Physical Analytics Integrated Repository and Services for Astronomy: PAIRS-A

Project consideration area: Scalable Storage and Analytics Solution for Combined Astronomical Data

Authors: Bruce Elmegreen, Conrad Albrecht, Hendrik Hamann, Levente Klein, Siyuan Lu, Johannes Schmude

Contact information:

Dr. Bruce Elmegreen
IBM T.J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
bge@us.ibm.com (914 945-2448)

Table of Contents

1. Key Science Goals and Objectives

There is a data explosion in Astronomy as new instruments have increasing ability to cover the whole sky at high resolution in a broad swath of wavelengths, from radio (SKA, NGVLA) to optical and infrared (SDSS, GAIA, LSST, Euclid, WFIRST), while including the time dimension (Pan-STARRS, ZTF, LSST) as well as gravity (LIGO, VIRGO, LIGO-India) and high energy events like cosmic rays and gamma rays (HESS, CTA). Each new observatory is challenged to devise a data plan specific to their instruments and the needs of their users. For future data scales measured in 100s of Petabytes, these plans need the most efficient strategies for data assimilation, curation, and analysis. As a result, much activity is currently underway to optimize these capabilities, e.g. [1,5,8,10,11,14,17,19].

This APC Whitepaper describes a new strategy for curation and analysis of position-time data that scales to hundreds of Petabytes, allowing a large fraction of astronomical observations to be analyzed "close to the data" to facilitate discovery and understanding. This new strategy is not meant to replace any data center currently planned for observatory or national use, but to collect all astronomical data in one distributed system for inter-comparisons and complex queries. Multi-layer searching like this typically increases layer value by revealing new correlations and possible causations. With billions of dollars in observatory investments worldwide, all efforts should be made to get the most science from the data.

Data volumes have greatly outpaced the speed of disk retrieval. For example, the daily Earth-image volume of 10 TB from the European Space Agency would take a whole day to move from disk to memory at a standard disk read speed of 100 MB/s. Thus, the benefits of using large data reserves for machine learning or multi-variable comparisons can only be realized if the processing happens close to the data. This means highly parallel systems without files to fetch, open and read locally. Data immobility also means that useful archives should co-exist because they cannot be compared using separate sites. Thus, Big Data has *gravity*: it tends to attract other data.

Traditional business-related database applications (e.g., banks) and platforms for image storage and analysis, including Geographic Information Systems used by governments, and the popular SkyServer for astronomy (http://www.sciserver.org), use a relational database management service (RDBMS) and consist of files (tables) that are linked by keys and addressed by Structured Query Language (SQL). This works well for structured data and modest data volumes (SkyServer has about 2 PB) but RDBMS typically do not scale out to thousands of processors and hundreds of Petabytes, nor do they readily include unstructured data (https://www.jamesserra.com/archive/2015/08/relational-databases-vs-non-relational-databases/). On the other hand, No-SQL databases do not use files and may have a variety of forms; they are generally simpler, more scalable, and more versatile in terms of data format. As data volumes reach multi-Petabytes and data types for inter-comparisons and discovery become more numerous and diverse, No-SQL databases are receiving growing interest and adoption. Here we describe an existing No-SQL database and analytics platform based on key-value stores. We show how a system like this would be well suited for astronomical data and analytics in the next decade.

2. Current Status: PAIRS Geoscope

The proposed system is based on the existing IBM PAIRS Geoscope[1], a multi-Petabyte database and analytics platform that achieves highly scalable use with Earth-based images (raster data) and vector data[2] designed for geo-temporal information [12,15]. PAIRS Geoscope currently curates around 10 TB/day of satellite imagery that is useful for crop predictions and forecasts, vegetation and disaster management, urban planning, and cross-layer analytics that may also include drone images, sensor data, and geo-located social media data such as twitter commentary. Today, it has thousands of layers of distinct data, including census information at high resolution as well as street maps. The coordinate grid is Earth-based longitude and latitude (EPSG:4326), the same as that used for GPS.

