UTILIZING MONITORING AND REPORTING TECHNIQUES IN DATA PROCESSING SYSTEMS TO IMPROVE THE VALUE ADDING OF DATA

Johanna Senft, Henrike Barkmann, Sven Stönner, Max Wegner

German Aerospace Center
German Remote Sensing Data Center
Oberpfaffenhofen, D-82234 Weßling, Germany
Neustrelitz, D-17235 Neustrelitz, Germany

Abstract

Every day, more than 20 TB of remote sensing data is processed at DLR's Remote Sensing Data Center (DFD) using more than 70 processing chains. However, the input data is of varying quality as it may sometimes be degraded by both external and internal influences. Thus, an accurate and effective monitoring and reporting mechanism is vital to ensure reliable data processing, adequate performance, effective operation, and stable resource usage, among others. The purpose of this paper is to describe the monitoring and reporting mechanism employed by DFD's Monitoring & Reporting System - the DFD M&R. Herein, the DFD M&R's application to ionospheric data provided by the Ionosphere Monitoring and Prediction Center (IMPC) is used as a case study to assess the DFD M&R's effectiveness.

This paper is structured into four sections. Section one describes the data development process used by the IMPC. Section two describes the DFD M&R system. Section three, applies the DFD M&R to a selected part of the IMPC data process – the IMPC-ROTI scientific processor (IMPC-ROTI). Section four discusses the case study, assesses the DFD M&R's limitations and provides an overview of future developments.

SECTION I: INTRODUCTION

The IMPC is developed by the DFD and the Institute for Solar-Terrestrial Physics and acts as a near real-time information and data service on the current state of the Earth's ionosphere [RD-1] [RD-2][RD-3]. The IMPC delivers forecasts and warnings about the ionosphere's prevailing conditions. This information is needed, because ionospheric disturbances affect the performance of radio systems used in space-based communication, navigation and remote sensing. Furthermore, ionospheric disturbances may degrade the accuracy, reliability and availability of Global Navigation Satellite Systems (GNSS), such as GPS or Galileo. Thus, the IMPC's near real-time forecasts and warnings serve to improve navigation and communication systems. International cooperation and data exchange are crucial for operating the IMPC. Therefore, the IMPC is involved in the Space Weather European Network (SWENET) of ESA and maintains intensive relationships with many European and international facilities in the space weather domain. An overview of the IMPC is shown in Figure 1.

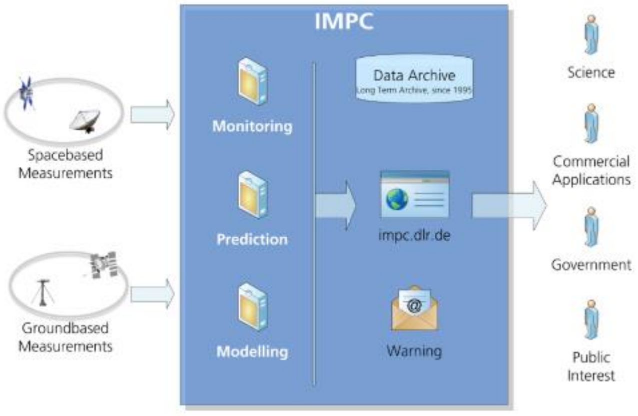


Figure 1: The Ionosphere Monitoring and Prediction Center (IMPC)

The IMPC data process permanently monitors the electron density and its structure in the ionosphere-plasmasphere system, using ground and space-based antennas. This data is used by several IMPC Scientific Processors to develop IMPC products such as the Total Electron Content Maps (IMPC-TEC) – which displays the number of electrons per square meter - or the Rate of Change of TEC index (IMPC-ROTI). A conceptual overview of IMPC data process is shown in Figure 2.[RD-4].

