spark, opencl, R, fastflow, cuda, 3D torus network, deep learning, machine learning*


## 1. RESEARCH QUESTION

*Can a Rapid Information Factory using agile and lean six sigma principles to solve the effective and efficient exascale heterogeneous computing based processing of a million terabytes data lake into a value-add deep learning knowledge source?.*

## 2. RAPID INFORMATION FACTORY

### 2.1 Rapid Information Factory Framework

*The rapid information factory framework is a methodology that guides a exascale [2] heterogeneous computing cluster to process a exabyte data lake. The framework processes a billion billion calculations per second against one million terabytes of disk storage. The framework generates a series of factories that together can process the data lake using custom designed parallel processes.*

#### 2.1.1 Functional Layer

The functional layer is the layer that handles the functional processes within the cluster. This layer is the bulk of the framework as it contains the main components of the factory process.

**High-Level View**

*The high-level view of the Homogeneous Ontology for Recursive Uniform Schema (HORUS) shows the users the current status of the Rapid Information Factory. This is achieve by visualisation of the rapid information factory via a Rstudio Shiny [8] and R [11] based web site.*

**Synaptic Assimilator (SA)**

*The synaptic assimilator is an artificial intelligence [7] engine that performs the processes assigned to the system to the most effective and efficient method.*

*The artificial intelligence uses machine learning as a investigation and testing method to improve and select the correct combination of processing artifacts to achieve the efficient and effective outcome.*

#### Exascale Data Lake

*The exascale data lake is a data source that is larger than thousand petabytes or one million terabytes or one billion gigabytes.*

#### Persistent Recursive Information Schema Manipulator (PRISM)

*The persistent recursive information schema manipulator is the central control framework for each data procesing flow through the system using a bulk synchronous parallel (BSP) abstract computer as a bridging model for rapid information factory's parallel algorithms that are pre-defined and tested by the factory.*

#### RAPTOR Supersteps

*The RAPTOR framework is the supersteps of the bulk synchronous parallel (BSP) based process. The framework uses basic building blocks like pipeline, farm and loopback to formulate more complex structures to handle the data requirements.*

*The supersteps are:*

#### Retrieve Superstep

*The retrieve superstep is responsible for data retrieval from other data sources.* *(See Retrieve Superstep for more details)*

#### Assess Superstep

*The assess superstep is responsible for the data validation in the factory.* *(See Assess Superstep for more details)*

#### Process Superstep

*The process superstep is responsible for the processing of the data into a data vault that keeps full record of the different phases that the data is process over time.* *(See Process Superstep for more details)*

#### Transform Superstep

*The transform superstep is responsible for transforming the data lake into a business formatted data warehouse.* *(See Transform Superstep for more details)*

#### Organise Superstep

*The organise supersteo is responsible to orginise data sets together for each business group.* *(See Organise Superstep for more details)*

#### Report Superstep

*The report superstop is responsible to perform the reporting requirements.* *(See Report Superstep for more details)*

### *2.1.2 Operational Management Layer*

#### Autonomous Node Transport (ANT) Definitions

*The autonomous node transport definitions are the set of cloud instances or physical servers that support the processing capability of the factory. The HORUS schema keeps a series of characteristics required by the factory to decide which ANT to use.*

*The nodes are truly heterogeneous computing supporting combinations of processors that covers the range from high-end servers and high-performance computing machines all the way down to low-power embedded devices like mobile phones and tablets.*

#### Autonomous Node Transport Management

*The autonomous node transport management oversees the complete process of running the systems.*

#### Monitoring

*Monitoring handles the monitoring tasks in the system.*

#### Persistent Uniform Protocol Agreement (PUPA) Definitions

*The persistent uniform protocol agreement definitions are the collection of the algorithmic skeletons within the system.The PUPAs are programs generate using other frameworks like OpenCL [9], ArrayFire [5], Spark [4], Titan graph database [10] [6], FastFlow Framework [1].*

#### Persistent Uniform Protocol Agreement (PUPA) Management

*The persistent uniform protocol agreement management oversees the complete collection.*

#### Alerting

*The alerts are manage from this singular point in the system.*

#### Parameters

*The parameters are stored in this singular place in the system.*

#### Scheduling

*The scheduling handles the schedules from this singular point in the system.*

#### Communication

*Communication handles the communication into and from the system from this singular point.*