PAIRS Geoscope can rapidly retrieve complex queries like *"Give me the average precipitation in the 2017 growing season on all planted corn fields in the Contiguous US where the Normalized Difference Vegetation Index (NDVI) in the first 15 days of June was larger than 0.5."* This query involves 216 data layers, 22 TB of processed data (temporarily matched to the highest resolution data set) and three different data sets: MODIS (Satellite), Precipitation (weather service), and USDA Cropscope (government). It recently took 693 seconds, which is 31.7 GB/s of data analysis or 2.7 PB/day equivalent. On the current platform, five such queries can be run in parallel without performance degradation, amounting to 13.7 PB/day (the current platform is 3.4 PB in commodity disk drives on ~65 data nodes with 16 TB memory). Figure 1 shows a snapshot of the results with an enlargeable map of the selected corn fields and a time series of the requested precipitation. Information like this is useful for commodities traders and agriculture insurance companies.

An analogous query in astronomy might be: *"Find all objects with galaxy-type colors that have had a radio band intensity vary by more than 5-sigma within the last year and which also appear in the search zone of a gravity wave detected within the last hour, and superpose the X-ray and cm-wave radio emissions from these galaxies on their optical images."* Such queries enable multilayer data discovery of heterogeneous physical parameters of astronomical objects. Because the data are indexed and the operation is parallelized, as described next, these queries can run quickly, allowing rapid iteration over search hypotheses.

Fig 1: Example of data retrieval and spatio-temporal browsing using today's PAIRS Geoscope graphical web user interface. For details and demos, visit: https://ibmpairs.mybluemix.net/ , for technical tutorial: https://pairs.res.ibm.com/tutorial/

[1] E.g., https://www.ibm.com/us-en/marketplace/geospatial-big-data-analytics
[2] referring to geo-temporally referenced location, line, and (multi-)polygon data

3. Technical Overview: PAIRS-Astroscope or "*PAIRS-A*"

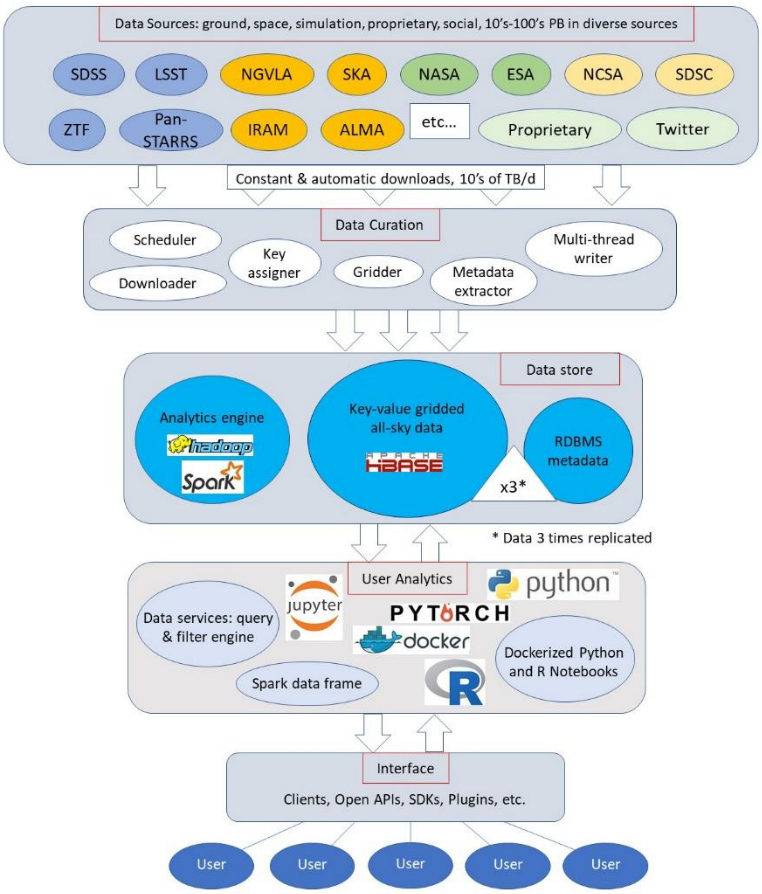


Fig 2: Schematic of PAIRS-A with data input at the top and user input at the bottom.

The proposed data and analytics platform for astronomical use could be structured in essentially the same way as PAIRS Geoscope, but with a different coordinate grid (see below) and much higher capacity in the future. Figure 2 shows a schematic. Data from various sources shown at the top of the figure would be automatically ingested to PAIRS-A using a scheduler with data validation (second panel). The Metadata would be extracted and stored in a Relational Database Management System (third panel) for later searching. The data would be placed on a fixed coordinate grid of sky position and time using key-value pairs (see below) and written to storage in multiplicate for fault tolerance (third panel). Close to the data (third panel), analytics engines would access and process the data according to user instructions shown by example in the fourth panel. An interface consisting of APIs, SDKs such as the open-source Python module https://github.com/ibm/ibmpairs and other tools (fifth panel) would be accessed by multiple users, who each have their own private work space and access to certain data available to them. Users would also be able to upload other data and analytics software for their personal use. The system could be available in the cloud and physically located at multiple sites for fault tolerance, including NSF facilities, and managed by, e.g., NOAO in Tucson [18].