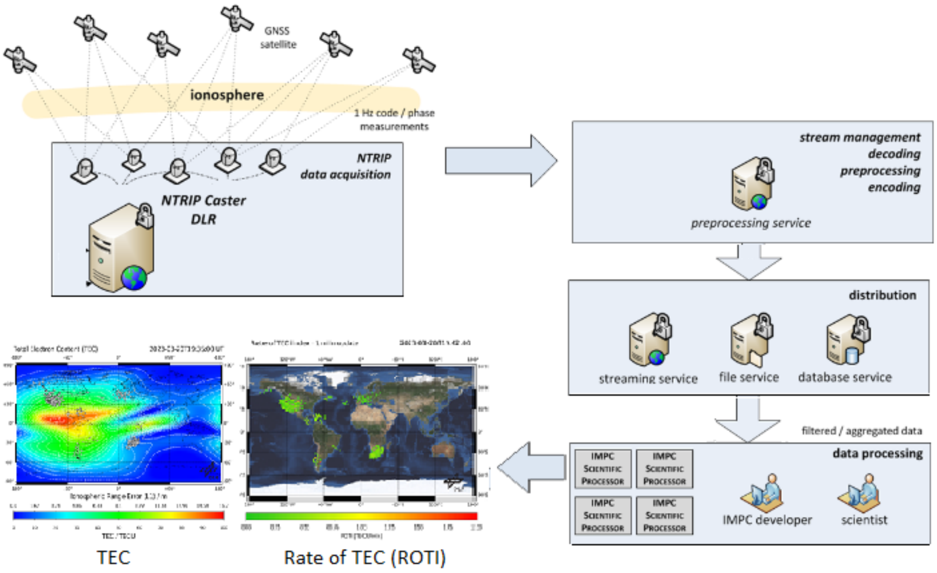


Figure 2: The IMPC data process

Research Approach - Challenges

Daily, more than 50 gigabytes of ionosphere-related data are processed and archived; 1588 products for GNSS users, 2888 ROTI products and 288 TEC products are generated and disseminated [RD-2]. Like all software and hardware systems dealing with such amounts of data, the IMPC data process chains are vulnerable to errors – introduced by hardware issues or software malfunctioning. The IMPC data process requires monitoring and reporting to ensure the well-functioning and performance of the deployed systems as well as the overall data quality. [RD-5][RD-6][RD-7]

Technical Background

The IMPC-ROTI scientific processor is built as a Kubernetes Cluster with a monitoring data collecting agent (Telegraf) inside each node. The monitoring and reporting components (InfluxDB, Grafana, Reporting System) lie outside the cluster, as shown in Figure 3.

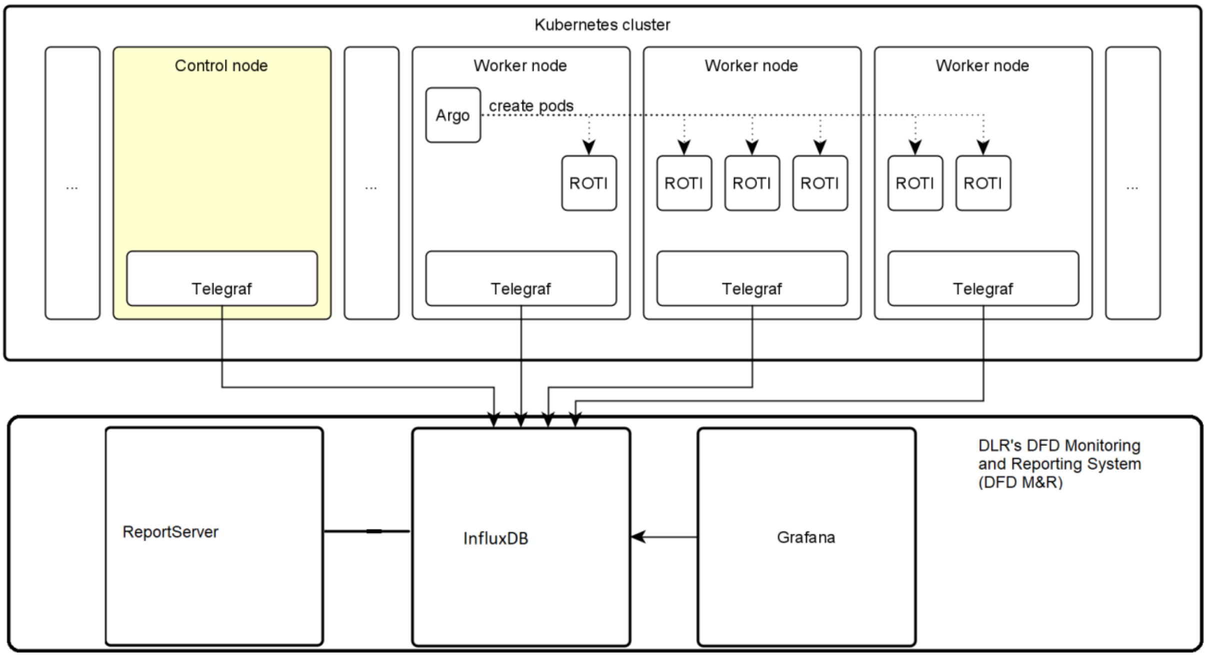


Figure 3: Kubernetes Cluster for IMPC-ROTI scientific processor with monitoring and reporting components

Kubernetes is an open-source system to automate deployment, scaling and management of containerized applications [RD-8]. In Kubernetes, *pods* are the smallest deployable and manageable units. A *pod* is a group of one or more *containers* with shared resources and a single "how-to-run" specification. A Kubernetes Cluster consists of a set of worker machines, physical or virtual servers (*worker nodes*) that run the *pods*. A *pod* represents a standardized, runnable and deployable software unit, including all of its dependencies. A control plane (*control node*) manages the *worker nodes* and the *pods* in the cluster. Kubernetes functions as a scalable orchestration system, which manages a given software based on a clear *how-to-do-it description*. For example, Kubernetes ensures a given software runs simultaneously on three different servers.