### *2.1.3 Audit, Balance and Control Layer*

#### Work Cells

*The work cells [3] is the basic building block of the processing system.*

#### Execution Statistics

*The execution statistics is the basic performance recording system.*

#### Remote Yoke

*The Poka-yoke is term that means "mistake-proofing". The Remote York is the rapid information factory's basic monitoring interface between the different work cells.*

#### Rejections and Error Handling

*The rejections and error handling in the rapid information factory handles the rejections and error handling within the system.*

*Balancing and Control*

*The balance and control mechanisms are the execute from the singular section.*

*Codes Management*

*The code management is the single section that holds the standard codes used in the system.*

*Stadard codes includes ISO codes, pre-agreed names and known lists of items for the factory.*

*The use of standard codes enable the effective deep data mining prescribed as output of the factory.*

### *2.1.4 Business Layer*

*Functional Requirements*

*A functional requirement defines a function of a system and its components. A function is described as a set of inputs, the behavior, and outputs. [?]. The set of requirements together as a unit describes the factory processing rules.*

*Non-functional Requirements*

*A non-functional requirement is a requirement that specifies criteria that can be used to test the operation of a factory, rather than specific behaviors of the process within the factory.The set of requirements together as a unit describes the factory verification rules.*

*The following are types of non-functional requirements*

- Accessibility
- Audit and control
- Availability
- Backup
- Capacity (current and forecast)
- Certification
- Compliance
- Configuration management
- Dependency on other parties
- Deployment
- Documentation
- Disaster recovery
- Efficiency (resource consumption for given load)
- Effectiveness (resulting performance in relation to effort)
- Emotional factors (like fun or absorbing or has "Wow! Factor")
- Environmental protection
- Escrow
- Exploitability
- Extensibility (adding features, and carry-forward of customizations at next major version upgrade)
- Failure management
- Fault tolerance (e.g. Operational System Monitoring, Measuring, and Management)
- Legal and licensing issues or patent-infringement-avoidability
- Interoperability
- Maintainability
- Modifiability
- Network topology
- Open source
- Operability
- Performance / response time (performance engineering)
- Platform compatibility
- Price
- Privacy
- Portability
- Quality (e.g. faults discovered, faults delivered, fault removal efficacy)
- Recovery / recoverability (e.g. mean time to recovery - MTTR)
- Reliability (e.g. mean time between failures - MTBF, or availability)
- Reporting
- Resilience
- Resource constraints (processor speed, memory, disk space, network bandwidth)
- Response time
- Reusability
- Robustness
- Safety or Factor of safety
- Scalability (horizontal, vertical)
- Security
- Software tools
- Stability
- Standards
- Supportability
- Testability
- Usability by user community
- User Friendliness

### 2.1.5 Utility Layer

*The utility layer stores processing structures across the factory for general or common requirements.*

*Maintenance Utilities*

*The maintenance untilities are procesing structures that perform work for the factory to maintain*

*Data Utilities*

*Spesific Utilities*

**Autonomous Logical Agreement Transport Executor (ALATE)**

*The autonomous logical agreement transport executor is a special utility that ...*

**Rapid Artifical Intelligence Data Extract Routine (RAIDER)**

*The rapid artifical intelligence data extract routine is a special utility that ...*

**Rapid Execute Artificial Protocol Engine for Routine (REAPER)**

*The rapid execute artificial protocol engine for routine is a special utility that ...*

**Sequencetial Converter into Ontology for Uniform Transport (SCOUT)**

*The sequencetial converter into ontology for uniform transport is a special utility that ...*

## 2.2 Functional Layer

### *2.2.1 Retrieve Superstep*

*The retrieve superstep uses a series of work cells with an assembly format that is .......*

### *2.2.2 Assess Superstep*

*The assess superstep uses a series of work cells with an assembly format that is .......*

### *2.2.3 Process Superstep*

*The process superstep uses a series of work cells with an assembly format that is .......*

### *2.2.4 Transform Superstep*

*The transform superstep uses a series of work cells with an assembly format that is .......*

### *2.2.5 Organise Superstep*

*The organise superstep uses a series of work cells with an assembly format that is .......*

### *2.2.6 Report Superstep*

*The report superstep uses a series of work cells with an assembly format that is .......*