4. A Design Enabler: A Universal Coordinate System in the Sky

The universality of sky coordinates allows all data pixels regardless of the size and orientation of their associated arrays to be individually placed on a universal sky grid. The grid should be optimized for resolution using hierarchically nested layers with exponentially decreasing pixel size, i.e., a quad-tree (Fig 3, e.g., [20]). Array data, which are pixel-based maps, and vector data, which are strings such as time-series at a single point or the spectrum of a star, are both indexed by the grid.

A sky grid that is well suited for astronomy is HEALPix [7], which has been used for the cosmic microwave background (e.g., [13]) because of its computational advantage in calculating Spherical Harmonics (Fig 4). HEALPix has equal-area pixels aligned to circles of constant latitude. The grid can be addressed sequentially in binary with a space-filling curve such as a Z-index (Fig 3), which allows data nearby in the sky to be nearby in storage for fast retrieval of contiguous regions and multi-layer queries. Other indexing schemes have various advantages [16].

5. Technical Details: Key-Value Storage with HBase on Hadoop

PAIRS Geoscope has been developed using the open source big data technologies Hadoop and HBase[3] [6,21], GeoMesa [9], and Spark [22]. Apache Hadoop distributes data across nodes in a computer cluster and transfers code into these nodes to process data locally. Apache HBase is a database management system that runs on Hadoop and consists of values identified by row, column, and qualifier keys. HBase is not a relational database and does not support a structured query language like SQL (but see Sect. 6.3). Hadoop automatically handles hardware errors. GeoMesa is able to use HBase as its backend to scalably store and process vector data such as star locations, or bounding box polygons for uploaded images. Apache Spark is an in-memory framework that allows for distributed, scalable analytics loading data from HBase. The storage system uses parallel architecture with HBase on Hadoop and allows the use of the Apache Map/Reduce function in which queries or analytical tasks are done in parallel on separate nodes (mapped) and then merged into single or Master nodes (reduced). Map/Reduce gives highly scalable performance [2,4]. Unlike relational databases, key-value stores (details below) are scalable to hundreds of Petabytes [3].

The organization of data is by "key-value" combinations. All data values are given keys during ingestion into PAIRS. Normally there are three keys in an HBase table, which may be thought of as row, column and qualifier. In PAIRS Geoscope, the row key is a binary number that consists of spatial and time parts corresponding to the value; this locates the data in storage. The column key is used to identify the type of data, such as temperature or humidity for PAIRS Geoscope, while

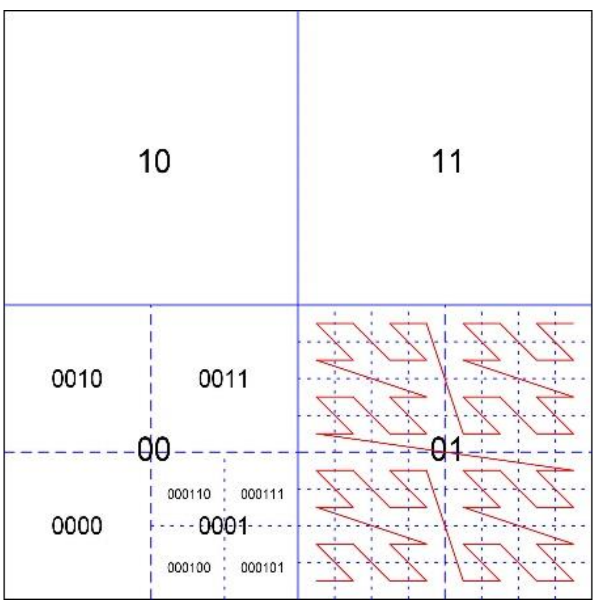


Fig 3: Quad-tree design with Z-order pixels.