Specifically for this case study, the configured Kubernetes cluster uses three *controller-nodes* and three or more *worker-nodes*. The *controller-nodes* include the API service, the cluster state and a scheduler. The *worker-nodes* are used to run any kind of container-based workload.

Inside Kubernetes, the **Argo-Workflow** engine is used and runs as a *container-deployed* service. Argo-Workflow is an open-source *container-native* workflow engine for orchestrating parallel jobs on Kubernetes [RD-9]. A workflow defines the sequential or parallel execution of a set of distinct steps. The IMPC-ROTI scientific processor chain - used in this case study - is shown in Figure 4, and consist of five key steps: read data from a data source; transform data; generate additional information; transfer data into a long-term archive; upload data to a remote data sink.

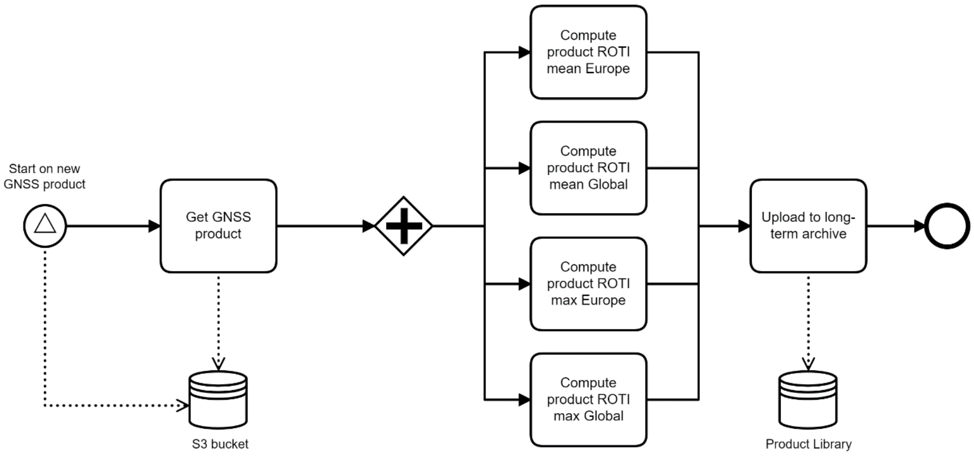

Start on new GNSS product
Get GNSS product
S3 bucket
Compute product ROTI mean Europe
Compute product ROTI mean Global
Compute product ROTI max Europe
Compute product ROTI max Global
Upload to long-term archive
Product Library

Figure 4: Argo-Workflow for IMPC-ROTI scientific processor chain

The Argo-Workflow runs one *container* for each workflow step. More specifically, it creates the container description based on the workflow definition, then passes it to Kubernetes for orchestration and scheduling.

SECTION II: THE DFD M&R PROCESS

Having established IMPC-ROTI scientific processor, the next step describes the DFD M&R process, which consists of four successive steps, as shown in Figure 5.

The DFD M&R described herein, is applied to one special part of the IMPC data process – the **IMPC-ROTI** scientific processor (IMPC-ROTI) – which calculates the standard deviation of the rate of TEC products [RD-5] [RD-7]. The Application of DFD M&R to IMPC-ROTI consists of four successive steps. Firstly, agents collect, extract, and transform the monitoring input data. Secondly, the transformed monitoring input data is stored into a time-series database. Thirdly, the monitoring input data is visualized to establish understanding for the end-user. Fourthly, key statistics of the IMPC-ROTI scientific processor are reported. This is shown in Figure 5.

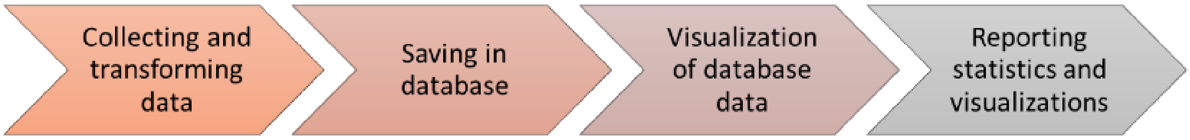

Collecting and transforming data
Saving in database
Visualization of database data
Reporting statistics and visualizations

Figure 5: The DFD Monitoring and Reporting process

Data Collection

DFD M&R uses agents to collect data – specifically Telegraf. **Telegraf** is a service-based agent used for data collection, transformation and dissemination of metrics and events. Its architecture is based on *plugins* for input and output metrics. Telegraf's large library of *plugins* allows for easy configuration of a collection of metrics from many different sources and metrics dissemination to major databases and applications for different purposes. [RD-10] When applying DFD M&R to IMPC-ROTI, Telegraf collects standard system information, Kubernetes metrics and Argo Workflow metrics. As seen in Figure 3, Telegraf runs on each node to collect metrics from each node itself as well as the Kubernetes containers running on that node.