## 2.3 Work Cells

*The remote work cells is the basic processing container of the rapid information factory.*

### *2.3.1 Monitor Work Cell*

*The monitor work cell consists of a persistent recursive information schema manipulator plus a remote assessment yoke for each processing work cell the spesific BSP flow requires in the rapid information factory*

### *2.3.2 Processing Work Cell*

*The processing work cell is a combination of a remote assessment yoke, an input persistent uniform protocol agreement, an autonomous node transport and an output persistent uniform protocol agreement. The remote assessment yoke communicates to the remote assessment yoke attached to the monitor work cell. The input persistent uniform protocol agreement holds the instructions to enable the work cell to import the data into the work cell. The output persistent uniform protocol agreement holds the instructions to enable the work cell to export the data from the work cell.The autonomous node transport supplies the processing power to execute the PUPA and the yoke instructions.*

### *2.3.3 Measure Work Cell*

*The measure work cell consists of an autonomous node transport that supplies the processing power and a measure agreement precision that supplies the tests to determine if the processing was successful..*

## 2.4 Rapid Information Factory Data Sources

### 2.4.1 Retrieve Data Sources

*The retrieve data sources are external data source that requires a spesial type of persistent uniform protocol agreement called a node extractor and schema transformer that supplies the data processing instructions to transform the extraernal data into HORUS compliant data structures. The additional data workspace supplies preloaded data to assist the retrieve superstep to load the data from the external data source to create the retrieve data workspace that is the main storage structure in HORUS any retrieve data loads.*

### 2.4.2 Assess Data Sources

*The assess data sources are a read only input from the retrieve data workspace, a reference data workspace that is a read only data source for supplying reference data for the assess procudures.Reference data can iclude lists of codes and description that are valid data or lookup data to enhance the quality of the data by adding extra information to the assess data. The assess data workspace is the main storage structure for HORUS data.*

### 2.4.3 Process Data Sources

*The process data sources are a read only input from the assess data workspace, a reference data workspace that is a read only data source for supplying reference data for the process procudures.Reference data can iclude lists of codes and description that are valid data or lookup data to enhance the quality of the data by adding extra information to the process data. The process data workspace is the main storage structure for HORUS data processed into a data vault containing hubs, links and satellites.*

### 2.4.4 Data Vault

*The Data Vault architecture offers a unique solution to data integration in the rapid information factory. The Data Vault is a detail oriented, historical tracking and uniquely linked set of normalised tables that support one or more functional areas of the factory that stores perfeactly as data island on top of the data lake structure to process unstructured and semi-structured data into structured data.*

*Benefits of Data Vault Modeling*

- Manage and Enforce Compliance to Sarbanes-Oxley, HIPPA, and BASIL II in factory
- Spot business data issues that were bot visible before the processing
- Rapidly reduce business cycle time for implementing changes
- Merge new business units into the organisation rapidly
- Rapid return-on-investment and Delivery of information to new star schemas
- Consolidate disparate data stores.
- Proper Master Data Management
- Implement and Deploy service-oriented architecture (SOA)
- Scale to exabytes of data
- SEI CMM Level 5 compliant (Repeatable, consistent, redundant architecture)
- Trace all data back to the source systems

### 2.4.5 Transform Data Sources

*The transform data sources are a read only input from the process data workspace, a reference data workspace that is a read only data source for supplying reference data for the transform procudures.Reference data can iclude lists of codes and description that are valid data or lookup data to enhance the quality of the data by adding extra information to the transform data. The transform data workspace is the main storage structure for HORUS data warehouse structure that supports any analytic inquiries.*

### 2.4.6 Organise Data Sources

*The organise data sources are read only input from the Tranform data workspace, the organise data workspace to handle any organise data manipulation, the rapid information framework for datamarts, the rapid information framework for analytics and the rapid information framework for cubes that is the main storage structures for the factory.*

### 2.4.7 Report Data Sources

*The report data sources are read only input from the rapid information framework for datamarts, the rapid information framework for analytics and the rapid information framework for cubes. The role based access contol security process enforces any role based security access to the data sources. The rapid information framework for visualistion handles the factory's visualisation requirements. The rapid information framework for exports are the export methord for the rapid information factory and formats the HORUS compliant data structures into external data formats via a persistent uniform protocol agreement..*