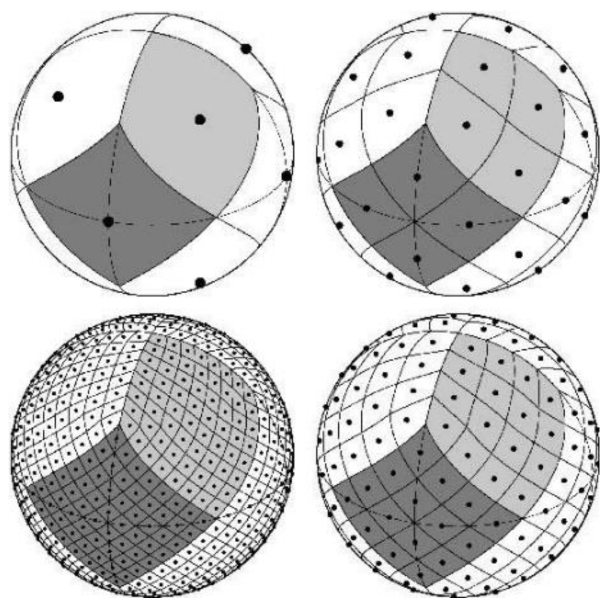


Fig 4: HEALPix coordinate space [7].

[3] Hadoop, Spark and HBase are registered trademarks of the Apache Software Foundation. GeoMesa is copyright by Commonwealth Computer Research, Inc. and licensed by Apache.

the qualifier key is used to give a third quantity, such as the value of the atmospheric pressure (i.e. elevation) for that temperature or humidity. The value associated with the three keys is the temperature or humidity itself, and is packed, for optimum storage, in a block of 32x32 other such values that come from nearby pixels in an image. This blocking makes the storage required for the keys (16 bytes in PAIRS Geoscope) much smaller than the storage required for the data, saving storage overall.

Blocking also significantly improves read/write speed in PAIRS Geoscope. The keys assigned at ingest contain the position coordinates of the lower left corners of the 32x32 pixel arrays ("supercells"), rather than the coordinates of all individual pixels. Reading and writing each key-value pair then processes 1024 pixels at once, enhancing performance by a factor of about 50 compared to single-pixel keys in benchmark tests. The "value" assigned to each supercell "key" is an array of 1024 variables representing the "pixel's content". Today these are typically integer or floating-point numbers of camera CCD readings, but nothing prevents the design from having more complicated values, e.g. tensors. Larger areas could be used for supercells too, for increased benefits in storage and speed. If acceptable, (lossy) compressions such as JPEG could be applied to the supercell's image data. However, HBase itself can intrinsically apply *lossless* compression by about a factor of 3.

In PAIRS-A, an image of the sky might have each pixel's position and time of observation encoded in the first key, the passband as the layer for the data in the second key, nothing in the third key (i.e. some default value), and the intensity as the value, blocked together with other intensity values from the same image, as above. The header of the original FITS file could be found for each pixel if desired by querying position and time in a much smaller relational database (see below). Similarly, the spectrum of an object might have the object's position in the sky and time of observation encoded in the first key, the wavelength in the second key, a default placeholder value in the third key, and the intensity at that wavelength for the value. Integral Field Unit spectral data would be well suited to this organization with neighboring pixels blocked together for efficient storage. The time series of radio emission from a pulsar that varies more rapidly than the first key varies with time might have the position and start time of the observation in the first key (to locate the record), the radio frequency as the layer in the second key, a placeholder in the third key, and the entire time sequence blocked together in storage as a list of values. In general, the value of the HBase (key1, key2, key3, value)-tuple might host any kind of spatio-temporally tagged astronomical data structure. The keys and values can be any sequence of bytes[4]. It is the query engine's responsibility to correctly interpret it using the Metadata (FITS header) or other information for data fusion. The format is fixed at ingestion by the PAIRS upload and curation module.

Some astronomical events are more complicated than images, spectra, IFU spectra and rapid time-sequences, but they can still be placed in the database and analyzed in the same fashion as, and jointly with, the other data. A sub-second gamma ray event might be detected with the Cherenkov Telescope Array as a rapid time sequence of light spread across the sky, with a different pattern for each telescope according to the 3D structure of the cascade trail. In such a case, PAIRS-A

[4] Technically, the HBase column family (key2) is limited to printable characters. Moreover, its value needs to be fixed on HBase table creation.

might have the initial sky position and approximate observing time in the first key to identify the event, spectral passband in the second key, a code for the Earth location of the particular detector in the third key (giving a useful data layer for all events detected by that detector), and a time-sequence of sky-mapped intensities as the block of values.

