SME-LET Announcement of Opportunities 2009: **Cal/Val and User Services - Calvalus**

Final Report
Version 1.0
31. October 2011

Prepared by
Norman Fomferra

Change Log

Version	Date	Revised by	Change	Authors
1.0 draft	12.10.2011	-	The initial version of this document.	N. Fomferra
1.0	31.10.2011	E. Kwiatkowska	Added new chapter and two figures that describe MapReduce better. Improved general readability, comprehensibility and comprehensiveness of various production scenarios using Hadoop. Included references to Calvalus documentation.	N. Fomferra

Contact

Brockmann Consult GmbH

Max-Planck-Str 2

21502 Geesthacht, Germany

www.brockmann-consult.de

info@brockmann-consult.de

Tel +49 4152 889300

Fax +49 4152 889333

Contact person

Norman Fomferra

norman.fomferra@brockmann-consult.de

Tel +49 4152 889303

Table of Contents

1 Executive Summary

1.1 Objective

ESA's Earth Observation (EO) missions provide a unique dataset of observational data of our environment. Calibration of the measured signal and validation of the derived products is an extremely important task for efficient exploitation of EO data and the basis for reliable scientific conclusions. In spite of this importance, the cal/val work is often hindered by insufficient means to access data, time consuming work to identify suitable in-situ data matching the EO data, incompatible software and limited possibilities for rapid prototyping and testing of ideas. In view of the future fleet of satellites and the fast-growing amount of data produced, a very efficient technological backbone is required to maintain the ability of ensuring data quality and algorithm performance.

The announcement of opportunities *EO Cal/Val and User Services* is a technology study of the ESA LET-SME 2009 call, investigating into an existing leading edge technology (LET) for their applicability in the EO domain. Specifically,

> *LET-SME is a **spin-in** instrument encouraging the participation of SMEs to ESA technology. The LET-SME focuses on early stage development of "Leading Edge Technologies", i.e. the ones likely to become the reference technologies for the near future, and have good chances of being infused into ESA projects and missions.*

In accordance with the SoW, Calvalus is a system that has been proposed to fully support the idea of LET-SME, thus with a strong focus on a selected LET which is described in this report.

1.2 Approach

Brockmann Consult GmbH proposed to develop a demonstration processing system based on the ***MapReduce*** *programming model* (MR) combined with a ***Distributed File System*** (DSF). The basic approach was first published in 2004 by the two Google computer scientists J. Dean and S. Ghemawat [RD-4]. The technology has been designed for processing of ultra large amounts of data and is based on massive parallelisation of tasks combined with a distributed file system, both running on large, extendible clusters solely comprising commodity hardware. All nodes in the cluster are equally configured and provide both disk storage and CPU power. Well known online services provided by Google, Yahoo, Amazon and Facebook rely on this technology. Its spin-in application to space born, spatial data is feasible and pertinent. The results of this study demonstrate that the processing of large amounts of EO data using MR and a DSF is efficient and advantageous.

The demonstration system, **Calvalus**, basically comprises a cluster of 20 commodity computers with a total disk capacity of 112 TB at a total cost of **30 k€**. The processing system software is based on ***Apache Hadoop*** – an open-source implementation of MR and DSF in Java.

Calvalus gains its performance from massive parallelisation of tasks and the data-local execution of code. Usual processing clusters or grids first copy input data from storage nodes to compute nodes, thereby introducing I/O overheads and network transfer bottlenecks. In Calvalus, processing code is executed on cluster nodes where the input data are stored. Executable code can be easily deployed; the code distribution and installation on all cluster nodes is done automatically. Multiple versions of processing code can be used in parallel. All these properties of the Calvalus system allow users to

efficiently perform cal/val and EO data processing functions on whole mission datasets, thus allowing an agile product development and fast improvement cycles.

The different production scenarios and analyses implemented in Calvalus are inspired by the needs of the current ESA projects, such as *CoastColour* and *Climate Change Initiative* (CCI) for Land Cover and Ocean Colour, both of strong interest to an international user community. The implementation is focused on ocean colour:

1. L2-Production: Processing of Level-1b radiance products to Level-2 ocean reflectances and inherent optical property (IOP) products.
2. L3-Production: Processing of Level-1b and Level-2 products to spatially and temporally aggregated Level-3 products.
3. Match-up analysis: Processing of Level-1b data extracts and generation of match-up plots for Level-2 product validation with in-situ data.
4. Trend analysis: Generation of time-series of data extracts and plots from Level-3 products processed from Level-1b and Level-2 data.

The Level-2 products in production scenarios 2 to 4 are generated on-the-fly from Level-1b using selected Level-2 processors and their required versions, processing parameters and LUTs. The Calvalus demonstration system currently holds the full mission Envisat MERIS Level-1b RR data as well as all MERIS Level-1b FR product subsets required by the CoastColour project.

Calvalus has a web front-end that allows users to order and monitor productions according the to four production scenarios, and to download the results. It also offers a Java production API, allowing developers to write new production scenarios.

1.3 Project Activities

This project has been performed in two phases. Phase I was dedicated to requirements engineering and feasibility studies. In order to gather feedback, the project has been presented to a number of potential users including presentations to the Envisat MERIS QWG and on the ESA Living Planet Symposium in Bergen. In the first phase, Brockmann Consult has also performed technology studies during which key technologies have been experimentally tested for their applicability. A contact to the Hadoop developers (Cloudera) has been established in order to discuss various technical approaches. Phase 2 was dedicated to the realisation of a demonstration system. The system architecture has been established, the cluster hardware has been set-up, and Calvalus software has been developed.

The Calvalus study has been carried out in the time from January 2010 to October 2011. The following table summarises the work that has been performed:

January 2010	Evaluation of Apache Hadoop and alternative systems, e.g. Oracle Grid Engine
February 2010	Requirements analysis. Hadoop test cluster setup (5 desktop nodes). Performance analysis with various data storage formats. Experiments with various Hadoop APIs.
April 2010	First processing on the 5 node test cluster. Analyze Hadoop performance metrics and reduce data traffic. Presentation to ESA GECA project and MERIS Validation Team (MVT).
May 2010	Completed Requirements Baseline. Completed technology study.
June 2010	Prepared and delivered Technical Specification draft.

	Presented first results at ESA Living Planet Symposium in Bergen.
July 2010	Definition of an intermediate EO data format to be used with HDFS.
August 2010	Procured hardware for a 20-nodes demonstration cluster.
September 2010	Hardware setup of the demonstration cluster. Prepared a proposal for T-Systems cluster.
October 2010	Hardware setup of the demonstration cluster. Delivered final version of the Technical Specification. Performance analysis of different data storage and processing configurations.
November 2010	Implementation of Level 3 binning algorithms utilizing the map/reduce method.
December 2010	Implemented simple command-line interface for submission of jobs. Mid Term Review Meeting, presentation of first Level-2 and Level-3 processing results.
January 2011	Implementation of processing system core. Added ability to execute any shell executables on Hadoop.
February 2011	Implemented L2 and L3 processing workflows and staging sub-system.
March 2011	Developed first version of the Calvalus portal, the web frontend. Deployed portal onto public application server. Released intermediate version 0.1.
April 2011	Released intermediate version 0.2: Implemented trend analysis workflows.
June 2011	Released intermediate version 0.3: Implemented region management functionality.
August 2011	Released final version 1.0: Implemented match-up analysis.
September 2011	Preparation and delivered acceptance test plan. Carried out acceptance tests.
October 2011	Prepared and delivered final report.

Table 1: Study activities

The Calvalus team is

- Dr Martin Böttcher, Brockmann Consult GmbH – Developer
- Olga Faber, Brockmann Consult GmbH – Tester
- Norman Fomferra, Brockmann Consult GmbH – Project manager / Developer
- Dr Ewa Kwiatkowska, ESA – Project initiator / Technical ESA representative
- Marco Zühlke, Brockmann Consult GmbH – Developer

1.4 Documentation

All deliverables documents of the Calvalus study can be downloaded from the Calvalus web page www.brockmann-consult.de/calvalus. The documents are:

- Requirements Baseline [RD 22]
- Technical Specification [RD 23]
- Acceptance Test Plan [RD 24]
- Final Report (this document)

2 Context and Application Domain

Calibration of the measured EO sensor signal, algorithm development and validation of the derived data products are extremely important tasks for efficient exploitation of the EO data and the basis for reliable scientific conclusions. In spite of this importance, the cal/val work is often hindered by insufficient means to access and process data, time consuming work to match suitable in-situ and other EO data, incompatible software and no possibility for rapid prototyping and testing of ideas.

The goal of Calvalus is to apply a leading-edge technology to develop an efficient processing and analysis system for EO satellite data. The technology manages massive EO datasets and provides a large-scale, efficient, rapid, on-the-fly processing power to test concepts concerning instrument on-orbit characterization and its science algorithms. The instant feedback on the ideas enables rapid prototyping and idea transfer to operations.

In Figure 1 the development cycle is shown, that is run through in order to improve the quality and consistency of EO data products.

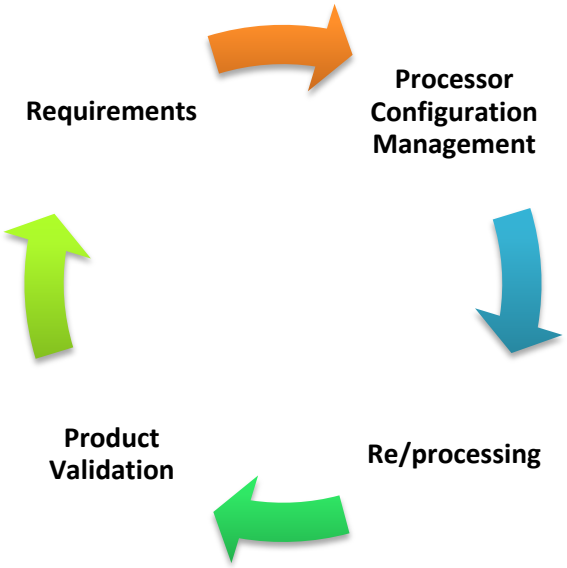

Figure 1: Development cycle for the improvement of EO data products

Requirements on higher level and value-added products originate from their user communities. They drive the initial development of instruments and of algorithms capable of producing the required data products. The products are generated by data processors which implement the algorithms that are used to transform the lower level (L1) input data to the higher level products. The resulting L2 and L3 products are then subject to validation, usually performed by scientists. The specific cal/val analyses include the processing of extracts corresponding to match-ups with ground observations and vicarious calibration sites as well as the processing of mission-long sampled global or regional time series to evaluate the quality of long-term data records. Typically, the validation results generate new requirements on the processor configuration management, for example,

- updates in instrument calibration,
- modified parameterisation in terms of processing parameters and auxiliary data,
- algorithm (science code) adjustments,
- and implementation of new algorithms

The ultimate goal of Calvalus is to accelerate the development cycles by allowing users to perform repeated processing of the same primary inputs with different algorithms or parameters and performing automated analyses on the resulting dataset. The validation activities such as

- comparisons with reference data,
- inter-comparisons with other sensors,
- and detection of trends and anomalies

are supported by two automated standard analyses, namely the match-up and trend analyses.

The Calvalus study is envisioned to assist instrument quality working groups, validation teams, and ESA projects such as CoastColour [RD 12] and Climate Change Initiative (Ocean_Colour_cci) [RD 15]. This context is described in the following sections.

2.1 Data Quality Working Groups

ESA has established the Data Quality Working Groups (DQWG) after the completion of Envisat's commissioning phase in fall 2002. The mission of the data quality working group is to monitor the quality of the instrument products as generated by the satellite ground segments, and to recommend algorithm improvements including suggestions for new products. DQWGs exist for MERIS, AATSR and Atmospheric Composition instruments.

The DQWGs are composed of scientists being expert (or even developer) of the science algorithms, and technical experts on algorithm implementation. The main tool of the DQWGs is the instrument data processor prototype. Algorithm changes, auxiliary data changes and ideas for new products are prototyped in this environment and tested before they are proposed for implementation in the operational ground segment processor. An efficient and robust tool like Calvalus provides an opportunity to the DQWGs to process massive amounts of data and to obtain an instantaneous feedback on the proposed improvements.