# 3. RAPID TEST FRAMEWORK

## 3.1 Unit testing

### 3.1.1 Static Testing

*YOKE Unit Testing*

*The YOKE unit testing enables the rapid information factory to test all the YOKE structures individually.*

## 3.2 Solution Testing

*The solution testing performance the testing of the solution.*

*Solution Testing Plan*

*The solution testing plan is the process description of how to test the solution as a complete factory.*

### 3.2.1 Link Testing

*Generate Link Test Data*

*Prepare data for each Link Test to match the spesific measure agreement precision instructions.*

*Execute Singular Link Test*

*Execute the Link Test instructions by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Execute Parallel Link Test*

*Execute the Link Test instructions in parallel by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

### 3.2.2 System Testing

*Generate System Test Data*

*Prepare data for each System Test to match the spesific measure agreement precision instructions.*

*Execute Singular System Test*

*Execute the System Test instructions by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Execute Parallel System Test*

*Execute the System Test instructions in parallel by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

### 3.2.3 Performance Testing

*Generate Performance Test Data*

*Prepare data for each Performance Test to match the spesific measure agreement precision instructions.*

*Execute Singular Performance Test*

*Execute the Link Performance instructions by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Execute Parallel Performance Test*

*Execute the Performance Test instructions in parallel by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Solution Completion Report*

*The solution completion report is the combined data for the solution testing.*

## 3.3 Acceptance Testing

*The acceptance testing performance the testing of the solution.*

*Acceptance Testing Plan*

*The acceptance testing plan is the process description of how to test the solution for acceptance by the users.*

### *3.3.1 Acceptance Testing*

*Generate Acceptance Test Data*

*Prepare data for each acceptance test to match the spesific measure agreement precision instructions.*

*Execute Singular Acceptance Test*

*Execute the acceptance test instructions by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Execute Parallel Acceptance Test*

*Execute the acceptance test instructions in parallel by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

### *3.3.2 Exploratory Parallel Testing*

*Generate Exploratory Parallel Test Data*

*Prepare data for each exploratory parallel test to match the spesific measure agreement precision instructions.*

*Execute Singular Exploratory Parallel Test*

*Execute the exploratory parallel test instructions by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Execute Parallel Exploratory Parallel Test*

*Execute the exploratory parallel test instructions in parallel by combining a remote assessment yoke, an appropiate autonomous node transport and the spesific measure agreement precision.*

*Acceptance Completion Report*

*The acceptance completion report is the combined data for the solution testing..*

# 4. PROCESS LAYER - DATA VAULT

## 4.1 Time-People-Object-Location-Event

### *4.1.1 Time (Hub)*

### *4.1.2 Time Details (Satellite)*

### *4.1.3 People (Hub)*

### *4.1.4 People Details (Satellite)*

### *4.1.5 Object (Hub)*

### *4.1.6 Object Details (Satellite)*

### *4.1.7 Location (Hub)*

### *4.1.8 Location Details (Satellite)*

### *4.1.9 Event (Hub)*

### *4.1.10 Event Details (Satellite)*

### *4.1.11 Time People (Link)*

### *4.1.12 Time Object (Link)*

### *4.1.13 Time Location (Link)*

### *4.1.14 Time Event (Link)*

### *4.1.15 People Object (Link)*

### *4.1.16 People Location (Link)*

### *4.1.17 People Event (Link)*

### *4.1.18 People People (Link)*

### *4.1.19 Object Location (Link)*

### *4.1.20 Object Event (Link)*

### *4.1.21 Object object (Link)*

### *4.1.22 Location Event (Link)*

### *4.1.23 Location location (Link)*

### *4.1.24 Event Event (Link)*

# 5. TRANSFORM LAYER - SUN MODELS

## 5.1 Dimensions

### *5.1.1 Type 0 Dimension*

The Type 0 method is passive as it only insert but never updates.

### *5.1.2 Type 1 Dimension*

The Type 1 method is active as it overwrites old with new data but keeps no track of historical data.

### *5.1.3 Type 2 Dimension*

The Type 2 method is active as it tracks historical data by creating multiple records for a given natural key and keeps track of period the values was valid.

### *5.1.4 Outrigger Dimension*

The outrigger dimension is ....

### *5.1.5 Bridge Dimension*

The bridge dimension is ...

### *5.1.6 Mini Dimension*

The mini dimension is ...