Every observation has Metadata involving facility, the coordinates and times of observation, calibration, and so on, as usually listed in a FITS header but not limited to that. Often the same Metadata applies to all pixels in an image, which could be in the millions, so Metadata should be written only once (in triplicate for fault tolerance) for all associated pixels. This can be accomplished using a relational database management service with SQL queries that input the spatial position of a pixel or block of pixels, and, if also desired, a time range for the observations, and which returns links to all the associated Metadata or the Metadata itself. In this way the exact FITS header and any other relevant information can be found with the appropriate SQL for each stored value in PAIRS-A. Data pixels would never be completely detached from their associated Metadata, although each would be stored separately for optimum efficiency.

By design, the current PAIRS Geoscope HBase key is 16 bytes long comprising a spatial and temporal part plus a hash for the randomization of storage positions. Recall that the cells in a quad tree require 2 bits for each level (Fig 3). PAIRS Geoscope has 24 spatial levels (using 48 bits in the key) down to the 32x32 supercell and then 5 more levels in each supercell down to individual pixels. The cell size is $2^{(29-L)} x10^{-6}$ deg with L running from 6 for the largest cells (8.4 deg) to 24 for the 32x32 blocks of pixels. The smallest pixels are 10^{-6} deg = 0.0036" ~ 0.11m on the equator. Time is represented by the 64 lowest significant bits of the key, counting seconds in a symmetric interval around 1st January 1970 to the future and past. 12 bits are used as a hash (8 for space and 4 for time) in the high significant bit positions to avoid local hotspots when reading and writing neighboring pixels to the same HBase server; 4 bits are held in reserve.

For astronomical use, the angular grid might range from $1/12^{th}$ of the sphere (58.6^o) in the highest level of HEALPix, to 0.8 microarcsec, which is smaller by $1/2^{38}$, i.e., 38 levels down. This would take 80 bits of the key if the first level has 4 pixels to represent the 12 largest regions. The time of observation could, for example, cover 70 years in units of 2 milliseconds, which would take 40 bits to represent. The remaining 8 bits could be for the hash. There is no restriction on the bit length of the keys, so greater resolution or a bigger hash are possible with larger keys.

The keys would be assigned at the time the data is downloaded using the sky coordinates for each pixel for the spatial bits and the time of observation for the time bits, both obtained from the Metadata (e.g., the header in the FITS file or some other source). Pixels in the detector get translated into pixels in sky coordinates (e.g., HEALPix) via Nyquist sampling with interpolation. Note that the data are not propagated to all resolutions but stay close to their native resolution. If later blended with higher resolution data, each pixel can be subdivided on-the-fly during the blending. Similarly, high-resolution data can be blended later with low resolution data by averaging pixels together.

One might consider including raw data as well as reduced and science-ready data, but it is envisioned in this proposal that the raw data will be available in the observatory data repositories and only the highest-quality science-ready data will be systematically archived by PAIRS-A for general community use and discovery. Of course, a user may upload raw data to their own file system on PAIRS-A, for password-controlled reduction or assimilation with other data. The user could also upload their finished product into PAIRS-A for general use, with a tag identifying who made it and at what time. Such uploads might be a useful supplement to publication, in which case they could be provided as links to PAIRS-A and be unchangeable thereafter. PAIRS-A could satisfy the increasing mandate by funding agencies to make Principle Investigator data from public facilities secure, accessible, and long-lasting.

Key-value structure is not only good for high-speed parallel control, but it also conserves storage by eliminating empty space in files: only data with meaningful key-value pairs are loaded into the system. Also, with no relational database or separate files from discrete images or vectors, the user will never need to fetch, align, interpolate and join separate regions into single mosaics. Individual pixels do not have an intrinsic orientation and do not need to be stored in the same files as sky-adjacent pixels received at the same time. Rather, pixels and their times of observation can be stored independently using position-time keys to indicate the storage locations. Finding a time series in a sky region or at a point is relatively simple as different times are stored with the same spatial key and located close to each other in storage. Querying and combining many different layers of space and time, e.g., representing different wavelengths or multi-messenger sources, is also relatively easy as the layers in the same direction have the same spatial keys.