2.2 Instrument Validation Teams

After the launch of Envisat, the MERIS and AATSR Validation Team (MAVT) and the Atmospheric Chemistry Validation Team (ACVT) were established. The activities and lifetime of these validation teams were linked with the Envisat commissioning phase. However, the MERIS Validation Team was reactivated in 2009 in order to further support the work of the MERIS DQWG, in particular for the validation of the Case2 Water processing. A calibration and validation team was also implemented for the SMOS mission. This group started its activities with the advent of SMOS data products in late 2009. The main goal of the validation teams is to obtain in situ and other reference observations to provide independent evaluation of Envisat data streams and to improve existing algorithms and develop new ones. The teams also maintain and evolve respective in situ measurement and validation protocols. Calvalus can support these groups with quick data evaluation and algorithm development cycles.

2.3 ESA Climate Change Initiative

The key objective of the Climate Change Initiative (CCI) is to provide best quality long-term time series of Essential Climate Variables (ECV). The CCI consortia have the task to review L1 processing including calibration and geo-location, and all Level 2 processing algorithms. Where necessary and possible, new and better algorithms than the standard ones (those used in the ground segment processors) can be deployed and error estimates are aimed to be added to derived variables. The results from the CCI project should then feed back to improvements in the ground segment

processors. Phase 2 of CCI of the ECV projects is concerned with the future operational implementation of the ECV processors and with the systematic and automated validation of the products, including reprocessing. The CCI projects started in late 2010. The powerful data processing capabilities of Calvalus are already exploited in Ocean Colour and Land Cover parts of the CCI.

2.4 CoastColour

The CoastColour project kicked-off in January 2010 and will last until 2011. This project is contributing a coastal component to the CCI as coastal waters are excluded from the ECV-OC statement of work with reference to CoastColour. The requirements on product quality and on critical review of L1 and L2 processing are identical in CoastColour and ECV-OC.

There are several key requirements in CoastColour on validation:

- Definition of a standard set of products from different satellite missions; primary interest is on MERIS but MODIS and SeaWiFS are considered for comparison
- Compilation of an in-situ database with reference data to be used for validation of the standard products (point 1 of this list)
- Definition of standard tests to be applied to the standard products after algorithm changes, and for inter-comparison of different products and algorithms
- Frequent repetition of the tests upon algorithm changes
- Keeping history of algorithm changes and processing versions
- Automated processing of the tests and evaluation
- Transparency of the process through an open, web based system

These CoastColour requirements perfectly match the objectives of Calvalus. The coincident aspects are as follows:

- the instrument concerned: MERIS,
- link with the international community, IOCCG, CEOS WGCV, UNFCCC
- the perspective of continuity within the CCI and ESA DQWGs
- timing in parallel with Calvalus and the results expected in line with the Calvalus schedule

CoastColour L2 processing is therefore an ideal candidate to be linked with Calvalus. The L2 processing consists of an atmospheric correction based on a neural network inversion of the radiative transfer equation, and of an extraction of inherent optical properties of in-water constituents using two methods: a neural network and a semi-analytical approach. The neural network inversions are realised by the GKSS Case2R scheme. The semi-analytical algorithm uses the Quasi-Analytical Algorithm (QAA) from Mississippi State University (Zhongping Lee).

An important component of the CoastColour project is an inter-comparison with standard MERIS processing, SeaWiFS and MODIS products, as well as with in-situ data. This links the CoastColour processing with standard MERIS processing, and with NASA standard processing. Scientists from MERIS QWG (Doerffer, Fischer, Brockmann) and from NASA (Franz, Feldman) are contributing to this inter-comparison.

2.5 ESA Sentinel Missions and the Future

ESA's Living Planet programme is currently in a transitional phase, characterised by the maturity of ENVISAT, the preparation of the future operational Sentinel missions, and by the growing number of Earth Explorer missions. The successful work of the DQWGs will further evolve to meet the new challenges. The European FP7 programme includes validation activities in its R&D projects (e.g.

MyOcean validation, Aquamar Downstream project has two work packages on validation and validation technique evolution). The recently extended ESA GSE projects (e.g. MarCoast) also incorporate important activities on validation. All these undertakings are preparing the future for the operational calibration and validation of the ESA Sentinel missions, and for the scientific Earth Explorer missions. The immediate steps are the ESA CCI and the projects contributing to it today, such as CoastColour.

The aim of Calvalus is to also prepare for the future and to test novel concepts that support large-scale calibration and validation activities. The project has learned from today's cal/val needs and limitations and has linked with the prospective ESA projects in order to develop a technological base for their work. Focusing on the CCI and developing major technical concepts using the CoastColour as an example have been the basis for achieved these primary goals.

3 Technical Approach

3.1 Hadoop Distributed Computing

The basis of the Calvalus processing system is **Apache Hadoop**. Hadoop is an industry proven open-source software capable of running clusters of tens to ten thousands of computers and processing ultra large amounts of data based on massive parallelisation and a distributed file system.

3.1.1 Distributed File System (DFS)

In opposite to a local file system, the Network File System (NFS) or the Common Internet File System (CIFS), a distributed file system (DFS) uses multiple nodes in a cluster to store the files and data resources [RD-5]. A DFS usually accounts for transparent file replication and fault tolerance and furthermore enables data locality for processing tasks. A DFS does this by subdividing files into blocks and replicating these blocks within a cluster of computers. Figure 2 shows the distribution and replication (right) of a file (left) subdivided into three blocks.

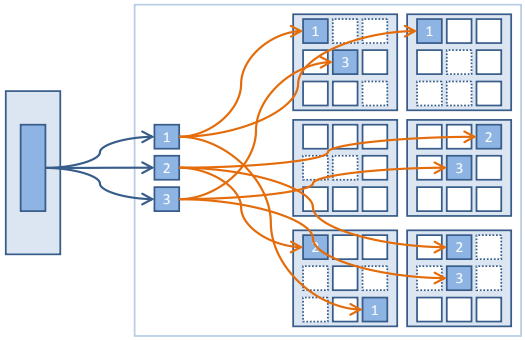

Figure 2: File blocks, distribution and replication in a distributed file system

Figure 3 demonstrates how the file system handles node-failure by automated recovery of under-replicated blocks. HDFS further uses checksums to verify block integrity. As long as there is at least one integer and accessible copy of a block, it can automatically re-replicate to return to the requested replication rate.

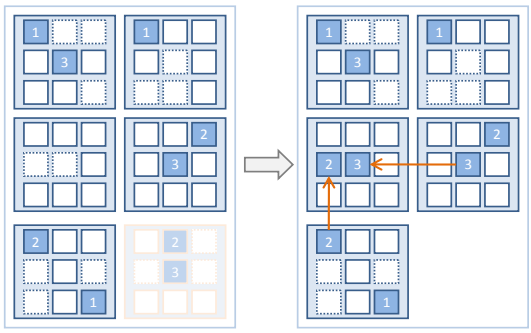

Figure 3: Automatic repair in case of cluster node failure by additional replication

Figure 4 shows how a distributed file system re-assembles blocks to retrieve the complete file for external retrieval.

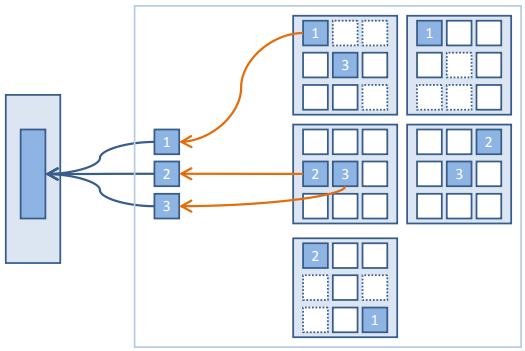

Figure 4: Block assembly for data retrieval from the distributed file system

3.1.2 Data Locality

Data processing systems that need to read and write large amounts of data perform best if the data I/O takes place on local storage devices. In clusters, where storage nodes are separated from compute nodes, two situations are likely:

1. Network bandwidth is the bottleneck, especially when multiple tasks work in parallel on the same input data but from different compute nodes and when storage nodes are separated from compute nodes.
2. Transfer rates of the local hard drives are the bottleneck, especially when multiple tasks are working in parallel on single (multi-CPU, multi-core) compute nodes.

A solution to these problems is to first use a cluster whose nodes are both, compute and storage nodes. Secondly, it is to distribute the processing tasks and execute them on the nodes that are "close" to the data, with respect to the network topology (see Figure 5). Parallel processing of inputs is done on *splits*. A split is a logical part of an input file that usually has the size of the blocks that store the data, but in contrast to a block that ends at an arbitrary byte position, a split is always aligned at file format specific record boundaries (see next chapter, step 1). Since splits are roughly aligned with file blocks, processing of input splits can be performed data-local.

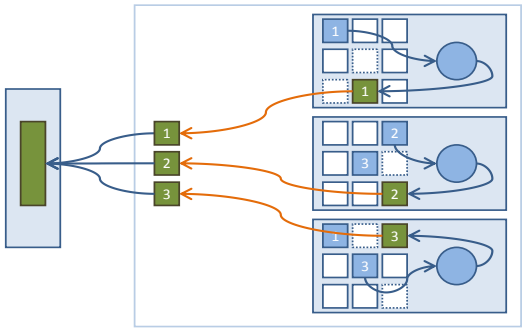

Figure 5: Data-local processing and result assembly for retrieval

3.1.3 MapReduce Programming Model

The MapReduce programming model has been published in 2004 by the two Google scientists J. Dean and S. Ghemawat [RD 4]. It is used for processing and generation of huge datasets on clusters for certain kinds of distributable problems. The model is composed of a *map function* that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce function* that merges all intermediate values associated with the same intermediate keys. Many real world problems can be expressed in terms of this model and programs written in this functional style can be easily parallelised.

The execution model for programs written in the MapReduce style can be roughly characterised by three steps:

1. Split input
    a. input → *split* → {split}
2. Mapper task: process input split
    a. split → *read* → {<k_1, v_1>}
    b. <k_1,v_1> → *map & partition* → {<k_2,v_2>}
3. Reducer task: process mapper output
    a. {<k_2,v_2)>} → *shuffle & sort* → {<k_2,{v_2}>}
    b. <k_2,{v_2}> → *reduce* → (k_3,v_3)
    c. {<k_3,v_3>} → *write* → output

The steps are explained by using the popular word-count example, a MapReduce implementation of an algorithm used to count the occurrences of words in text files. There may be N_M mapper tasks (step 2) and N_R reducer tasks (step 3) executed in parallel.

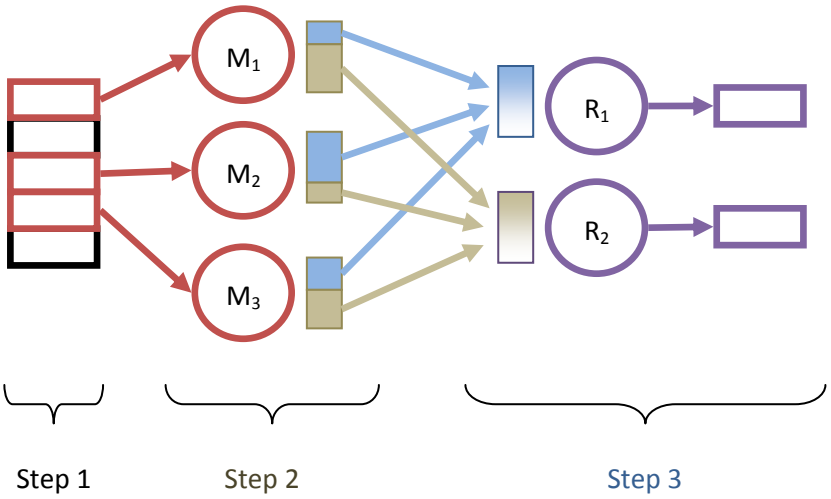

Figure 6: MapReduce execution model

Step 1: A usually very large input file is subdivided into a number of logical "splits". Each split starts and ends at *record* boundaries specific to a given file format. In the case of text files, a record may represent a single line and thus, splitting occurs at line endings.

Step 2: Each split is passed as input to a mapper task. Up to N_M mapper task may run in parallel. The mapper tasks reads the split and converts it into a vector of key-value pairs {<k_1,v_1>} (step 2.a). If the input is a text file, the key k_1 could be the line number and the value v_1 the line text. Each input pair <k_1,v_1> is then passed to the user supplied map function which transforms it into zero, one or more intermediate key-value pairs {<k_2,v_2>} (step 2.b).

In the word-count example, the text line v_1 is split into words. Each word becomes a key k_2, with the value v_2 being the integer 1 and meaning one occurrence.

