Enhancements in Functionality of the Interactive Visual Explorer for ATLAS Computing Metadata

M.A.Grigorieva[1,2], A.A.Alekseev[3], A.A.Artamonov[4], T.P.Galkin[4], D.V. Grin[6], T.A.Korchuganova[3], S.V.Padolski[5], M.A.Titov[1], A.A.Klimentov[5] on behalf of the ATLAS Collaboration[1]

[1] Lomonosov Moscow State University, Leninskie Gory, 1, building 4, Moscow, 119234, Russian Federation

[2] Moscow Center of Fundamental and Applied Mathematics, GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation

[3] Ivannikov Institute for System Programming, Alexander Solzhenitsyn st., 25, 109004, Moscow, Russian Federation

[4] National Research Nuclear University "MEPhI", Kashirskoe shosse, 31, 115409, Moscow, Russian Federation

[5] Brookhaven National Laboratory, Upton, New York, United States of America

[6] National Research Center "Kurchatov Institute", Akademika Kurchatova pl., 1, 123182, Moscow, Russian Federation

Abstract. The development of the Interactive Visual Explorer (InVEx), a visual analytics tool for the computing metadata of the ATLAS experiment at LHC, includes research of various approaches for data handling both on server and client sides. InVEx is implemented as a web-based application which aims at the enhancing of analytical and visualization capabilities of the existing monitoring tools and facilitates the process of data analysis with the interactivity and human supervision. The current work is focused on the architecture enhancements of the InVEx application. First, we will describe the user-manageable data preparation stage for cluster analysis. Then, the Level-of-Detail approach for the interactive visual analysis will be presented. It starts with the low detailing, when all data records are grouped (by clustering algorithms or by categories) and aggregated. We provide users with means to look deeply into this data, incrementally increasing the level of detail. Finally, we demonstrate the development of data storage backend for InVEx, which is adapted for the Level-of-Detail method to keep all stages of data derivation sequence.

[1] Copyright 2020 CERN for the benefit of the ATLAS Collaboration. Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.

1 Introduction

InVEx (Interactive Visual Explorer) [1] is a toolkit for visual analysis of multidimensional data. This visualization system was developed to cover the needs of representation of the computing metadata of the ATLAS experiment [2] at Large Hadron Collider (LHC) and can address similar tasks in other domains. The InVEx functionality for data analysis, building interactive 3D models and visualization of the results of clustering stands on machine learning methods. The development of InVEx started with implementation of a 3D interactive tool for cluster analysis. Its further evolution was driven by requirements formulated within the ATLAS computing experts community and aims to ensure stability and efficiency of the distributed computing environment. In the process of the integration of InVEx with ATLAS computing metadata sources we faced two main challenges:

1. big data volumes needed to be analyzed in real time. For example, one ATLAS computing task consists of more than a thousand jobs. Each job is described by two hundred different parameters.
2. unsupervised clustering algorithms are not sufficient for visual cluster analysis. This is why we decided to add the ability of user-defined clustering/grouping (by nominal or ordinal parameters). It makes the process of data analysis more manageable.

This paper demonstrates three main enhancements in the InVEx functionality: user-manageable data preparation, Level-of-Detail, and improving of the data storage format.

2 Data Preparation

Data preparation is the prerequisite step of the real-time data analysis. Data taken from a remote source, i.e., from one of the ATLAS computing monitoring systems - the BigPanDA monitor [3], contains metadata of computing tasks and jobs, filtered by particular conditions, like time range, job status, computing site, etc. Each job may have up to 200 features of various data types. InVEx allows using an arbitrary arbitrary set of features for the data exploration process and therefore may download the entire available set. InVEx GUI provides a special feature navigation panel, called *DatasetInfo*, where data descriptors are grouped by types: numerical continuous, ordinal, nominal, range non-categorical (such as certain string data that can’t be treated as categorical, e.g., error log messages, job names).