Data Storage

Following a preconfigured schedule, all collected data is actively pushed into an externally deployed database which does not run inside of Kubernetes. As part of DFD M&R, InfluxDB is used. **InfluxDB** is a time series database system (TSDB), especially made for data that can be evaluated as a time-series. InfluxDB consist of *Buckets* which form the individual databases of the database system. InfluxDB offers the built-in functionality to work with aggregation, down-sampling, data lifecycle management and summarization. Querying InfluxDB is performed using *Flux-lang* - a popular functional data scripting language. [RD-11][RD-12]

To use DFD M&R for the IMPC-ROTI scientific processor, the InfluxDB is configured with one *Bucket* containing all received data stored during different measurements to reduce complexity. Such configuration can be extended to more buckets – similar to tables in common SQL databases.

Data Visualisation

Having used Telegraf to collect data and InfluxDB to store the data, the third step of the DFD M&R process is to visualize the data which is done using Grafana. **Grafana** – being part of a complete software application stack – is an open-source visualization platform for querying, exploring and analyzing, metrics, logs and traces wherever they are stored. Connected to various data sources, Grafana offers tools to visualize the stored time-series into meaningful graphs. How Grafana organizes information into *dashboards* and *panels* is shown in Section III [RD-13]. As part of the DFD M&R, Grafana uses InfluxDB as a data source to display the collected data in an operator friendly manner, so that the operator can understand the current state of the Kubernetes cluster and the workflows running within.

Data Reporting

In general, suitable reports cannot be extracted directly from Grafana Open Source (OS version). However, Grafana Enterprise and Cloud Pro versions enable the automatic report generation as *pdf-files* from any existing dashboard and offer the option to disseminate these reports via E-Mail. In the OS version, the extraction and download of *csv-Data* of each dashboard panel is a rough but useful option for creating reports. Using *csv-files* for data download allows for quick reporting directly derived from the visualized data out of a Grafana dashboard panel. Thus, Grafana OS can be used as visualization tool and it offers some reporting capabilities. However, generating full-service reports are best assembled using another route.

Instead of involving Grafana into the reporting process itself, DFD M&R uses an additional tool – ReportServer. The **ReportServer** is a Business Intelligence Suite platform, provided in a basis version as open-source software. It integrates and offers many reporting options from within one single user interface [RD-14]. As part of the DFD M&R, the ReportServer compiles reports by accessing the InfluxDB directly. Using the ReportServer, the reports can be disseminated in many formats to users based on scheduling. They can also be stored in team spaces and can be archived.

SECTION III: APPLYING DFD M&R TO IMPC-ROTI SCIENTIFIC PROCESSOR

Having described the IMPC-ROTI scientific processor and the DFD M&R, this section forms the synthesis of the two previous sections - showcasing the application of the DFD M&R process to the IMPC-ROTI scientific processor. As mentioned before the input data of IMPC-ROTI may be sometimes degraded by both external and internal effects.

During the adaption of the DFD M&R process to the IMPC-ROTI scientific processor, the authors identified three core problems plaguing IMPC-ROTI: the overload of computing systems, the effects of the introduction and deployment of new product computing steps inside the IMPC-Roti scientific processor, and problems due to failed interoperability of tools inside the Kubernetes cluster.

By applying the DFD M&R to the IMPC-ROTI, these problems are identified automatically and reported to operators. Thereby, the DFD M&R process enables operators to mitigate each core problem and increase the overall quality of data operation. This is achieved by applying the previously described four steps of the DFD M&R: data collection, data storage, data visualisation, and reporting.

Data Collection

Three kinds of metrics are collected to properly monitor the IMPC-ROTI scientific processor: system metrics, Kubernetes metrics, and Argo Workflow Metrics. Firstly, system metrics are key indicators for the system’s health. Secondly, the Kubernetes metrics describe the Kubernetes resources and are used to monitor the Kubernetes cluster’s health. Lastly, the Argo Workflows Application (Argo-WF) provides two kinds of metrics: *controller metrics* and *custom metrics*. However, as part of the DFD M&R, in its current iteration, only *controller metrics* are used and describe the state of the Argo WF.