Step 3: N_R reducer tasks run in parallel. Each reducer gets one or more specific partitions of the output of a mapper. The partition number ranging from 1 to N_R is computed from each intermediate key-value pair <k_2,v_2> by using a partitioning function (usually a hash function of k_2). This step is already performed by mapper tasks (step 2.c). Each reducer task reads all the intermediate key-value pairs {<k_2,v_2>} of its partitions of all mappers, merges and sorts them by key k_2 (step 3.a). All values v_2

that have same keys k_2 are aggregated in a list and passed as $<k_2,\{v_2\}>$ to the reducer function. The reducer function will reduce all the intermediate values $\{v_2\}$ for a given key k_2 and output a new key-value pair $<k_3,v_3>$ (step 3.b). Finally the new key-value pairs are collected, formatted and written to the output file (step 3.c).

In the simple word-count example, each word k_2 arrives at the reducer function with a list of numbers $\{v_2\}$ (all set to the value 1). So the length of this list represents the number of occurrences of a word and the word-count reducer outputs the new pair $<k_2, \text{length}(\{v_2\})>$.

3.1.4 Apache Hadoop

Apache Hadoop is an open-source Java implementation of the MapReduce programming model and a dedicated DFS, namely the Hadoop DFS (HDFS). Hadoop offers a software framework used to create data-intensive, distributed applications. It enables applications to work with thousands of computers (nodes), collectively referred to as a cluster, and petabytes of data. The design of Hadoop was inspired by Google's MapReduce [RD 4] and Google File System papers [RD 5].

The real benefit of Hadoop lies in the combination of the HDFS and the MapReduce programming model. As far as possible, Hadoop will execute mapper tasks on cluster nodes which store the input data. If this is not possible (e.g. maximum number of concurrent mappers on a node reached), the execution will take place on another node and data will be transferred to that node over the network.

Tasks that fail or not respond within a given time period are executed again on another node before the whole job fails. Optionally, tasks are speculatively executed a second time on idle nodes. The first returning node contributes the result. This should prevent slower nodes from slowing down the whole cluster.

3.2 Calvalus Approach for Concurrent Processing

Hadoop MapReduce has been designed for efficient highly-parallel processing. But is this immediately applicable to EO data processing and does it help to solve problems in this domain?

One of the salient challenges in EO is the large amount of data. Due to this, the bottlenecks are processing power and network transfer rates. In architectures with a central archive, the processing involves transfer of all data from the archive. The data is not local to the processing algorithm, known as the data locality problem.

The theses of this study are:

- Bulk processing of large EO file sets on a Hadoop cluster is efficient and reliable.
- L1-to-L2 processing can be parallelised by processing each L1 input independently
- L2-to-L3 processing can be parallelised by inputs and by geographic partitioning with the MapReduce approach

The selected sub-domain for this study is instrument cal/val and the development and validation cycle of L2 algorithms and data products. It is a computationally challenging to minimise the validation time for large product file sets, e.g. one year of satellite data. For computationally expensive algorithms this can only be achieved through parallelisation.

The L2 processing itself is directly parallelisable because each processing task can be performed independent of each other. The approach for this class of input-to-output processing is to use simple mapper-only workflows. The MapReduce model is very well suited for workflows that include L3

processing, because it includes geographic sorting and spatio-temporal aggregations. The four production types used for Calvalus demonstration and their process implementations using Hadoop are described in more detail in chapter 4, Production Types and their Realisations.

3.3 Supported Processor Interfaces

Calvalus supports two types of data processors that take a single input product and generate a single output product. They are a BEAM GPF Operator Interface and a simple shell interface, which are described in more detail below. Calvalus processors, their parameters and LUTs can be easily deployed across the system and many versions can run in parallel.

3.3.1 BEAM GPF Operator Interface

The BEAM GPF Operator interface is part of the official BEAM development platform. It allows developers to easily implement new data processors using a very effective programming model. Processors developed against this interface are ported to MapReduce using the Calvalus BEAM Adapter. One of the most important concepts of the BEAM development platform is an internal representation of remote sensing products away from their external file format. Product readers create instances of a product data model. Once the data model is instantiated in memory, various higher-level BEAM APIs and frameworks can use it for data processing, analysis and visualisation.

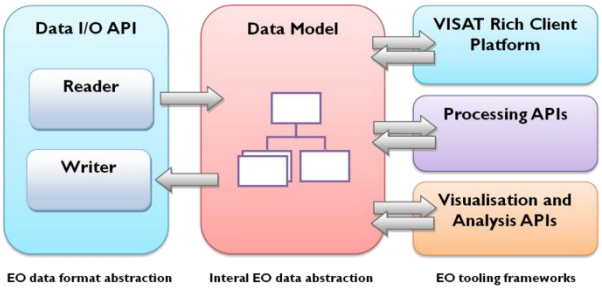

Figure 7: BEAM core concept

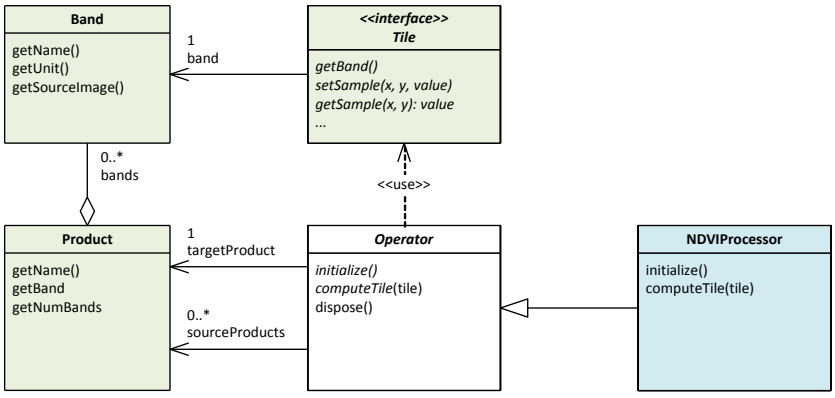

Figure 8: BEAM graph processing framework

One of those higher-level APIs is the BEAM Graph Processing Framework (GPF). It allows for developing new operators that can later serve as nodes in processing graphs. A development of a new operator is actually an implementation of the Operator interface for the BEAM GPF. The interface comprises basically three operations that are called by the framework. The first operation initialises the operator and defines a target product including all target bands (variables) to be computed (operation *initialise*). The second operation is used to compute all the pixels for a given tile. The tile represents a part of a target band's raster data (operation *computeTile*). The last operation is called by the framework in order to allow an operator to release its allocated resources (operation *dispose*), e.g. file pointers to auxiliary data.

3.3.2 Shell Interface

The shell interface allows incorporating the executables that can be invoked from a command line shell and that do not have any user interactions beyond setting up the command line processing parameters. The interface comprises a process descriptor file. This is a plain text file (XML) that describes the inputs files (name and type), the processing parameters (name, type, value range), the output file (name and type) and provides a template for the command-line that is used to invoke the executable.

The Unix executables that have been used so far with the shell interface are **l2gen** (the OC Level-2 processor from the NASA's SeaDAS software package), **AMORGOS** (MERIS geo-correction tool developed by ACRI) and **childgen** (a MERIS/AATSR subsetting tool developed by BC). It is planned to integrate **MEGS** (prototype processor for the standard MERIS Level-2 product), in the near future.

4 Production Types and their Realisations

This chapter describes operational scenarios in terms of production types that have been implemented in Calvalus. Four production types that realize a typical calibration, algorithm development and validation cycle are in the focus. In addition, system use cases from the user's point of view are defined.

4.1 Production Types Overview

The Calvalus processing system realises four important scenarios triggered by EO and Cal/Val users. They are:

1. **L1 to L2 Bulk-Processing** from L1b top-of-the-atmosphere radiances to geophysical products of water-leaving reflectances, IOPs, and chlorophyll,
2. **L1/L2 to L3 Bulk-Processing** from L1/L2 data to their spatio-temporally gridded products,
3. **Match-up Analysis** on water-leaving reflectances, IOPs, and chlorophyll,
4. **Trend Analysis** on water-leaving reflectances and chlorophyll.

As described in more detail in the following, the matchup analysis compares static reference measurements with L2 data that are processed from L1b. The trend analysis generates time-series from spatially and temporally aggregated L2 data which are processed or read from L1b or L2 data. The matchup and trend analyses produce comprehensive reports including diagrams and data tables.

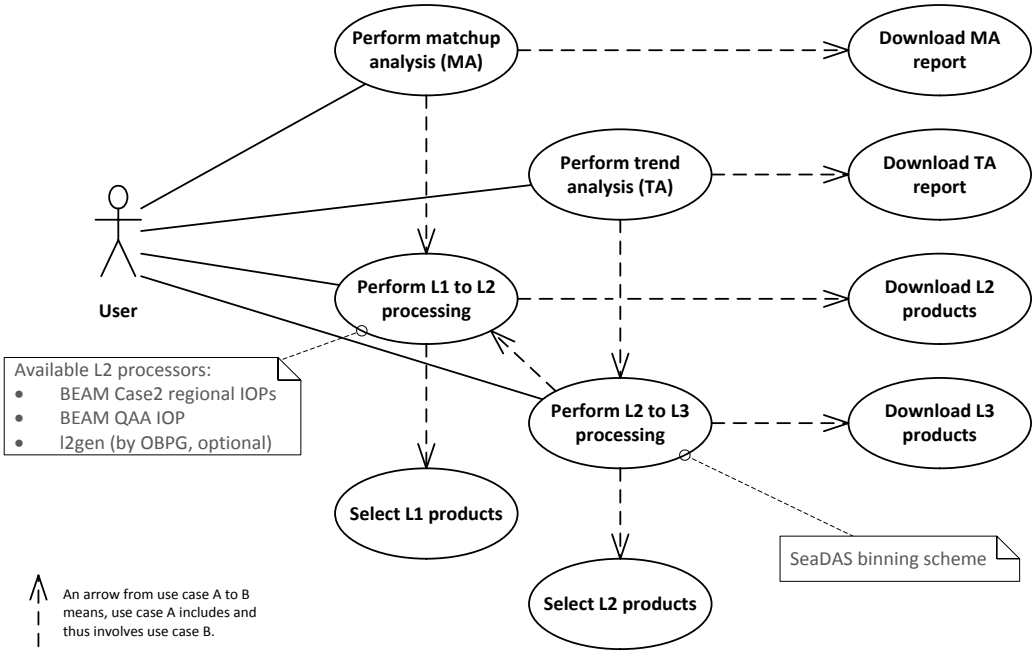

Figure 9: Top-level use cases

As shown in the use case diagram in Figure 9, the efficient generation of L2 and L3 data are important scenarios on their own. Users can select a number (or all) L1b data products and bulk-process them to L2 and L3 and download the generated data.

The major aim and the most challenging task of the Calvalus study is to implement an efficient processing system that utilises and exploits the power of Hadoop in order to realise these four

production types. The trade-off analyses and technology studies that led to the various realisations using Hadoop are described in detail in the Calvalus Technical Specification [RD 23].

4.2 Level-2 Bulk Processing

4.2.1 Production Type Description

The production type Level-2 Processing (L2) allows user to process a (filtered) input file set using a selected processor to an output product set. If a spatial (region) filter is applied, the input scenes are first extracted to match the given region geometry, thus the output product files may also be subsets. The result of the production is a zipped set of output files in a user selected EO data format (currently BEAM-DIMAP, NetCDF, GeoTIFF), that can be downloaded by the user.

For the demonstration of the Calvalus system, the CoastColour L2W Level-2 processor is used. It includes the Case2R [RD 5] atmospheric correction and Case2R or QAA [RD 6] chlorophyll and IOP retrieval algorithms.

The following table lists geophysical variables of the output product of the CoastColour L2W processor:

Name	Description
iop_a_pig_443	Absorption coefficient at 443 nm of phytoplankton pigments
iop_a_ys_443	Absorption coefficient at 443 nm of yellow substance
iop_bb_spm_443	Backscattering of suspended particulate matter at 443 nm
iop_a_total_443	Total absorption coefficient of all water constituents at 443 nm
K_min	Minimum down-welling irradiance attenuation coefficient
Kd_λ	Downwelling irradiance attenuation coefficient at λ, where λ is one of 412, 443, 490, 510, 560, 620, 664 and 680
turbidity	Turbidity in FNU (Formazine Nephelometric Unit)
Z90_max	Inverted value of k_min
conc_chl	Chlorophyll concentration (mg m^{-3}).
conc_tsm	Total suspended matter dry weight (g m^{-3}). $tsm\_conc = tsmConversionFactor \cdot b\_tsm^{tsmConversionExponent}$
chiSquare	A low value in the product indicates a higher success in the retrieval and that the conditions, which have led to the measured spectrum, are in (sufficient) agreement with the conditions and the bio-optical model used in the simulations for training the neural network. A value above a threshold of spectrumOutOfScopeThreshold (default is 4.0) triggers the out of training range == out of scope flag.
l1_flags	Quality flags dataset from L1b product
l1p_flags	CoastColour L1P pixel classification
l2r_flags	CoastColour L2R atmospheric correction quality flags
l2w_flags	CoastColour L2W water constituents and IOPs retrieval quality flags

Table 2: Output of the L2W Level-2 processor

Calvalus is capable to host any number of processors. However, the Calvalus portal currently offers to users only BEAM-installed processors.