## 5.2 Facts

Facts are ....

### *5.2.1 Measures*

Measures are ...

### *5.2.2 Factless*

Factless facts are ...

## 6. SCHEDULE FRAMEWORK

### 6.1 Cycle Time

*Cycle time describes the time use to complete a specific task from start to finish.*

*The time is calculated as .....*

### 6.2 Value Stream

*Value stream mapping is the lean-management method for analysing the current state and designing a improved state for the series of activities.*

### 6.3 Value Added Procesing Time

*The value added processing time is the time the factory process the data to add value.*

### 6.4 Non Value Added Procesing Time

*The non value added processing time is the time the factory wastes while process the data.*

### 6.5 Production Lead Time

*Production Lead Time is the total time (Value Added Procesing and Non Value Added Procesing) use to execute the data through an entire value stream.*

### 6.6 Schedule Backlog

*The Schedule Backlog is all the required processing Persistent Uniform Protocol Agreements (PUPA) prioritised, ordered list, sorted by business value and risk. It contains the tasks to accomplish the RAPTOR flow. The Schedule Backlog often contains user stories covering functional requirements, non-functional requirements for the RAPTOR flow.*

#### *6.6.1 INVEST RAPTOR flow*

*Independent*

*RAPTOR flow must be independent.*

*Negotiable*

*A RAPTOR flow always negotiable. It is not an explicit contract for features; rather, details will be co-created by the customer and programmer during development.*

*Valuable*

*A RAPTOR flow must be valuable to business.*

*Estimable*

*A RAPTOR flow must be estimated.*

*Small*

*RAPTOR flow must be as small as possible.*

*Testable*

*A RAPTOR flow must be testable.*

#### *6.6.2 SMART Tasks*

*Specific*

*A RAPTOR flow task needs to be specific and tasks to add up to the full RAPTOR flow.*

*Measurable*

*The key measure is testability*

*Achievable*

*The task should be achievable.*

*Relevant*

*Every task should be relevant and contributing to the RAPTOR flow in the factory. Stories are divided into tasks for the achievement of RAPTOR flow.*

*Time-boxed*

*A task should be time-boxed: limited to a specific duration.*

## 6.7 Active Process Backlog

*The active process backlog consists of the committed process tasks attached to the scheduled PRISM controlled end-to-end RAPTOR flows.*

## 6.8 Active Process Work Cells

*The active process work cells is the combination of processing structures that is currently active in the rapid information factory. The quantity is determined by the available and required parallel processing units active in the factory.*

### *6.8.1 Process Set-up Time*

*The process set-up time is the time it takes the factory to construct the work cell and bring it online ready for processing data.*

### *6.8.2 Process Run Time*

*The process run time is the time the work cell uses to process the data against the spesific input and output PUPAs' algoritmes.*

### *6.8.3 Process Reset Time*

*The process reset time is the time it takes the factory to reset the spesific work cell ready for the next work cell.*

## 6.9 Verify Backlog

*The verify backlog is .......*

## 6.10 Active Verify Backlog

*The active process backlog consists of the committed verify tasks attached to the scheduled PRISM controlled end-to-end RAPTOR flows.*

## 6.11 Active Verify Work Cells

*The active verify work cells are .......*

### *6.11.1 Verify Set-up Time*

*The verify set-up time is .......*

### *6.11.2 Verify Run Time*

*The verify run time is .......*

### *6.11.3 Verify Reset Time*

*The verify reset time is .......*

## 6.12 Information Process Log

*The information process log is ........*

# 7. IMPROVEMENT PROCESSES

## 7.1 The 8 Wastes

### 7.1.1 Defects

*Defects are mistakes that require extra time, resources and money to reprocess.*

*The defects are the result of:*

- Poor quality control of processes
- Poor repairs on servers
- Poor documentation procedures
- Lack of standards
- Weak or missing processes
- A misunderstanding of customer requirements
- Poor inventory control
- Poor design of processes
- Undocumented design changes

### 7.1.2 Overproduction

*Overproduction is when too many of a spesific deliverable is delivered.*

*The factory would cause overproduction if too many ANT is prepared for running PUPAs:*

- Just-in-case production
- Unclear customer requirements
- Producing to a incorrect forecast
- Long set-up times
- Attempts to avoid long set-up times
- Poorly applied automation