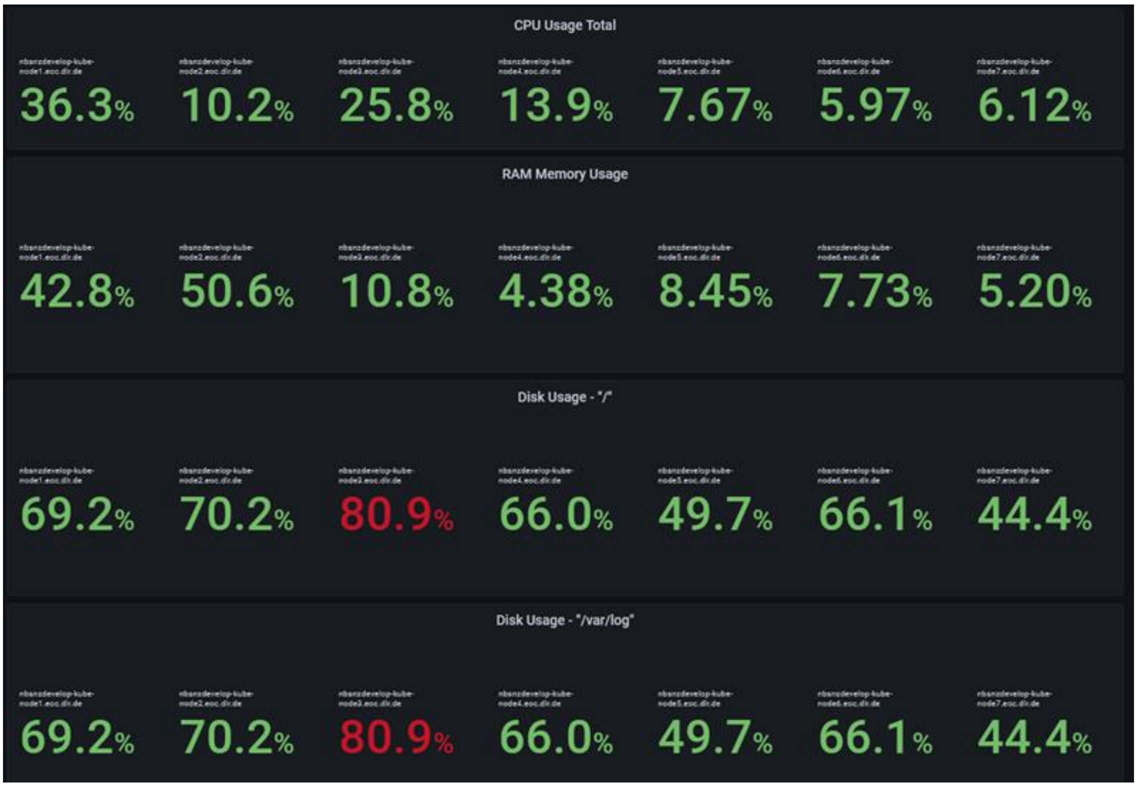
Data Storage

All collected metric data are stored in the configured InfluxDB *Bucket*.

Data Visualization with Grafana-Dashboards

The acquired and stored data is visualized using Grafana-Dashboards. As part of DFD M&R, a set of dashboards and panels were configured specifically for the IMPC-ROTI scientific processor. These dashboards show the collected system, Kubernetes and Argo-WF metrics. First and foremost, these dashboards focus on the monitoring of the Argo Workflows, their interaction with Kubernetes and the observation of system components. These Grafana-Dashboards visualize the monitoring process and thus serves as a key element in the human-machine interaction. The DFD M&R results in four dashboards: The Quick Overview Dashboard, the Workflow Status Dashboard, the Used Components Dashboard, the Kubernetes Dashboard. Each dashboard is made up of multiple panels.

One core problem for the IMPC-ROTI scientific processor is the recognition of system overload caused by extensive system usage by competitive users. Therefore, it is necessary to monitor the CPU-, Memory-, and Disk-Usage inside the whole Kubernetes Cluster. For this purpose, **the Quick Overview Dashboard** was created; there operators can monitor the system load and identify an impending system overloading. Within this dashboard, a set of the panels shows the CPU-, RAM Memory Usage- and Disk Usage. Their configuration is set at a threshold of 80%; with greater values being highlighted in red to instantly alert the operator. Furthermore, alarm situations are immediately reported to operators via E-Mail. Doing so means that a potential system overload can easily be noticed and thus a system crash be averted. These panels are shown in Figure 6.



CPU Usage Total

nbanzdevelop-kube-node1.eoc.dlr.de	nbanzdevelop-kube-node2.eoc.dlr.de	nbanzdevelop-kube-node3.eoc.dlr.de	nbanzdevelop-kube-node4.eoc.dlr.de	nbanzdevelop-kube-node5.eoc.dlr.de	nbanzdevelop-kube-node6.eoc.dlr.de	nbanzdevelop-kube-node7.eoc.dlr.de
36.3%	10.2%	25.8%	13.9%	7.67%	5.97%	6.12%

RAM Memory Usage

nbanzdevelop-kube-node1.eoc.dlr.de	nbanzdevelop-kube-node2.eoc.dlr.de	nbanzdevelop-kube-node3.eoc.dlr.de	nbanzdevelop-kube-node4.eoc.dlr.de	nbanzdevelop-kube-node5.eoc.dlr.de	nbanzdevelop-kube-node6.eoc.dlr.de	nbanzdevelop-kube-node7.eoc.dlr.de
42.8%	50.6%	10.8%	4.38%	8.45%	7.73%	5.20%