4.2.2 Realisation using Hadoop

The processing of a set of L1 input products into a set of corresponding L2 output products belongs to the class of problems that can be directly parallelized across the input data. For each file in the (possibly filtered) input product file set, the Calvalus system creates a mapper task on a dedicated node in the Hadoop cluster. The Hadoop processing engine tries to select the node according to the location of the data in the cluster so that the tasks most probably work data-local. No reducer tasks are required for Level-2 processing. In its current configuration (20 nodes cluster), and in the ideal case (no other tasks running), the Calvalus system can perform a L2-processing of 20 files 20 times faster than in sequence on a single computer.

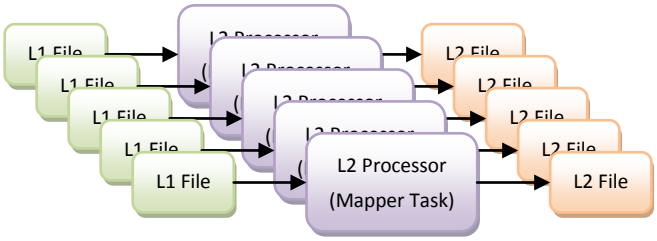

Figure 10: Level-2 Processing using Hadoop

An analysis has shown that when processing a whole set of products from L1 to L2 the best approach is to process a single product by a *single* mapper. In order force Hadoop to process data-local, the block size of input files has been set to the file size. Thus, the splitting function is redundant because HDFS blocks represent complete input files and each single mapper processes the one and only split per input file. This leads to the desired behaviour to execute the mapper task, whenever possible, on a cluster node that stores a complete replica of the input file.

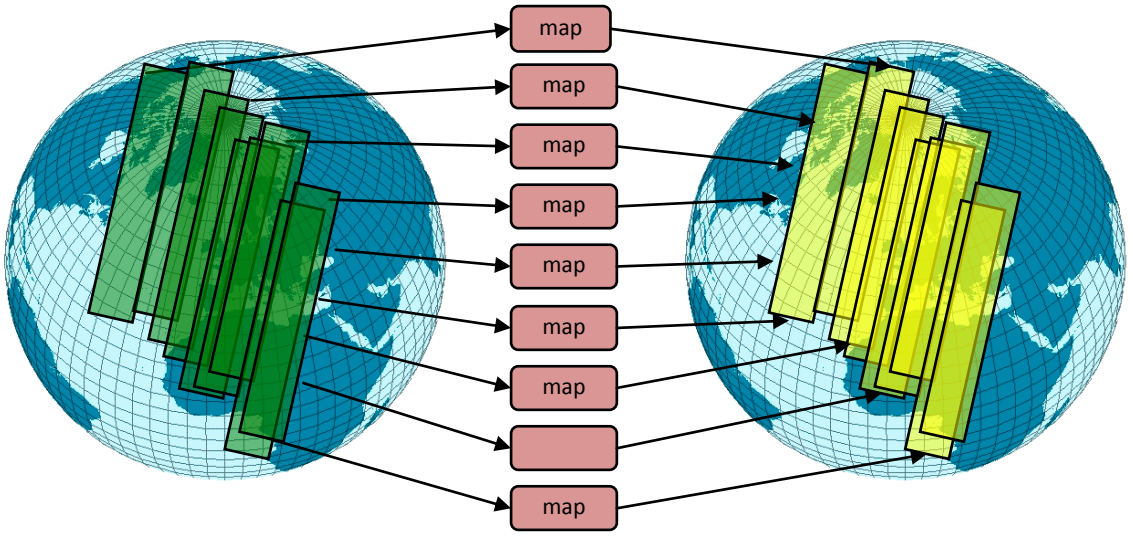

Figure 11: Concurrent mappers for L2 processing of multiple files

When only one product has to be processed, this approach results in a single mapper processing the input file on a single node. So there is no advantage of using the cluster. In this case, multiple splits could be created to foster parallel processing on multiple nodes. This would lead to many nodes processing splits of the input product, but the number of splits that are processed data local, will depend on the replication rate of the block that represents the input file. So this approach is only useful when the computation time outweighs the time for the data transfer. A study within Calvalus has shown that for a computationally expensive algorithm, like the CoastColour L2W, using multiple splits per input file is an advantage. However, Calvalus has been designed to operate on sets of input

files and not on single input files, so the latter approach has not been considered in the implementation.

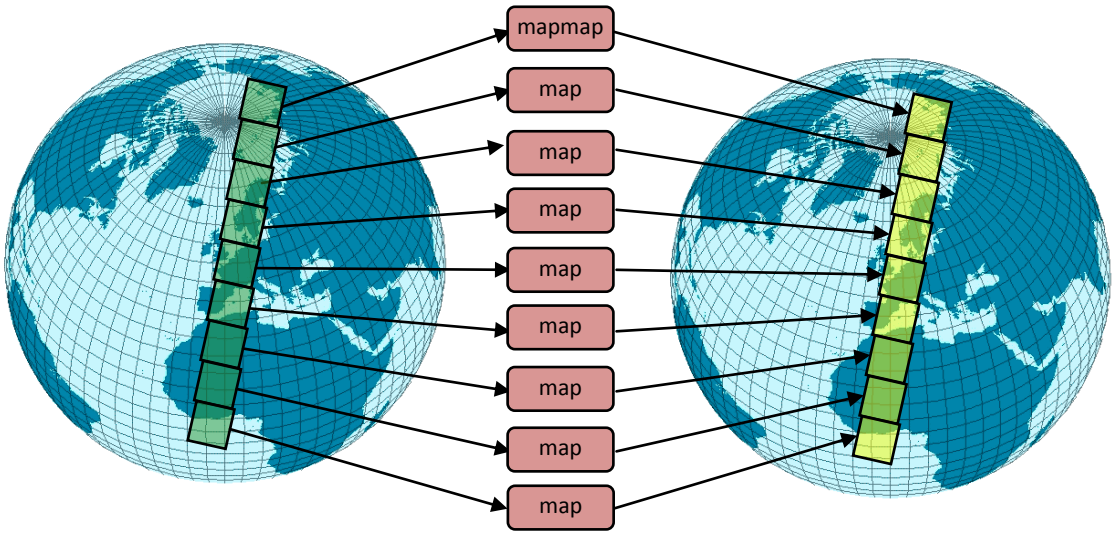

Figure 12: Concurrent mappers for L2 processing of a single file

4.3 Level-3 Bulk Processing (L3)

4.3.1 Production Type Description

The production type Level-3 Processing (L3) allows user to process a (filtered) input data product file set using a selected Level-2 processor to one or more Level-3 data products. The result of the production is a zipped set of output files in a user selected EO data format (currently BEAM-DIMAP, NetCDF, GeoTIFF), that can be downloaded by the user.

For the demonstration of the Calvalus system, the Level-2 processor for L3 testing is again the CoastColour L2W processor (same as for L2).

The L3 production type can generate many L3 output variables at the same time. Users simply add a new variable using the Add button below the table of variables. The list of available variables is specific to the selected L2 processor. In the case of the CoastColour L2W processor, all variables listed in Table 2: Output of the L2W Level-2 processor" may be added.

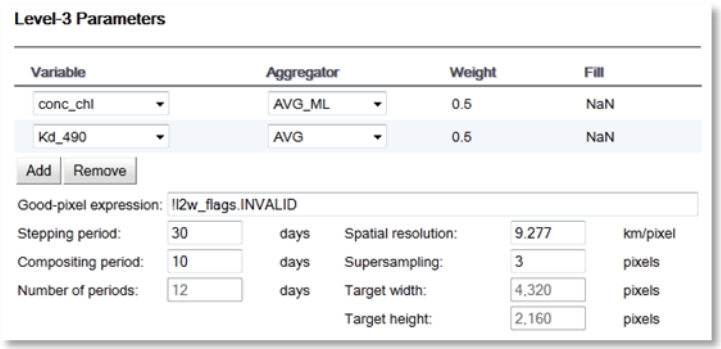

Figure 13: Level-3 parameters

The pixels used for the L3 products must pass a test given by the *good-pixel expression*. This expression is a BEAM band maths expression that may contain all the bands and flags present in the L2 output products. The expression is used to screen L2 pixels before passing them to L3 binning.

The time range used for generating the L3 output products is given by the *Temporal Filter* selection. The frequency L3 output files are selected within the time-series is determined by the parameter

stepping period, e.g. every 30 days. The resulting number of L3 products in the time-series is the number of days of the total time range divided by the number of days given by the *stepping period*. The actual number of input product days that are used to produce each L3 output file is given by the parameter *compositing period*, which must be equal to or less than the *stepping period*, e.g. 4-days, 8-days, monthlies.

The default *spatial resolution* is 9.28 km per output pixel resulting in a grid resolution of 4319 x 2190 pixels for global coverage L3 products. Finally, the *supersampling* parameter can be used to reduce or avoid the Moiré-effect, which occurs in output images if the binning grid is only sparsely filled by input pixels. This situation usually occurs when the spatial resolution used for the binning is similar or smaller to the input pixel resolution. The *supersampling* subdivides every input (Level-2) pixel to n x n subpixels which all have the same values but different and unique geographical coordinates. This way, an input pixel may be distributed to more than one adjacent bin cell.

The binning algorithm implemented in Calvalus is the same that is used by the NASA OBPG for creating the SeaWiFS and MODIS ocean colour Level-3 products [RD-11].

4.3.2 Realisation in Hadoop

As for L2, the L3 production scenario creates a *mapper* task for each file in the (possibly filtered) input product file set on a dedicated node in the Hadoop cluster. The mapper task reads in the input product, processes it to Level-2 data, and, according to the binning parameters, performs a spatial binning of the data. The output of the mapper are spatially aggregated bin cells. A number of reducer tasks are then responsible for the temporal binning at the individual bin latitude ranges. They get the spatially binned outputs from all the mappers, perform the temporal binning and output bin cells for each bin latitude range. A special formatter task is used during the staging process to collect all the

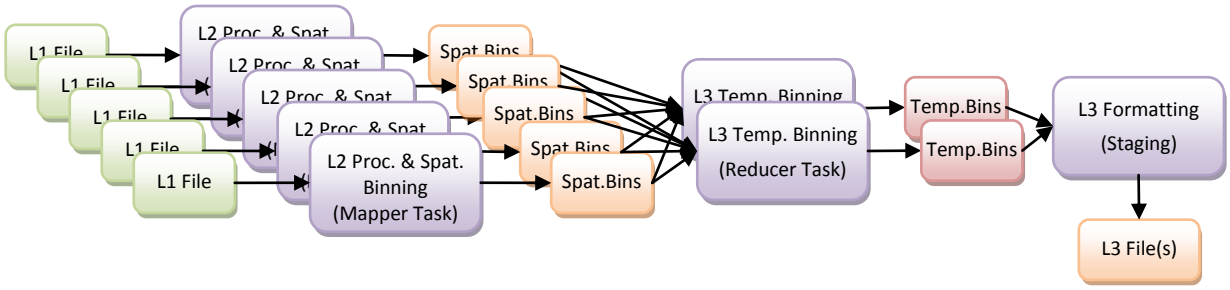

Figure 14: L3 production type in Hadoop

latitude bin ranges parts and compile the final binned data product.

The Calvalus implementation of the OBPG binning algorithm is very efficient. The binning scheme is a perfect use case for the Hadoop map reduce programming model. Data locality is in most cases fully exploited. Level-2 processing is performed on-the-fly and no intermediate files are written. The following Figure 15 provides another view on how concurrency is utilised to generate a single L3 product.