- *Continuous* features are represented by the following attributes: feature name, data type, min, max, mean, standard deviation, quantiles and percentage of missing values.
- *Ordinal, nominal, range* and *other non-categorical* features have another set of attributes: feature name, data type, the number of unique values, distributions of all unique values, percentage of missing values.

The *DatasetInfo* panel is used to [de]select features for further analysis (grouping, clustering).

3 Level-of-Detail Method

The data sample for real-time exploration of PanDA [4] computing payload starts from thousands of jobs and may exceed hundreds of thousands, depending on the filtering conditions and selected time range. The implementation of the Level-of-Detail Generator (LoD) [5] was initiated to avoid visualization of such a huge amount of data objects in one scene. Initially, LoD provided clustering of the initial data sample using the

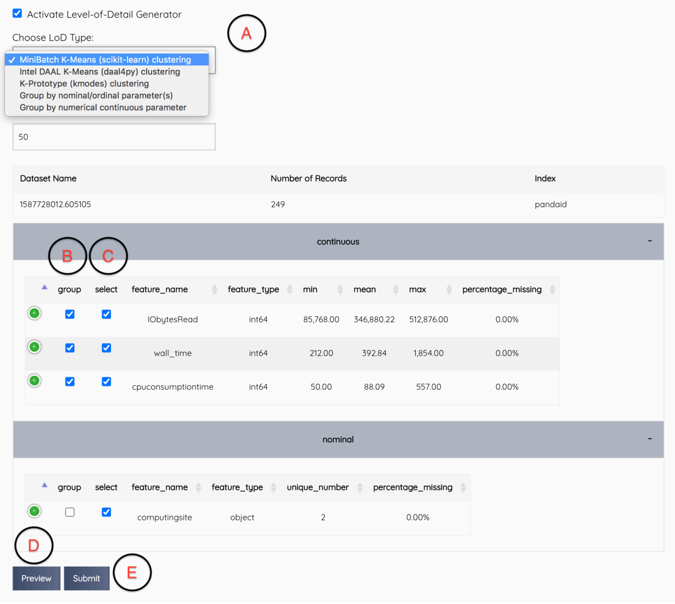
https://doi.org/10.1051/epjconf/202024505032

MiniBatchKMeans[2] algorithm only and returned clusters (i.e., groups) as aggregated data objects. Thus, the analyst could work with the small number of data objects, reflecting the structure of the initial data (moreover, visualization of a reduced number of objects works faster and is perceived easier).
The current InVEx release was enhanced by using the performant implementation of KMeans algorithm from Intel Data Analytics Acceleration Library (DAAL)[3], and also by providing initial grouping of the source large data sample by known features, which, sometimes, is more representative for the analyst. LoD supports the following methods: MiniBatchKMeans Clusterization *(python/sklearn)*, Fast KMeans Clusterization *(python/daal4py)*, K-Prototypes Clusterization *(python/kmodes)*, Grouping by nominal/ordinal parameters *(python/pandas)*, Grouping by numerical continuous parameters *(python/pandas)*.

4 Data Exploration Pipeline

The pipeline of data exploration in the InVEx consists of the following steps:

1) Upload initial data sample from a remote source or the local file system (examples of the figures below show 249 computing jobs)
2) Prepare data for the analysis (*DatasetInfo Panel* (Figure 1)):
    a) choose Level-of-Detail method ***(A)***
    b) select features for clustering/grouping ***(B)***
    c) select features for further data analysis ***(C)***
    d) execute clustering/grouping with LoD ***(D)***
    e) submit the prepared data sample ***(E)***



☑ Activate Level-of-Detail Generator

Choose LoD Type:

✓ MiniBatch K-Means (scikit-learn) clustering
Intel DAAL K-Means (daal4py) clustering
K-Prototype (kmodes) clustering
Group by nominal/ordinal parameter(s)
Group by numerical continuous parameter