Disk Usage - "/"

nbanzdevelop-kube-node1.eoc.dlr.de	nbanzdevelop-kube-node2.eoc.dlr.de	nbanzdevelop-kube-node3.eoc.dlr.de	nbanzdevelop-kube-node4.eoc.dlr.de	nbanzdevelop-kube-node5.eoc.dlr.de	nbanzdevelop-kube-node6.eoc.dlr.de	nbanzdevelop-kube-node7.eoc.dlr.de
69.2%	70.2%	80.9%	66.0%	49.7%	66.1%	44.4%

Disk Usage - "/var/log"

nbanzdevelop-kube-node1.eoc.dlr.de	nbanzdevelop-kube-node2.eoc.dlr.de	nbanzdevelop-kube-node3.eoc.dlr.de	nbanzdevelop-kube-node4.eoc.dlr.de	nbanzdevelop-kube-node5.eoc.dlr.de	nbanzdevelop-kube-node6.eoc.dlr.de	nbanzdevelop-kube-node7.eoc.dlr.de
69.2%	70.2%	80.9%	66.0%	49.7%	66.1%	44.4%

Figure 6: The Quick Overview Dashboard

Another core problem of the IMPC-ROTI scientific processor arises during the introduction and deployment of new product computing steps inside the IMPC-Roti scientific processor (and on the Argo-

WF), e.g., a new ROTI product calculation. In these cases it is vital to monitor the effects these deployments have on the *Kubernetes cluster* over the time. Monitoring can be done using panels inside the **Workflow Status Dashboard** and the **Used Components Dashboard.** For example, an increase of erroneous Argo-WF is displayed in *the WF Error – 1m Rate panel* (inside the Workflow Status Dashboard), so that the operator can immediately recognize when errors are caused by the new computing steps. This is shown in Figure 7.

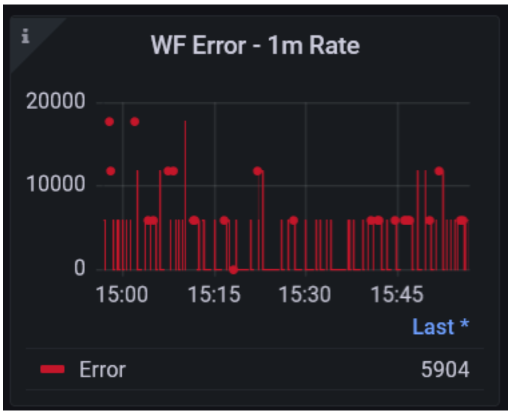


Figure 7:The WF Error – 1m Rate panel

Introducing new computing steps can also impact the WF duration of operation. This is shown in the *WF durations of operation – 1m Rate* panel (inside the Workflow Status Dashboard). There, operators can easily set the threshold for the durations of operations to an appropriate value, ensuring full customization and control. Any alarm situation is, once again, reported via E-Mail.

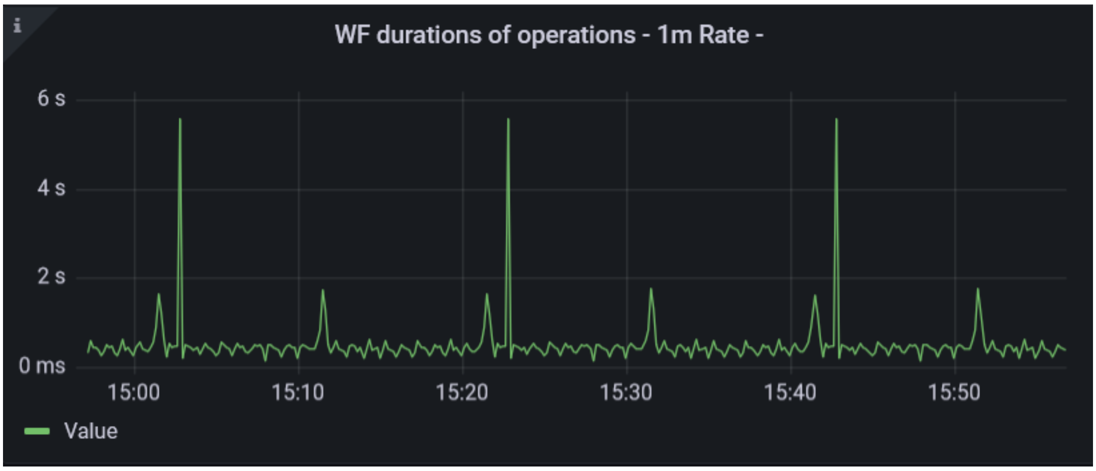


Figure 8: WF durations of operation – 1m Rate

Another core problem of the IMPC-ROTI scientific processor is the detection of bottlenecks in the processing workflow. Therefore, the **Used Components Dashboard** offers a panel showing the additions to the workflow queue and another showing the latency of the processing queue. Here again, operators can easily set appropriate threshold and alarm values, so that bottlenecks can be immediately recognized and reported.