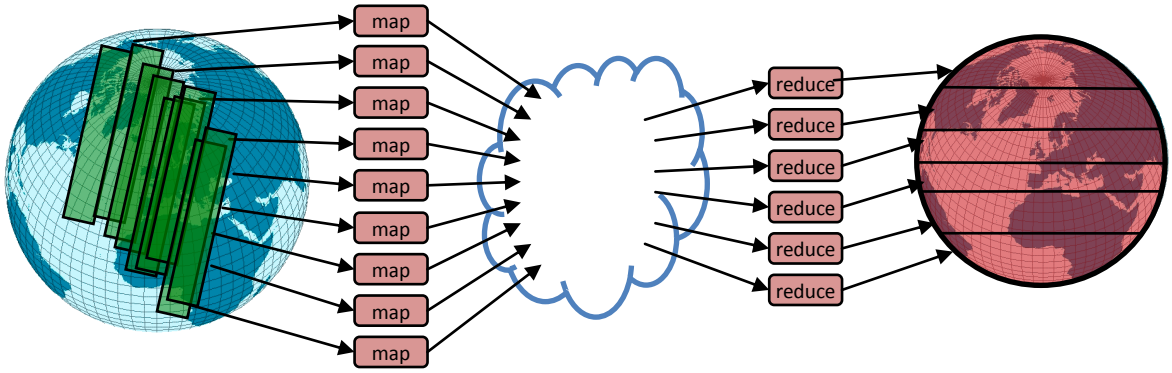

Figure 15: Concurrent mappers for inputs and concurrent reducers for regions

The approach is characterised by

- a *mapper* for each L2 input performing spatial binning, this generates intermediate data for bin cells with weight information, using the bin cell ID as a key
- *partitioning* into ranges of bin cells, the ranges cover the region of interest
- a *reducer* for each partition doing temporal binning for every bin cell in the range, it writes a segment of the output
- a *formatting* step to create the user output from the output segments distributed in the HDFS

For large datasets the approach is expected to perform better than the processing on a single node:

- Processing on the cluster is expected to be faster by a factor limited by the number of concurrent mappers and the number of concurrent reducers. It is expected that the number of mappers is more important since the mappers have to read more input (depends on the bin cell size, see below).
- Some overhead for distributed tasking can be expected, which relates to the overhead produced by task invocation and for data transfer between mappers and reducers. As binning is not a computationally intensive process the overhead cannot be neglected.
- The process uses data-locality, as the inputs are distributed and mappers are largely scheduled on data-local nodes.
- The amount of data transferred to reducers is by a factor smaller than the overall input size that corresponds to the ratio between pixel size times the compositing period and the bin cell size.

4.4 Match-up Analysis (MA)

4.4.1 Production Type Description

The match-up analysis (MA) is a production scenario that compares reference point measurements, such as in situ observations, with corresponding extracts from Level-1 or Level-2 data. The measurement points are taken from a user-provided data table. A record in this table may not only contain the geographical coordinate of a point but also any number of reference data (in-situ data, or EO data) and measurement metadata such as the measurement date and time. For any variables contained in the data table which are also found in the Level-1 or Level-2 data products, the MA generates scatter plots and provides a linear regression of how the reference data matches the data

found in the data products at given points. The following screenshot in Figure 16 shows the MA parameters in the Calvalus portal.

Match-ups are not done on single pixels but on macro pixels that include neighbours to the centre pixel that exactly corresponds to the given geographical point coordinate. In the Calvalus implementation of the MA, the macro pixels span 5 x 5 “normal” pixels or more.

All pixels in the macro pixel are screened and a list is generated of values that are compared against the reference measurement data.

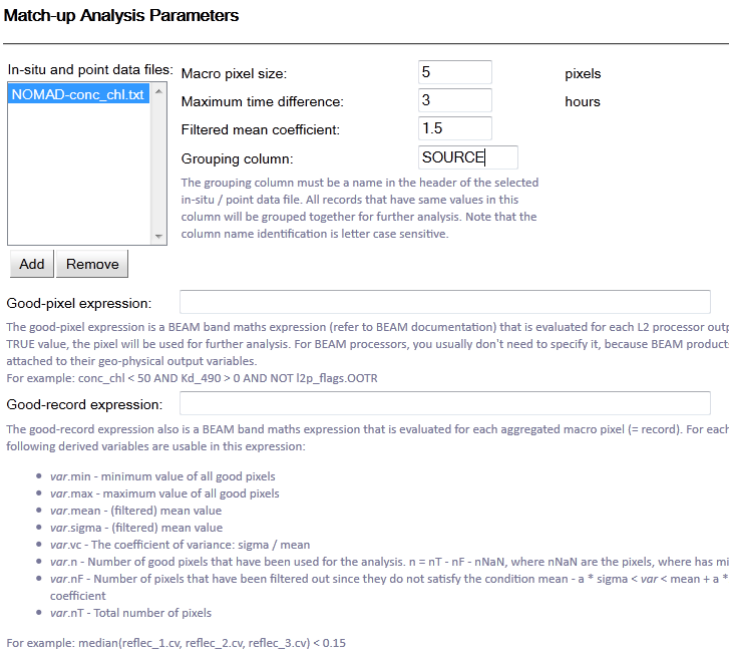

Figure 16: Match-up analysis parameters

4.4.2 Realisation in Hadoop

Similar to the L2 production type, the MA production scenario starts with L1 data and creates a *mapper* task for each file in the (possibly filtered) input product file set on a dedicated node in the Hadoop cluster. The mapper task reads in the in-situ / point data records and creates output records using the L1 input file processed to L2. Again, the Hadoop processing engine tries to select the node according to the location of the data in the cluster so that the tasks most probably work data-local. All output records are passed to a single Reducer task, which aggregates the records, computes statistics and generates the plots. The Calvalus system can process a 1000 Level-1 input files in a few minutes, because the L2 processing is only performed on sub-regions given by the point records and the macro pixel's size.

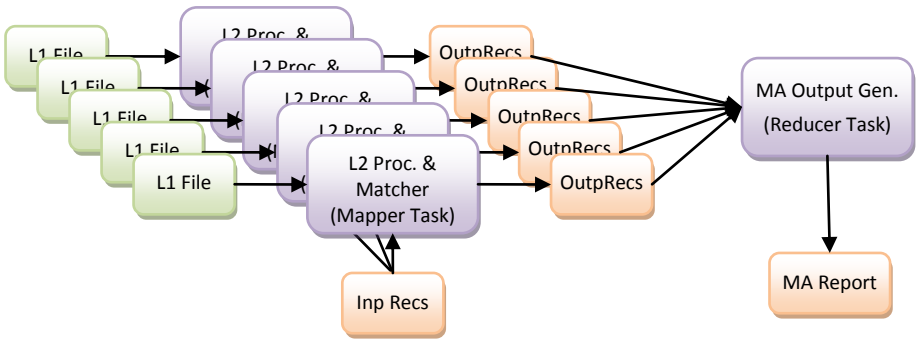

Figure 17: MA production type in Hadoop

4.5 Trend Analysis (TA)

4.5.1 Production Type Description

The TA production type is used to create time-series of Level-3 data. It has therefore the same parameters as the ones described in the chapter on Level-3 bulk processing. However, the time range

for a meaningful analysis is typically many months; and the compositing period is usually significantly smaller than the stepping period. For example, the TA automatically performed by the OBPG for the SeaWiFS and MODIS ocean colour products uses a stepping period of 32 days and a compositing period of 4 days. The *spatial resolution* is fixed to 9.28 km, the *supersampling* fixed to 1.

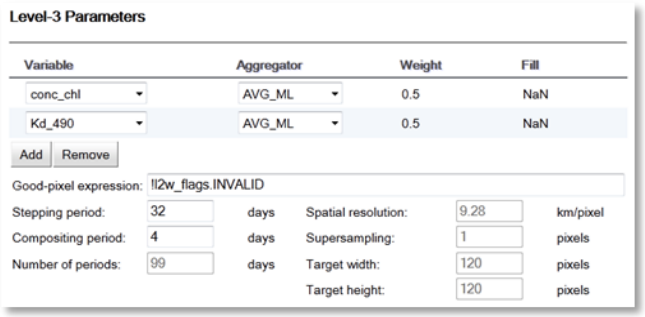

Figure 18: TA parameters

Opposite to L3 production type, the temporal bin cells for the compositing period are all aggregated and averaged. So every compositing period results in a single value for each variable forming the time series over the entire time range of the analysis.

4.5.2 Realisation in Hadoop

The TA production type is implemented in the same way as the L3 production type with the exception that the temporal bin cell outputs by the *reducer* tasks are all averaged again. So every compositing period results in a single value for each variable forming a time series over the entire time range of the analysis.

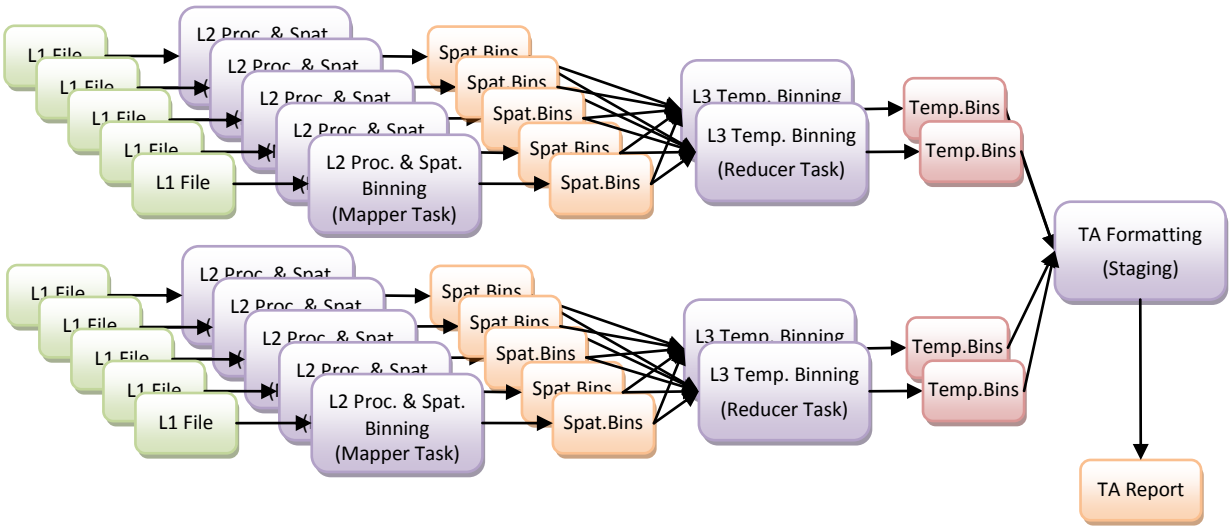

Figure 19: TA production type in Hadoop

5 System Architecture

5.1 Prototype System Context

The prototype system has been developed to demonstrate parallel processing for the four production types and its use via a portal. The following Figure 20 shows Calvalus with user portal and web services as the front-end, and the Hadoop cluster for distributed processing and data storage as the back-end.

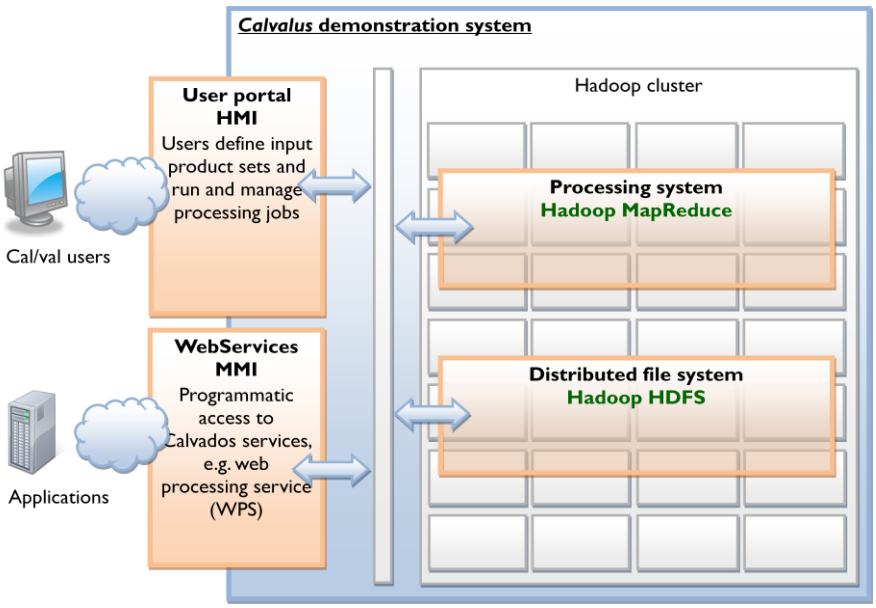

Figure 20: Hadoop cluster services and user interfaces in Calvalus

To demonstrate parallel processing and its usability the Calvalus prototype has implemented

- a portal as a user interface
- a set of services for data and processing management
- L2 and L3 code as Hadoop-driven parallelized processors
- aggregation and analysis functions

In favour of focusing on the parallel processing, other functions are simplified (catalogue query, metadata schemata, online data access, web service interfaces), re-used from existing components, or implemented by simple shell scripts to be used by the operator (data ingestion). So, readers should not expect the Calvalus implementation to cover all functions described in this design. Figure 21 shows the Calvalus system in its context between the user, the EO data processor and the operator.