50

Dataset Name	Number of Records	Index
1587728012.605105	249	pandaid

continuous

	group	select	feature_name	feature_type	min	mean	max	percentage_missing
⊕	☑	☑	IObytesRead	int64	85,768.00	346,880.22	512,876.00	0.00%
⊕	☑	☑	wall_time	int64	212.00	392.84	1,854.00	0.00%
⊕	☑	☑	cpuconsumptiontime	int64	50.00	88.09	557.00	0.00%

nominal

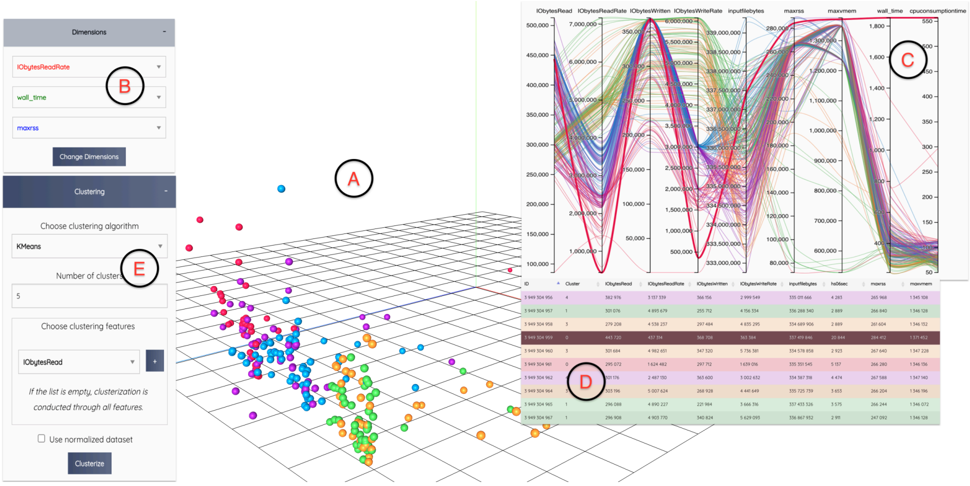
	group	select	feature_name	feature_type	unique_number	percentage_missing
⊕	☐	☑	computingsite	object	2	0.00%

Figure 1. Dataset Info Panel

3) Explore (aggregated) data objects on interactive visual scenes (*Visualization and Parallel Coordinates Panels* (Figure 2))

[2] https://scikit-learn.org/stable/modules/clustering.html#mini-batch-kmeans

[3] Intel Data Analytics Acceleration Library, https://software.intel.com/en-us/daal



Dimensions

IObytesReadRate

wall_time

maxrss

Change Dimensions

Clustering

Choose clustering algorithm

KMeans

Number of clusters

5

Choose clustering features

IObytesRead

If the list is empty, clusterization is conducted through all features.

□ Use normalized dataset

Clusterize

ID	Cluster	IObytesRead	IObytesReadRate	IObytesWritten	IObytesWriteRate	inputfilebytes	hs06sec	maxrss	maxvmem
3 949 304 956	4	382 976	3 137 339	366 156	2 999 549	335 011 666	4 283	265 968	1 345 108
3 949 304 957	1	301 076	4 893 679	255 712	4 156 334	336 288 340	2 889	266 840	1 346 128
3 949 304 958	3	279 208	4 538 237	297 484	4 835 295	334 689 906	2 889	261 604	1 346 132
3 949 304 959	0	443 720	437 314	368 708	363 384	337 419 846	20 844	284 412	1 371 452
3 949 304 960	3	301 684	4 982 651	347 320	5 736 381	334 578 858	2 923	267 640	1 347 228
3 949 304 961	0	295 072	1 624 482	297 712	1 639 016	335 351 545	5 137	266 280	1 346 136
3 949 304 962	4	301 176	2 487 130	363 600	3 002 632	334 387 318	4 474	267 588	1 347 140
3 949 304 964	3	303 196	5 007 624	268 928	4 441 649	335 725 739	3 653	266 204	1 346 196
3 949 304 965	1	296 088	4 890 227	221 984	3 666 316	337 433 326	3 575	266 244	1 346 072
3 949 304 967	1	296 908	4 903 770	340 824	5 629 093	336 867 932	2 911	247 092	1 346 128