In order to assure a stable data operation flow, where data are processed without disturbances inside the Kubernetes cluster, it is important to monitor the interoperation between Argo-WF and Kubernetes. This can be done by using the **Kubernetes Dashboard**. Inside this dashboard The *Requests to Kubernetes -1m Rate* panel shows the total amount of API requests sent to the Kubernetes API. With this panel an exceeding system load can be recognized and avoided. Errors in the operation of Kubernetes and Argo-WF can be seen by the *number of pods with more than 10 restarts* panel, which shows the number of pods restarted by Kubernetes. A huge value indicates an error in the operation of Kubernetes and Argo-WF. In addition, missing pods in Argo-WF, which are terminated by Kubernetes are an indication of operation failure. The Kubernetes Dashboard can be seen in Figure 9.

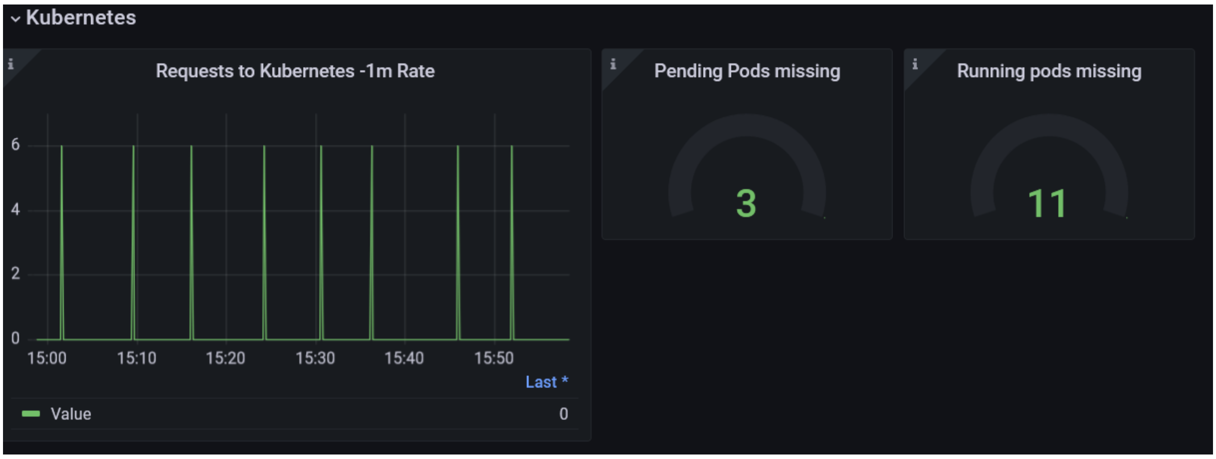


Figure 9: The Kubernetes Dashboard

Reporting

Following the visualization of the monitoring data relevant for the IMPC-ROTI scientific processor mechanism, their information is then included in reports based on clients' requirements.

Reports of the IMPC-ROTI scientific processor are accessible from the DLR DFD's ReportServer. The reports are designed to report about the identified core problems discussed before. The reports are addressed to operators, developers, and project leaders. Therefore, three major reports have been established.

The system usage report containing a daily overview of CPU-, Memory-, and Disk-Usage inside the whole Kubernetes Cluster, the health of Argo-WF inside the Kubernetes Cluster, the status of Argo-WF's, and the WF components usage. The system usage report is sent to designated operators. **The system error** report informs software and system developers daily about occurred failures and errors. **The resources report** is sent monthly to project leaders and software/system developers. This report informs about the used resources and is a good instrument for future resources planning for the introduction of new computing steps to calculate new products in the IMPC-ROTI scientific processor.

SECTION IV: OBSERVATIONS & DISCUSSION

So far, the DFD M&R process has been introduced and the IMPC-ROTI scientific processor has been described. Then the M&R process has been applied to the IMPC-ROTI scientific processor in an effort to showcase its use. This section will identify the M&R's benefits and drawbacks specific to the IMPC-ROTI scientific processor and discuss their implications beyond this case study.

Discussion Benefits and Drawbacks

The application of the DFD M&R process on the IMPC-ROTI described here focused firstly on the identification of an impending system overload, resulting in an increase of the systems stability. Secondly, the focus lies on the effects on the system in case of the introduction of new computing steps inside IMPC-ROTI, resulting in an improved system performance. Thirdly, by monitoring the interoperation inside the Kubernetes cluster, the system stability is further increased. All these interventions result in a better data quality, a more reliable data operation, and an uninterrupted data flow.

However, monitoring complex Kubernetes clusters, workflows, system information and even log-file information can create a large amount of data. Large data volumes inevitably strain the system load, the application capacity, data traffic, and can be memory consuming. Thus, if the amount of data is too large, the M&R process may slow down the entire system – potentially reducing data quality – or, in a worst-case scenario, cause the whole system to crash. Therefore, it is necessary to carefully consider the collected metrics, the retention policy of stored data, and their cardinality in databases.