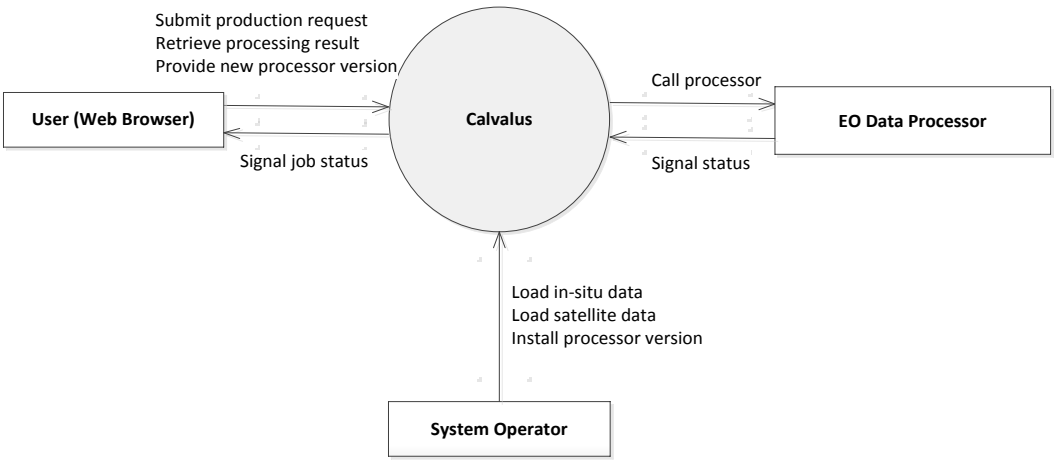

Figure 21: Calvalus system context

5.2 System Decomposition

The Calvalus demonstration system is composed of an EO data processing system based on Hadoop, a number of dedicated service components, and a user portal. The UML component diagram shown in Figure 22 identifies the system's components and interface dependencies between them.

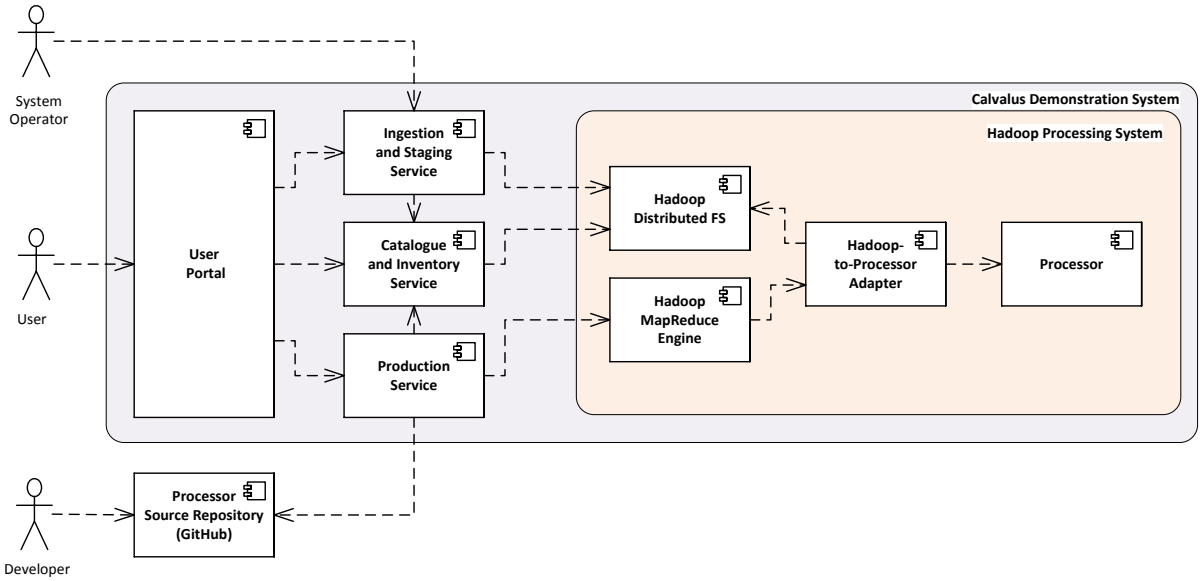

Figure 22: Calvalus system decomposition

Registered users interact with the processing system solely via the user portal. The portal is a usual web application implemented with Google Web Tool Kit, GWT. It is accessible from typical Internet browsers. It communicates with the processing system exclusively via the Calvalus services. Beyond the user portal, developers submit updates to existing processors and new processors that are stored and maintained within the system. Operators monitor operations of the entire Calvalus system.

The processing system comprises the MapReduce engine, the HDFS and the actual processors to be executed by the processing system. The MapReduce engine and the HDFS are both part of the Hadoop software. HDFS also serves as the one and only Calvalus data archive. The processors realise the various Calvalus processing scenarios and are actually independent of the MapReduce API of Hadoop. A dedicated Hadoop-to-Processor adapter is used to invoke the processors in a MapReduce-specific style.

The interfaces of the service components shown above abstract away from the Hadoop-specific concepts, such as those introduced by the Hadoop MapReduce engine and the HDFS. The service components are more common to the EO domain, e.g. data product files, product file ingestion, catalogue, inventory, production and staging.

5.2.1 User Portal

The Calvalus user portal is the main human-machine interface to the Calvalus system. For registered users, it provides an intuitive access to the internal Calvalus services such as data catalogue query, data inventory manipulation, production control and staging of output data. The entry page of the portal is public and provides to visitors a detailed description of the Calvalus study.

The computational service provided by the user portal involves the following:

- user authentication, user management
- selection of available input file sets, as well as spatial and temporal file set filters
- configuration and submission of production requests
- management of submitted productions, progress observation, cancellation
- download of results

As a web application, the portal is accessed through its URL from a typical Internet browser. The entry page of the portal is public and provides to visitors a detailed description of the Calvalus study. In order to use the Calvalus data services, users must be registered and signed-in. After signing in, the users see product sets they own and the status of jobs they are currently running. The users can choose from a menu to perform a data catalogue query, to manage product sets and production requests.

The Calvalus portal is described in more detail in Chapter 7.

5.2.2 Catalogue and Inventory Service

The catalogue and inventory service is the place for metadata and collection information in Calvalus holdings. It hosts metadata of EO products, of reference data and of auxiliary data, it serves queries, and it maintains predefined collections and user-defined product sets. Besides temporal and spatial coverage the metadata comprise product file information available from the respective product file types.

The computational service provided by the catalogue and inventory service is:

- product file identification, each product file gets a unique identifier in Calvalus
- catalogue search based on metadata, including temporal and geo-spatial criteria, and on predefined collections or user-defined product sets
- presentation of results with detailed metadata records
- inventory lookup to locate products in the archive, translate from identifier to physical archive location

5.2.3 Production Service

The production service manages and controls production processes for the generation of new products within Calvalus. It handles production requests from users, maintains production recipes, organises processing chains, and ensures cataloguing and archiving of results.

The computational service provided by the production service is:

- Production request handling, generation of production jobs, maintenance and display of their states, command handling (cancellation)

- Production job execution by translation into one or more processing steps, driven by production recipes
- Issue of processing requests to execute in steps in the Hadoop MapReduce engine and to be monitored
- Interaction with catalogue and inventory service to resolve product sets, to get product file locations, to create result product set, to catalogue and archive results
- Production failure handling
- Maintenance of production request templates (get, add, remove, update) to be used for request composing by the users in the portal
- (optional) Automated retrieval of requested processor versions from repository and deployment on the Hadoop cluster
- Maintenance of processor updates and processor versions

5.2.4 Ingestion and Staging Service

The ingestion and staging services are the data gateways of Calvalus. They implement both ingestion of new EO products and reference data into the system, and access to produced and archived data by staging into a user-accessible download area.

The computational service of the ingestion and staging provides:

- Extraction of metadata
- Validation of inputs
- Thumbnail generation
- Archiving rules application to determine archive location
- Consistent archiving, inventorying and cataloguing

The computational service provided for staging is:

- Data retrieval from archive
- Formatting of output products files from distributed concurrently generated partial results
- Data analyses, plot generation, statistics generation, provided by plug-ins (see also section 5.2.8 Processor)
- Data provision in staging area (in order to isolate the cluster from direct user access)
- Notification of data provision
- Deletion of data from staging area after successful retrieval

The formatting function, in particular, converts temporary partial outputs into user formats like NetCDF, GeoTIFF, BEAM-DIMAP.

5.2.5 Hadoop Distributed File System (HDFS)

The Hadoop distributed file system serves as an archive for primary and auxiliary input and output data products. On the data provider and user side, the data in the archive is encapsulated by the ingestion and staging service. On the processor side, it is accessed locally or remotely, in a controlled way, via the Hadoop-to-processor adapter.

The computational service of the HDFS is:

- File system functions to store files, to organise them in directories (create, read and delete files; create, list and delete directories)
- Data replication to different nodes to improve fail safety and to support data locality
- Distributed data access to support data locality

The functions are accessible by the Hadoop namenode and a client API.

5.2.6 Hadoop MapReduce Engine

The Hadoop MapReduce engine is the cluster scheduler and the workflow engine for the map-reduce programming model. It distributes tasks to the cluster of computing nodes in a way that obeys data-locality.

The computational service of the Hadoop MapReduce engine is based on the following:

- Parallelisation, creation of concurrent tasks for a Hadoop job with a set of inputs
- Distributed processing, scheduling of tasks on the cluster of processing nodes
- Data-locality, considering data-locality for scheduling
- Orchestration of map and reduce tasks, partitioning and sorting (re-shuffle) of intermediates
- Monitoring of task execution, status handling
- Failure handling with automated retry (failover)
- Speculative execution (preventive failover)

5.2.7 Hadoop-to-Processor Adapter

The adapter integrates existing processors into Calvalus and binds them to the Hadoop MapReduce engine. The adapter is foreseen in two variants, for BEAM GPF processors and for executable/shell script processors. The adapter further serves as an example or pattern for the implementation of specific Calvalus processors that directly interface to the Hadoop MapReduce engine.

The UML diagram in Figure 23 shows the two variants of the Hadoop-to-processor adapter in order to bind different types of processors to the MapReduce engine: one for BEAM GPF operator processors and the other for executable shell script processors.

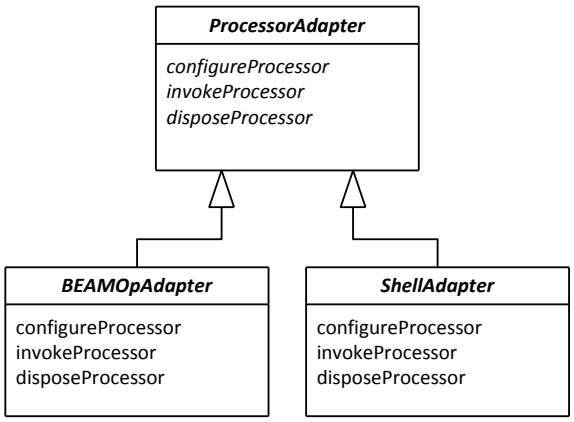

Figure 23: BEAM and shell script variants of the Hadoop-to-processor adapter

The computational service of the Hadoop-to-processor adapter is concerned with:

- Transformation of the Hadoop MapReduce implementation interface to interfaces of existing processors
- Invocation, control and monitoring of the processors
- Parameter provision as method parameters or parameter files
- Input data product provision as input streams or local copies of files
- Output data product archiving provided as output streams or local files
- Preparation of the environment before processing and cleanup of local files after processing
- Status propagation

5.2.8 Processor

The processor implements the algorithm to transform input data into output data. It is executed on the cluster to perform a processing step. There are different types of processors for different algorithms. Processors are versioned.

In the demonstration system, different versions of the CoastColour L2W Level-2 Processors are used. The latest version (1.3) uses the Case2R atmospheric correction combined with Case2R [RD 5] and QAA [RD 6] IOP and chlorophyll retrieval algorithms. Another processor is l2gen [RD 14] that is currently becoming a selectable processor in Calvalus.

The computational service of the processor is:

- Transformation of inputs to one or more outputs in a processing step considered as atomic. Outputs may be EO products of a higher level or reports.
- Data analyses, plot generation, statistics generation
- Status provision

6 Calvalus Cluster Hardware

Calvalus is considered to be an independent, self-contained demonstration system. It relies heavily on the Hadoop technology, which in turn is supposed to be operated on a Linux cluster.