### 7.1.3 Waiting

*Actual downtime that occurs whenever processing has to stop for some reason.*

*Causes of waiting can also include:*

- Mismatched production rates
- Very long set-up times
- Poor shop layout
- Insufficient staffing
- Work absences
- Poor communications

### 7.1.4 Non-Utilised Talent

*Non-Utilised Talent is the poor utilization of available talents, ideas, abilities and skill sets.*

- Lack of teamwork between staff members
- Lack of training in operations of the factory
- Poor communications between staff members and the factory
- Management's refusal to include employees in problem-solving
- Narrowly defined jobs and expectations for staff
- Poor management in general of staff

### 7.1.5 Transportation

*Transportation is the unnecessary moving data around within the factory.*

- Poor factory layout
- Excessive or unnecessary handling of data
- Misaligned process flow in the factory
- Poorly-designed factory PUPAs
- Unnecessary steps in factory processes

### 7.1.6 Inventory

*Lean are based on the practice of Just-In-Time production of data in the factory.*

*Excess inventory is caused by:*

- Overproduction
- Poor layout
- Mismatched production speeds
- Unreliable suppliers
- Long set-up times
- Misunderstood customer needs

### 7.1.7 Motion

*Excess motion is to move around too much and then causes the factory slow down significantly.*

*Causes of excessive motion include:*

- Poor workstation/shop layout
- Poor housekeeping
- Shared tools and machines
- Workstation congestion
- Isolated operations
- Lack of standards
- Poor process design and controls

### *7.1.8 Extra-processing*

*Excess Processing is any unnecessary effort expended in order to complete a task: double-handling data, seeking permission during processing, unnecessary processing steps, unnecessary data useage, re-entering data, making too many copies of data.*

*Excess Processing arises from:*

- Poor process control
- Lack of standards
- Poor communication
- Overdesigned equipment
- Misunderstanding of the customer's needs
- Human error
- Producing to forecast

## 7.2 Plan-Do-Act-Check Improvement Process

### *7.2.1 Plan*

*Establish the objectives and processes necessary to deliver results in accordance with the expected output (the target or goals). By establishing output expectations, the completeness and accuracy of the spec is also a part of the targeted improvement. When possible start on a small scale to test possible effects.*

### *7.2.2 Do*

*Implement the plan, execute the process, make the product. Collect data for charting and analysis in the following "CHECK" and "ACT" steps.*

### *7.2.3 Check*

*Study the actual results (measured and collected in "DO" above) and compare against the expected results (targets or goals from the "PLAN") to ascertain any differences. Look for deviation in implementation from the plan and also look for the appropriateness and completeness of the plan to enable the execution, i.e., "Do". Charting data can make this much easier to see trends over several PDCA cycles and in order to convert the collected data into information. Information is what you need for the next step "ACT".*

### *7.2.4 Act*

*If the CHECK shows that the PLAN that was implemented in DO is an improvement to the prior standard (baseline), then that becomes the new standard (baseline) for how the organization should ACT going forward (new standards are enACTed). If the CHECK shows that the PLAN that was implemented in DO is not an improvement, then the existing standard (baseline) will remain in place. In either case, if the CHECK showed something different than expected (whether better or worse), then there is some more learning to be done... and that will suggest potential future PDCA cycles. Note that some who teach PDCA assert that the ACT involves making adjustments or corrective actions... but generally it would be counter to PDCA thinking to propose and decide upon alternative changes without using a proper PLAN phase, or to make them the new standard (baseline) without going through DO and CHECK steps.*

## 7.3 Define-Measure-Analyse-Improve-Control Improvement Process

### *7.3.1 Define*

*The purpose of this step is to clearly articulate the business problem, goal, potential resources, project scope and high-level project timeline. This information is typically captured within project charter document.*

### *7.3.2 Measure*

*The purpose of this step is to objectively establish current baselines as the basis for improvement. This is a data collection step, the purpose of which is to establish process performance baselines.*

### *7.3.3 Analyse*

*The purpose of this step is to identify, validate and select root cause for elimination.*

### *7.3.4 Improve*

*The purpose of this step is to identify, test and implement a solution to the problem; in part or in whole.*

#### *7.3.5 Control*

*The purpose of this step is to sustain the gains. Monitor the improvements to ensure continued and sustainable success. Create a control plan.*