Figure 2. 3D visualization and parallel coordinates with linked table

a) initial exploration of data objects on the 3D scene ***(A)*** with the ability to select the dimensions interactively ***(B)***
   - o features shown on the Figure 2(B): *IObytesReadRate* (rate of the input data processing), *wall_time* (timing metrics of the process), *maxrss* (fraction of shared memory used by the CPU)

b) observation of features trends on the parallel coordinates diagram ***(C)***. Interactivity allows to select objects on the diagram, and explore them in the linked data table ***(D)***
   - o features shown on the parallel coordinates (Figure 2(C)) are: *IObytesRead, IObytesReadRate, IObytesWritten, IObytesWriteRate* (metrics for the input/output data), *inputfilebytes* (size of input dataset), *maxrss* (fraction of shared memory used by the CPU), *maxvmem* (fraction of virtual memory), *wall_time* (timing metrics of the process), *cpuconsumptiontime* (CPU consumption time)

c) choose available clustering methods for the visualized (aggregated) data and define its settings at the *Clusterization Panel* ***(E)*** (this stage could be repeated many times with different algorithms and settings).
   - o The following algorithms are currently available: *KMeans/MiniBatchKMeans, DBSCAN, K-Prototypes* (clustering of mixed data – numerical and categorical features), *Hierarchical* and *GroupData* (group data by selected categorical features)

d) explore results of clusterization on the interactive visual scenes (as in stages *a* and *b*)

e) choose an object of interest and click on it to visualize in a separate window. This option is available only if this object represents a group, generated at the LoD stage (*2, d*)

f) in the new window all stages of the data exploration pipeline, beginning from the stage *2* are repeated.

5 Data Storage Format

The storage backend of InVEx serves to store the whole analysing data sample consisting of every stage of hierarchical data in the derivation sequence, which starts with the initial data sample as raw input data. The reason of the persistent storing of the entire input data consists in the peculiarity of the exploratory data analysis: it implies that user doesn't know in advance the structure of data and might change selection criteria many times. Storing the entire raw data at the beginning obviate the need to download the data again and again. Together with data storage, InVEx stores also the operations performed against it such as clustering, grouping and LoD changes.

Different storage implementations were considered to fulfill the following requirements: the sufficient size of the input data sample (up to millions of records) and the arbitrary number of data analysis pipelines that also should be stored. We considered various storage backends, such as file storage, relational database and non-relational database. In the case of file storage, the data sample itself and all stages of the processing pipeline can be stored in a single file or in a folder with a number of files. Such storage is scalable, but navigation might become complex with increasing size (with a plain text format, e.g., JSON). Relational database is not sufficient for storage of data samples containing variable schema. Non-relational database is not reviewed as InVEx doesn't assume the common storage for all sessions. As we deal with big data samples, each analytics session must be encapsulated as much as possible. Thus, we choose HDF5 file storage format as one of the most suitable for InVEx.

The HDF5 data model allows to store data sample and metadata in one file. HDF5 is a compressed format that is designed to support large, heterogeneous, and complex datasets. An HDF5 file (an object in itself) can be represented as a container (or group) that holds a variety of heterogeneous data objects or datasets, where objects are images, tables, graphs, or even documents, such as PDF or Excel. Two primary objects in the HDF5 Data Model are groups and datasets. Groups include other group objects (sub-groups) or datasets, thus are introduced as organizational units. All objects are self-describing, since every object has assigned attributes (as one of HDF5's advantages).

The structure of the InVEx storage backend based on HDF5 is shown in Figure 3.