The hardware for Calvalus has been procured by Brockmann Consult and is hosted at Brockmann Consult's premises. It will be operated and maintained for 2 years after the official study end.

The hardware system simulates a larger system supporting multiple users with multiple, simultaneously running jobs. However, it comprises enough nodes to adequately test the scalability and reliability of the system. With this requirement the following selection criteria for the cluster hardware have been established:

- Prefer a higher number of servers over the performance of each server
- Prefer a high computational performance of single servers over their fail-safety

The current cluster has been built from rack-mounted barebones. Compared to standard desktop computers they have the advantage of being made of components of server quality that are designed to run 24/7. Furthermore space occupied by 20 desktop cases is significantly larger than a single server rack with the possibility of hosting up to 42 servers.

The Calvalus project has acquired Supermicro servers. The servers are very similar to the Intel barebones but feature a 4th drive bay for future expansion of the storage space as well as KVM over LAN for remote hardware maintenance. In contrast to the barebone servers from Intel, they are delivered fully assembled.

The full specification of the procured servers:

- Supermicro Barebone 5016I-MTF
- 1HU Rackmount with 280W power supply
- 4x HotSwap SATA trays
- 1x Intel Xeon X3450 Processor with 2,66 GHz Quad Core, 8 MB Cache
- 6 Memory Slots (max. 32GB) - 8 GB Memory (4x 2 GB DDR3 reg. ECC)
- 2x Gigabit Ethernet controller onboard with RJ-45 LAN connector.
- IPMI 2.0 incl. KVM over LAN Expansion Slots: 1x PCI-Express x16
- 3x 1,5 TB S-ATA Seagate Disks, 7,2K UPM, 32 MB Cache ST31500341AS (one disk tray remains empty)

All 20 servers are connected using a Gigabit Ethernet switch. They are installed in a rack as shown in Figure .

Figure 23: Calvalus cluster hardware

The operating system on the servers is “Ubuntu Server 10.04 LTS (Long Term Support), 64bit”. We currently have a configuration with one server being the dedicated master (*namenode* and *jobtracker* in Hadoop terminology) for the cluster and 19 servers operating as slaves (*datanode* in Hadoop).

7 Calvalus Portal

The Calvalus portal is the main user interface to the Calvalus system. It is a simple, JavaScript-based web application that lets users submit production requests and download the produced results. The name *portal* is justified by the fact that it provides users a portal to the actual processing system, the Hadoop cluster comprising 20 Linux machines (quad core, 8 GB) and 112 TB of data storage.

The Calvalus system currently hosts MERIS RR Level 1b data from 2002 to 2010. With this data set, users can submit production requests according to the production types described in the chapters above:

1. L1 to L2 bulk processing
2. L1/L2 to L3 bulk processing
3. L2 match-up analysis or point data extraction
4. L3 trend analysis

The following screenshot in Figure 24 shows the portal after signing in:

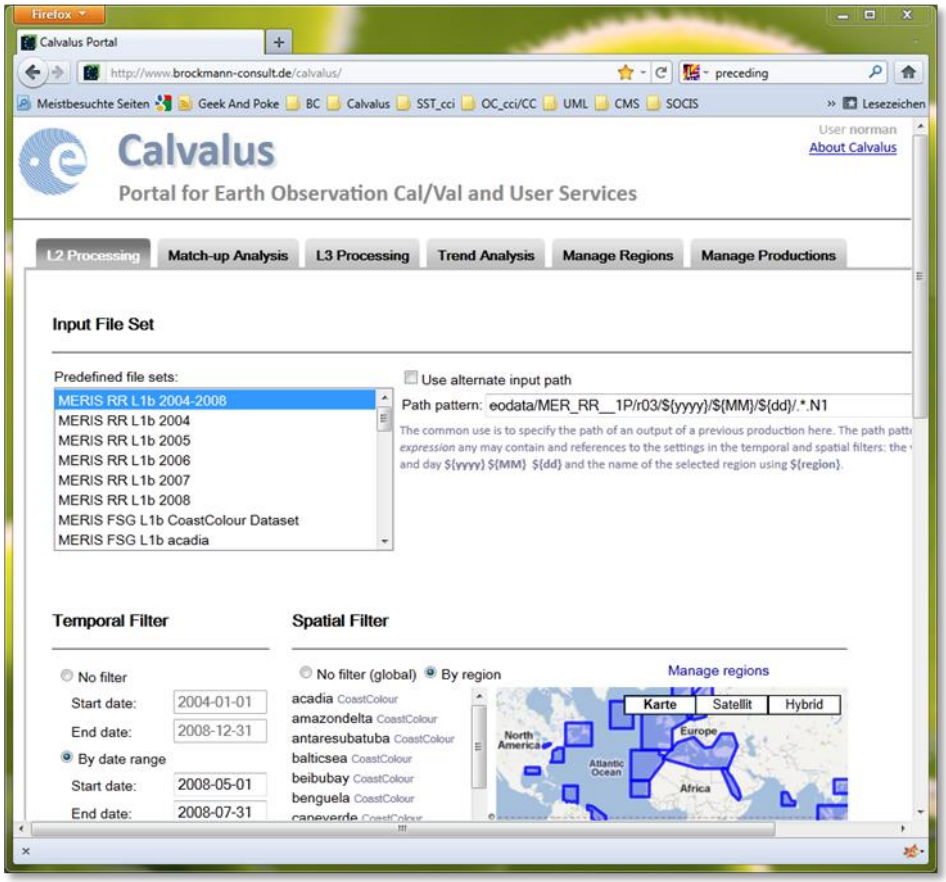

Figure 24: Calvalus portal

The input dataset is organised into product file sets which can be temporally and spatially filtered for all types of productions before they are passed as input to the processing system. Temporal filters are realised as date range or date list while the spatial filter is based on geographical regions. A dedicated region manager is used to manage predefined and user defined regions.

After a production request has been submitted, users can observe, cancel, stage productions and finally download the results.

At the time of this writing, the web application is compatible with most of the Internet browsers. It has been developed using the Google Web Toolkit 2.3 and has been tested with Firefox 6, Chrome 13 and Internet Explorer 9.

7.1 Input File Set

An input file set, as displayed in Figure 25, comprises a list of EO data product files that are identified by a file path that may contain *regular expressions* (wildcards). A production scenario can only have a single file set as input. The current file sets comprise MERIS RR for the years 2002 to 2010, and regional subsets for the South Pacific Gyre (SPG) and North Atlantic (NA) as used by the ESA Ocean Colour CCI project. Each file set also "knows" the date range of its contained data.

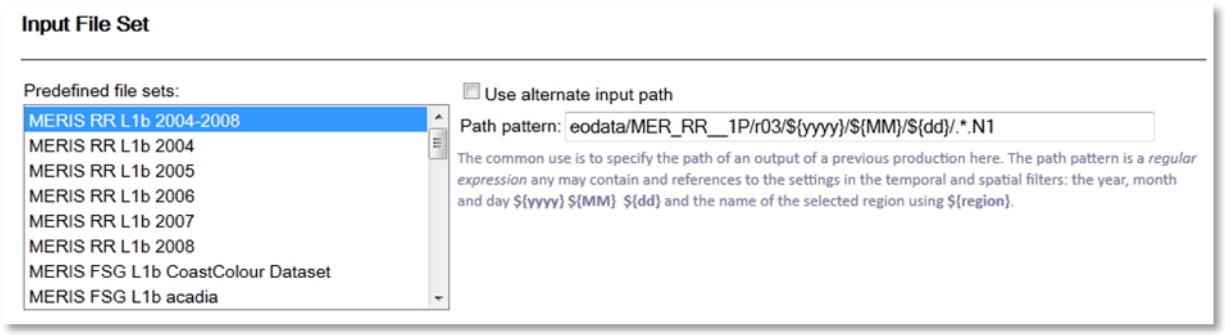

Figure 25: Input file set

Alternatively, users can specify an input path in a text box. The common use is to specify the path of an output of a previous Calvalus production. The path pattern is also a regular expression and may contain the references to the settings in the temporal and spatial filters: the year, month and day **${yyyy}, ${MM}**, **${dd}** and the name of the selected region using **${region}**.

7.2 Spatial and Temporal File Filters

The files determined by the input file set can be further limited by specifying a temporal filter comprising either a date range or a list of single days, Figure 26. Single days are very useful for testing L2 or L3 processing on a small subset of files before ordering a larger number of files, that may take some time to process.

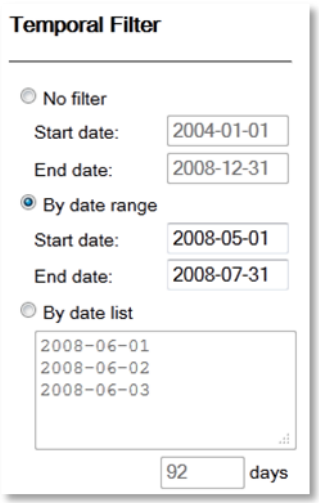

Figure 26: Temporal file filter

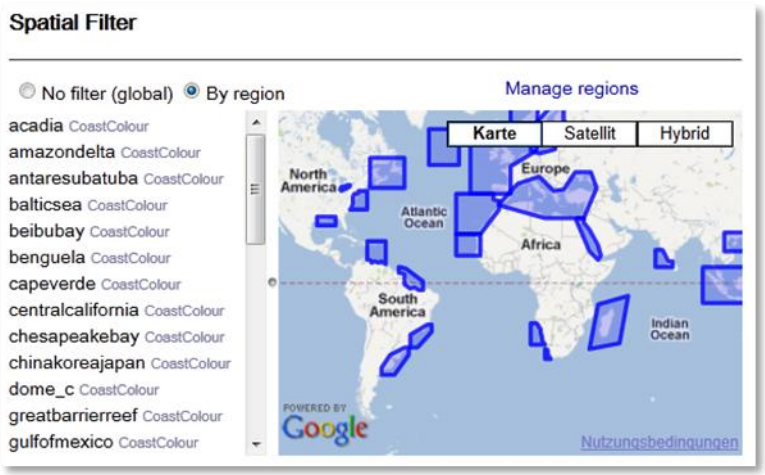

Figure 27: Spatial file filter

The spatial filter is used not only to filter out files but also to create spatial subsets of the input data before the further processing takes place, Figure 27. Users can define their own regions by using the region manager.

7.3 Level-2 Processor and Parameters

The Calvalus system has been designed to be easily extended to new data processors developed using the BEAM Graph Processing Framework [RD-2] as well as executable/shell scripts. For BEAM GPF, one or more compiled processors are packed as Java archive files (JARs) in a *Calvalus processor bundle* and installed on the Hadoop cluster using the Calvalus **cpt** command-line tool. The processors that are currently installed are shown in the *Level-2 Processor* list, Figure 28.

Figure 28: Level-2 processor list

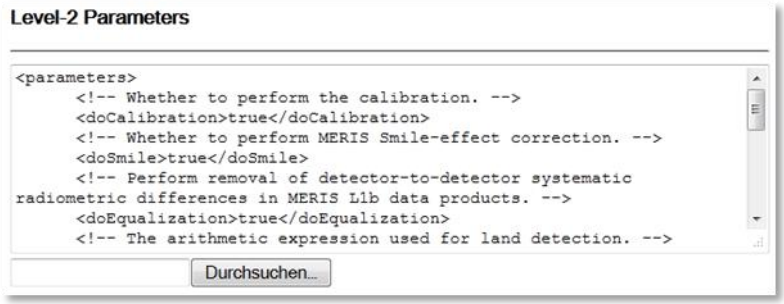

Figure 29: Level-2 parameters

The area *Level-2 Parameters* lets user specify in text format the processor parameters used for a L2-production. The default set of parameters is processor-dependent and read form the processor bundle's metadata.

Currently the Calvalus system uses various CoastColour processors comprising a radiometric correction and pixel classification (L1P), an atmospheric correction using neural networks (L2R), and a L2 IOP and chlorophyll retrieval using neural networks (L2W) by default but also including a parameter switch to perform the QAA IOP retrieval. L2W comprises the L1P and L2R steps and also their outputs.

It is planned to also integrate other processors into the portal in the near future: for example NASA's **l2gen** and ESA's **MEGS (ODESA)** processors.