The tools of the DFD M&R process provide some useful techniques to mitigate this risk. For example, Telegraf offers metric filtering techniques, so that only useful metrics are collected and disseminated, thereby, reducing the data traffic and preventing system overload. Additionally, InfluxDB *Buckets* can be configured with adequate retention policy so that data with timestamps older than a threshold value are deleted, further reducing the amount of stored data and preventing storage systems overflow. InfluxDB also offers a special parameter for measuring the high series cardinality, which can be monitored with a Grafana Dashboard. Thereby, a slowdown of reads and writes to InfluxDB can be prevented, again reducing the data traffic, and preventing a system overload.

Outlook and Future Development

As discussed in Section III, the Argo-Workflow metric system provides two kinds of metrics, however, only the controller metrics are used. In the future, the addition of the other metrics – the **Custom metrics** – to the DFD M&R process, could offer distinct benefits. Specifically, for each computing step inside a scientific processor, user-defined metrics can be introduced. They could be used to monitor and report on the specific computing step inside the scientific processor. Obviously, such additional features have to be defined carefully in order to avoid system overload.

In addition, monitoring and analyzing log-messages on the same operating platform as the provided system and application metrics could be provided by a modern monitoring system. In this context, a log-shipping agent – e.g., Promtail – could be installed. This agent would send log information to a log aggregation tool such – e.g., Loki. Such tools could be used inside Grafana’s visualization platform for the detailed exploration of log-files. Unfortunately, the standard configuration may prove insufficient and could require certain environment-specific optimizations.

Lastly, monitoring and reporting procedures could also be extended to components outside of the Kubernetes cluster. Collecting data from other data sources, data storage information or other application are also important future tasks and can improve data throughput and the system reliability.

CONCLUSIONS

This paper discussed methods to ensure the reliability of data processing chains used at DLR’s Remote Sensing Data Center (DFD). This is necessary because input data is of varying quality, as it may be degraded by both external and internal effects. Therefore, the application of DLR’s DFD Monitoring and Reporting System (DFD M&R) to a specific data process system – in this case the IMPC-ROTI scientific processor – was described and its impacts, benefits and drawbacks were assessed.

Overall, applying the DFD M&R process to a complex data processing system results in substantial improvements. Primarily, these improvements lie in an increased system reliability, faster troubleshooting, data throughput optimization, reduced operation efforts, and improved data consistency and quality. Together with log-file analysis, the extension of collected metrics on application and external components, applying the DFD M&R process can improve essentially all data processing chains. Nevertheless, any additional operation also requires computing power and may adversely affect a data processing chain. Thus, careful configuration and implementation is paramount to the successful use of the DLR DFD Monitoring and Reporting System.

REFERENCES

[RD-1] The German Remote Sensing Data Center, DFD, https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-5278/8856_read-15911/

[RD-2] Ionosphere Monitoring and Prediction Center, https://impc.dlr.de/

[RD-3] Large-scale research facilities, IMPC, https://www.dlr.de/content/en/research-facilities/ionosphere-monitoring-and-prediction-center-impc-en.html

[RD-4] Martin Kriegel and Jens Berdermann for the IMPC team, (2020), *Ionosphere Monitoring and Prediction Center, DLR, Institute for Solar-Terrestrial Physics Neustrelitz, Germany,* The European Navigation Conference ENC 2020, May 11-14, 2020, Dresden, Germany.

[RD-5] Senft, Johanna und Chereji, Cristian und Mario, Winkler und Molch, Katrin und Eberhard, Mikusch (2015) *The D-SDA Reporting System: Reporting and User Access at DLR-EOC.* Eumetsat PV Conference 2015, 03.11-05.11, Darmstadt, Deutschland.

[RD-6] Barkmann, Henrike und Voinov, Galina und Risch, Daniel und Stönner, Sven und Wegner, Max und Tegler, Mirco (2022) *The Ionosphere Monitoring and Prediction Center (IMPC): infrastructure and software implementation.* European Space Weather Week 2022, 24.-28.10.2022, Zagreb.

[RD-7] Barkmann, Henrike (2019) *Generierung von Reports über den Zustand und die Zuverlässigkeit automatischer Datenprozessierungssysteme von Fernerkundungsdaten.* Masterarbeit, Wilhelm Büchner Hochschule

[RD-8] Kubernetes, https://kubernetes.io/

[RD-9] Argo, https://argoproj.github.io/argo-workflows/

[RD-10] Telegraf Agent: https://www.influxdata.com/time-series-platform/telegraf/

[RD-11] InfluxDB, https://www.influxdata.com/

[RD-12] Flux Query Language: https://www.influxdata.com/products/flux/

[RD-13] Grafana, https://grafana.com/docs/loki/latest/fundamentals/overview/

[RD-14] ReportServer, https://reportserver.net/de/