### 7.4 Lean Six Sigma: 5S

#### *7.4.1 Sort*

*The first step is to go through all equipment and materials and determine what must be retained at the worksite. Only essential tools, aids, equipment, and so on are allowed to remain.*

#### *7.4.2 Set in Order*

*âĂIJA place for everything, and everything in its place.âĂİ*

#### *7.4.3 Shine*

*To help maintain the order youâĂŹve created, thoroughly clean everything remaining at the worksite.*

#### *7.4.4 Standardise*

*Where possible, make worksites consistent. All workstations for a particular job should be identical so that someone from another worksite can immediately step in and productively run the process if necessary.*

#### *7.4.5 Systematise*

*This final step means to put a schedule and system in place for maintaining and refreshing the 5S-ed worksite..*

### 7.5 Rapid Information Factory Cluster (RIFC)

#### *7.5.1 3D Torus Network Framework*

#### *7.5.2 MapR Data Lake*

*The rapid information factory cluster is a bulk synchronous parallel (BSP) engine. The cluster consisting of two hunderd thousand amazon cloud nodes (d2.8xlarge with thirty six processing cores, two hunderd forty four gigabyte memory and twenty four two thousand gigabyte hard disks) to support billion billion calculations per second against one million terabytes of disk storage.*

*The hunderd amazon graphical enhanced nodes (g2.8xlarge with four GPUs each with one thousand five hunderd CUDA cores, four gigabyte of video memory, thirty two cpus, sixty gigabyte memory and two hunderd and forty gigabyte solid state drives.)*

#### *7.5.3 Titan Graph Data Lake*

*The two hunderd amazon graphical enhanced nodes (g2.8xlarge with four GPUs each with one thousand five hunderd CUDA cores, four gigabyte of video memory, thirty two cpus, sixty gigabyte memory and two hunderd and forty gigabyte solid state drives.)*

#### *7.5.4 Cassandra Data Lake*

*The five hunderd amazon graphical enhanced nodes (g2.8xlarge with four GPUs each with one thousand five hunderd CUDA cores, four gigabyte of video memory, thirty two cpus, sixty gigabyte memory and two hunderd and forty gigabyte solid state drives.).*

# 8. EXPERIMENTS

# 9. RESULTS

# 10. REFERENCES


[1] Aldinucci, M., Danelutto, M., Kilpatrick, P., Meneghin, M., and Torquati, M. Accelerating code on multi-cores with fastflow. In *Euro-Par 2011 Parallel Processing*. Springer, 2011, pp. 170–181.

[2] Bergman, K., Borkar, S., Campbell, D., Carlson, W., Dally, W., Denneau, M., Franzon, P., Harrod, W., Hill, K., Hiller, J., et al. Exascale computing study: Technology challenges in achieving exascale systems. *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep 15* (2008).

[3] Feld, W. M. *Lean manufacturing: tools, techniques, and how to use them.* CRC Press, 2000.

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

[5] Malcolm, J., Yalamanchili, P., McClanahan, C., Venugopalakrishnan, V., Patel, K., and Melonakos, J. Arrayfire: a gpu acceleration platform. In *SPIE Defense, Security, and Sensing* (2012), International Society for Optics and Photonics, pp. 84030A–84030A.

[6] Mishra, V. Titan graph databases with cassandra. In *Beginning Apache Cassandra Development*. Springer, 2014, pp. 123–151.

[7] O'Leary, D. E. Artificial intelligence and big data. *IEEE Intelligent Systems*, 2 (2013), 96–99.

[8] Ortega, F., Moguerza, J. M., and Cano, E. L. Combining r and python for scientific computing. In *The R User Conference, useR! 2013 July 10-12 2013 University of Castilla-La Mancha, Albacete, Spain* (2013), vol. 10, p. 92.

[9] Stone, J. E., Gohara, D., and Shi, G. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering 12*, 1-3 (2010), 66–73.

[10] Tanase, I., Xia, Y., Nai, L., Liu, Y., Tan, W., Crawford, J., and Lin, C.-Y. A highly efficient runtime and graph library for large scale graph analytics. In *Proceedings of Workshop on GRAph Data management Experiences and Systems* (2014), ACM, pp. 1–6.

[11] Team, R. C. R language definition, 2000.