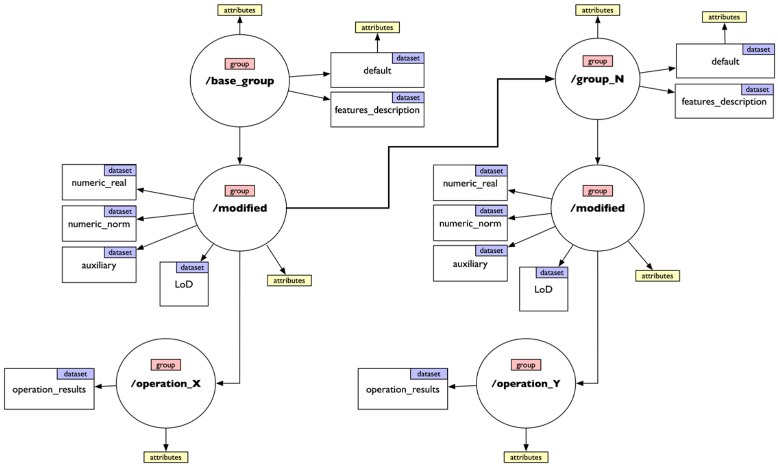


Figure 3. HDF5 file structure in InVEx

"Base_group" is a root object with datasets describing the input data sample ("default" dataset includes raw input data as a NumPy array). "Modified" sub-group object includes modified data samples: data with numeric values only and its normalized edition, and data with non-numeric values. Every "modified" group has a set of sub-groups (no restriction on its number) that represent applied operations with corresponding list of resulted clustering labels ("operation_X"). With LoD turned on there is a dataset "LoD" that contains information about grouping of the initial records, which is used to create sub-groups "group_N" (from "modified" group) dynamically from the initial or previously grouped data sample. Such sub-groups are created on user demand for analysis of a particular group of records.

Conclusion

The InVEx toolkit can be compared with visual analytics tools such as Clustervision [6], Clustrophile [7], XCluSim [8], however these tools don't provide 3D visualization, that may be useful for the exploration of spatial objects, and flexible Level-of-Detail mechanism, that is crucial for the analysis of big volumes of data. It should be noted, that the Level-of-Detail component in InVEx, in particular, uses Intel DAAL library, providing significant increase in the performance of the K-means clustering, in contrast with python/sklearn library.

InVEx was developed and tested on the computing metadata of the ATLAS experiment at LHC and it is applicable to the exploration and monitoring of various data.

The next stage of the development is the addition of supercomputing resources for data preparation and processing for visual analysis.

Acknowledgements

This work was financially supported by the Russian Science Foundation (grant No.18-71-10003).

References

1. T. Galkin et al., *An Application of Visual Analytics Methods to Cluster and Categorize Data Processing Jobs in High Energy and Nuclear Physics Experiments*. Scientific Visualization vol.10, no.5 (2018) pp.32-44
2. The ATLAS Collaboration, *The ATLAS experiment at the CERN Large Hadron Collider*. JINST vol.3 (2008) S08003
3. A. Alekseev et al., *ATLAS BigPanDA monitoring*. J. Phys.: Conf. Ser. vol.1085 (2018) 032043
4. P. Svirin et al., *BigPanDA: PanDA Workload Management System and its Applications beyond ATLAS*. EPJ Web of Conferences vol.214 (2019) 03050
5. M. Grigorieva et al., *Evaluation of the Level-of-Detail Generator for Visual Analysis of the ATLAS Computing Metadata*. Lobachevskii Journal of Mathematics vol.40, no.11 (2019) pp.1788-1798
6. B. C. Kwon et al., *Clustervision: Visual Supervision of Unsupervised Clustering*. IEEE Transactions on Visualization and Computer Graphics, Volume: 24, Issue: 1, 2018, pp. 142-151, doi: 10.1109/TVCG.2017.2745085
7. M. Cavallo, Ç. Demiral, *Clustrophile 2: Guided Visual Clustering Analysis*. IEEE Transactions on Visualization and Computer Graphics, Volume 25, Issue 1, pp. 267-276, doi: 10.1109/TVCG.2018.2864477
8. S. L'Yi et al., *XCluSim: a visual analytics tool for interactively comparing multiple clustering results of bioinformatics data*. BMC Bioinformatics, Volume 16, supplement 11, Article number: S5 (2015), doi: 10.1186/1471-2105-16-S11-S5