7.4 Output Parameters

The Level-2 and Level-3 processing production types generate data product files. The *Output Parameters* component is primarily used to specify the output EO data file format, see Figure 30. Currently BEAM-DIMAP, NetCDF and GeoTIFF are supported. Optionally users can specify an output directory, this is especially useful if the result shall serve later as an input to another production type. For example, users can perform Level-2 processing into a dedicated directory. Then, for Level-3 processing they provide that directory as an input path to the Level-3 processing. If left empty, a temporary output directory is used. Finally, users can select whether they

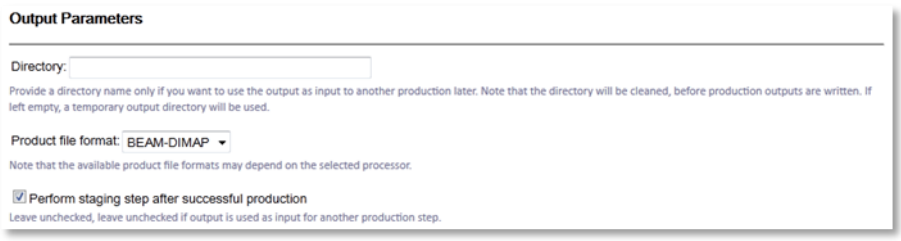

Figure 30: Output parameters

want to perform staging of produced files immediately after the processing is finished. Staging is a process of copying and reformatting the output files to a user-accessible location.

7.5 Check Request and Order Production

All four production scenarios have two common buttons, namely *Check Request* and *Order Production* which are located at the bottom of each production tab.

Check Request is used to validate the production request, and if it is valid, display the production summary that is used to order a given production. *Order Production* first validates the production request and, if it is valid, it submits the order to the server.

7.6 Production Manager

As its name suggests, *Production Manager* is used to manage scheduled, running and completed productions. Once a production request is submitted and the order accepted by the Calvalus server, the production is given a unique ID and it is displayed in the production table.

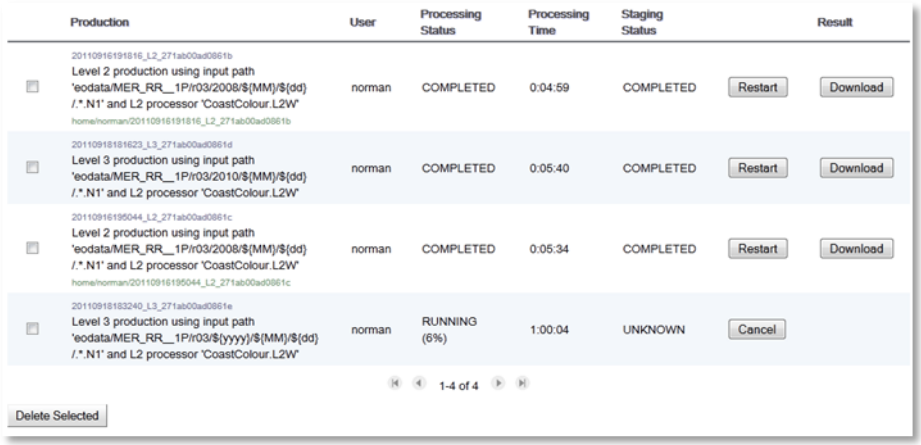

Figure 31: Production manager

Accidently submitted productions can now be cancelled while in a scheduled or running state. Productions that are not used anymore can be selected and then deleted. Details of a production request can be displayed at any time by clicking on a row in the *Production* column.

8 Achievements and Results

The Calvalus study has shown that Apache Hadoop with its MapReduce programming model and its distributed file system is a very suitable foundation for the development of high performance EO data processing systems.

For Level-3 bulk processing, the MapReduce programming model has been fully exploited because the binning algorithms exactly match the class of problems that are solved by MapReduce and can thus be most efficiently parallelised on Hadoop clusters. Consequently, Level-3 processing performs very fast on the Calvalus production system. For example, a global 10-day chlorophyll map can be processed in less than 1.5 hours including the processing from Level-1 to Level-2. Around 140 full orbit scenes are processed in this case. For the same processing request, it usually takes 20 to 30 minutes to process a single Level-1 orbit to Level-2 on a single node.

For Level-2 bulk processing, the MapReduce programming model is not exploited, because no actual data “reduction” takes place. Input files are transformed directly into output files. In the MapReduce model, large files are usually split into blocks and distributed over multiple nodes in the HDFS. Then, single splits aligned with these blocks are processed independently and in parallel. However, in the Calvalus processing, L2 files are stored as single blocks (see discussion in chapter 4.2, Level-2 Bulk Processing). Still, the L2 processing benefits of Hadoop job scheduling mechanism, which very successfully executes mapper tasks so that they run close to where their input data are stored. This means, that on a cluster comprising 20 nodes, 20 Level-2 processing jobs can run in parallel. This confirms the most important advantage of the system, run data local. Due to this fact, the performance of the parallelisation scales nearly linearly with the number of nodes in the cluster.

The capability to execute Level-2 processing steps on-the-fly, makes Calvalus an ideal platform for running processing tasks and analyses on mission wide datasets. Calvalus also allows users to modify Level-2 processing parameters and to run different processor versions.

The Calvalus system has already shown its operational capabilities in the CoastColour project, where it is used to generate various validation datasets starting from MERIS FRS Level-1b data. In the Land Cover CCI project, Calvalus is used to generate the Round-Robin dataset. The processing includes ortho-rectification of MERIS FRS with AMORGOS and Level-2 surface reflectance retrievals, followed by the generation of Level-3 pre-classified land coverage maps.

More detailed information on the Calvalus study is provided in these document deliverables:

1. The **Calvalus Requirements Baseline** [RD 22] serves as the primary source for the system technical specification and final acceptance testing. Particularly, the requirements baseline reflects ESA's and Brockmann Consult’s common understanding of the study goal and describes the expectations on the outcome.
2. The **Calvalus Technical Specification** [RD 23] describes how the Apache Hadoop MapReduce engine and the Hadoop Distributed File System are integrated into a system of services for cal/val data and processing management. It describes in detail the production scenarios to be implemented and the various trade-off analyses and technology studies performed.
3. The **Calvalus Acceptance Test Plan** [RD 24] comprises the end-to-end testing of the system that has been implemented according to the Requirements Baseline and the Technical specification.

9 Conclusion and Outlook

The result of the study is simple:

- Yes, the MapReduce programming model and the Distributed File System can be applied to Earth Observation data with large benefit to processing performance and reliability
- Yes, the Calvalus cluster with its combination of commodity computers and Hadoop provides the potential to efficiently support cal/val tasks as well as user services, such as full mission reprocessing

These are the foundations to continue with Calvalus into the future. Beyond the current LET-SME study, three lines of activities can be identified:

1. Use the current Calvalus system to support ESA activities, in particular the DUE CoastColour project and the CCI projects on Land Cover and Ocean Colour. These projects work on the MERIS data which are currently available on Calvalus. The two marine projects require exactly the validation tests, “match-ups” and “trend analyses”, which have been implemented as use cases within this study. The Ocean Colour CCI project plans to rely on Calvalus as on a rapid development platform during the next 2 years. Additionally, the system will be presented to ESA Data Quality Working Groups (MERIS and AATSR have been invited to the final presentation). If interested, the current cluster can also support their activities.
2. Continue the technological development, in order to raise the idea from a proof-of-concept implementation (technology study) to a prototype. This requires improvement of current map-reduce algorithms and implementation of new ones, such as classification and information extraction, application to other sensors and input data formats. The prototype should also run on significantly larger hardware.
3. Prepare a marketable product and service. The Calvalus system will be an appropriate environment to work on the large amount of future Earth Observation data, e.g. from ESA Sentinel missions, or national missions such as EnMAP. Ideally a first commercial system and service should be on the market within the next 2 years.

10 References

[RD 1]	Fomferra, N.: The BEAM 3 Architecture; http://www.brockmann-consult.de/beam/doc/BEAM-Architecture-1.2.pdf
[RD 2]	Brockmann, C., Fomferra, N., Peters, M., Zühlke, M., Regner, P., Doerffer, R.: A Programming Environment for Prototyping New Algorithms for AATSR and MERIS – iBEAM; in: Proceedings of ENVISAT Symposium 2007, ESRIN Frascati, Italy
[RD 3]	Fomferra, N., Brockmann C. and Regner, P.: BEAM - the ENVISAT MERIS and AATSR Toolbox; in: Proceedings of the MERIS-AATSR Workshop 2005, ESRIN Frascati, Italy
[RD 4]	Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters; OSDI'04: Sixth Symposium on Operating System Design and Implementation; San Francisco, CA, 2004 (http://labs.google.com/papers/mapreduce.html)
[RD 5]	Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung: The Google File System; in: 19th ACM Symposium on Operating Systems Principles, Lake George, NY, 2003 (http://labs.google.com/papers/gfs.html)
[RD 4]	Ariel Cary, Zhengguo Sun, Vagelis Hristidis, Naphtali Rishe: Experiences on Processing Spatial Data with MapReduce; Lecture Notes In Computer Science; Vol. 5566 - Proceedings of the 21st International Conference on Scientific and Statistical Database Management - New Orleans, LA, USA, 2009 (http://users.cis.fiu.edu/~vagelis/publications/Spatial-MapReduce-SSDBM2009.pdf)
[RD 5]	R. Doerffer, H. Schiller: MERIS Lake Water Algorithm for BEAM and MERIS Regional Coastal and Lake Case 2 Water Project, Atmospheric Correction ATBD; ESRIN Contract No. 20436
[RD 6]	Zhong Ping Lee, Kendall L. Carder, and Robert A. Arnone: Deriving inherent optical properties from water color: A multiband quasi-analytical algorithm for optically deep waters; APPLIED OPTICS, Vol.41,No.27
[RD 7]	Bryan A. Franz, Sean W. Bailey, P. Jeremy Werdell, and Charles R. McClain: Sensor-independent approach to the vicarious calibration of satellite ocean color radiometry; APPLIED OPTICS Vol.46,No.22, 1
[RD 8]	Bryan Franz: Methods for Assessing the Quality and Consistency of Ocean Color Products; NASA Goddard Space Flight Center, Ocean Biology Processing Group http://oceancolor.gsfc.nasa.gov/DOCS/methods/sensor_analysis_methods.html
[RD 9]	Janet W. Campbell, John M. Blaisdell, Michael Darzi: Level-3 SeaWiFS Data Products: Spatial and Temporal Binning Algorithms; SeaWiFS Technical Report Series, NASA Technical Memorandum 104566, Vol. 32
[RD 10]	K. Barker et al: MERMAID: The MEris MAtchup In-situ Database; ARGANS Limited (http://hermes.acri.fr/mermaid/doc/Barker-et-al-2008_MERMAID.pdf)
[RD 11]	NASA OBPG: Ocean Color Level 3 Binned Products (http://oceancolor.gsfc.nasa.gov/DOCS/Ocean_Level-3_Binned_Data_Products.pdf)
[RD 12]	CoastColour web site (http://www.coastcolour.org/)
[RD 13]	ECSS-E-ST-40C ECSS Space Engineering - Software, European Cooperation for Space Standardization, ESA-ESTEC, Noordwijk, The Netherlands

[RD 14]	Bryan Franz: OBPG l2gen User's Guide; (http://oceancolor.gsfc.nasa.gov/seadas/doc/l2gen/l2gen.html)
[RD 15]	Web site of the ESA Climate Change Initiative (http://earth.eo.esa.int/workshops/esa_cci/intro.html)
[RD 16]	OGC Web Processing Service Specification (http://www.opengeospatial.org/standards/wps)
[RD 17]	Case2R source code repository at https://github.com/bcdev/beam-meris-case2
[RD 18]	QAA source code repository at https://github.com/bcdev/beam-meris-qaa
[RD 19]	BEAM user manual (http://www.brockmann-consult.de/beam)
[RD 20]	Sean W. Bailey, P. Jeremy Werdell: A multi-sensor approach for the on-orbit validation of ocean color satellite data products; Remote Sensing of Environment 102 (2006) 12–23
[RD 21]	DUE CoastColour Product User Guide, http://www.coastcolour.org/documents/Coastcolour-PUG-v2.1.pdf
[RD 22]	Calvalus Requirements Baseline, Version 1.2.1, 16. July 2010, http://www.brockmann-consult.de/calvalus/documents/Calvados-RB-1.2.1-20100716.pdf
[RD 23]	Calvalus Technical Specification, Version 1.2.0, 21. March 2011, http://www.brockmann-consult.de/calvalus/documents/Calvalus-TS-1.2-20110221.pdf
[RD 24]	Calvalus Acceptance Test Plan, Version 1.1.1, 31. October 2011, http://www.brockmann-consult.de/calvalus/documents /Calvalus-ATP-1.1-20111012.pdf