6. Examples of Analytics

6.1 Basic Analytics in PAIRS

The simplest query in PAIRS Geoscope is for a time series at a single position, as that retrieves all data with the same 48-bit position key, resulting in an efficient[5] HBase scan. Range queries based on polygon outlines at single times, i.e., maps, are simple and fast too, as are map queries in multiple layers[6]. Recall that these data are stored for PAIRS Geoscope in 32x32 pixel units for rapid block retrieval and contiguous storage locations. Similar positions are stored in similar locations, although spread out somewhat among processors using the prefix hash in the key to prevent hotspots.

Multilayer queries can involve different types of data at different angular resolutions with interpolation and averaging to match pixels done on the fly. The quad-tree labeling system is useful for this. Consider one layer, such as an LSST field, with 0.7 arcsec native resolution and 0.2 arcsec pixels. This might be stored in level 21. Recall that level 0 is the 12-pixel sky at lowest resolution in HEALPix, where each pixel has a size equal to $(4\pi/12)^{0.5}*(180^o/\pi)=58.6^o$. The level number for 0.1 arcsec resolution, the Nyquist sample for 0.2 arcsec native pixels, is found from the formula 58.6^o x (3600 arcsec/deg)/2^n=0.1 arcsec, giving the result n=21. Level 0 needs 4 bits to identify

---

[5] Today PAIRS returns data for e.g. 14K timestamps at a given location for two physical quantities (28K data points) in about a second. When working with files, one would need to open the raster files individually for each timestamp of the scene containing the location and extract a single pixel. Note: The test was performed for the full history of PRISM temperature and precipitation data.

[6] For example, generating a raster image for the mean temperature in Iowa at 250m resolution over 4 months takes below a minute with PAIRS. In another scenario, it takes PAIRS about 3 minutes to filter out all the corn fields in Iowa.

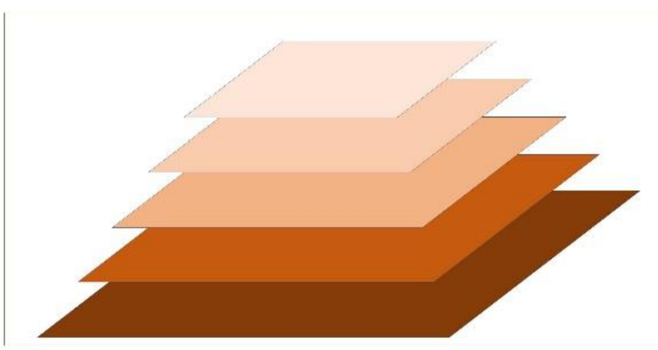
which of the low-resolution pixels is of interest, and level 21 needs 42 more bits, so the LSST positions would be identified by the most significant 46 bits in the position part of the key. Suppose now that this LSST field is to be combined with a radio observation at 20 arcsec resolution. That data might reside at HEALPix level 15 where the pixel size is 6.4 arcsec. Level 15 is represented by the 34 most significant bits in the position key. Thus, the radio data is easily identified in the radio layer at the sky positions given by the 34 most significant bits in the LSST keys. These 34 significant position bits for both LSST and radio will be stored close to each other, making comparisons fast. The additional 46-34=12 bits for LSST identify the higher resolution pixels in the same field of view, $4^{(21-15)} = 64^2$ of them, at each radio pixel. Other data could be combined with the LSST and radio data in the same way, producing overlays, maps and object lists as desired.

PAIRS Geoscope performs queries using a RESTful (Representational State Transfer) API (e.g., using http) that is agnostic to programming language. It allows one to define virtual layers that are computed and combined on-the-fly during query time. Moreover, an open-source wrapper of that API has been published (https://github.com/IBM/ibmpairs)[7] in order to load query results into Python data structures at the user's end for further consumption and local processing.

6.2. Accelerated Data Fusion by Overview Layers

Due to the unified spatio-temporal indexing, a major benefit of PAIRS Geoscope is its ability to fuse previously unrelated datasets for analytics to generate new value. As demonstrated above and shown in Fig. 1, information from surveys (crop type), satellite (NDVI) and weather models can be combined to determine derived layers. In addition, the weather data are aggregated in time on-the-fly before being combined with the crop type and biomass index NDVI.

In order to filter efficiently raster layers at different spatial resolutions, a pyramid with coarse-grained pixels commensurate with the PAIRS grid can be recursively built and stored at the time of data curation by combining 4 (or a power of 4) pixels into one new pixel, given certain aggregation methods. A cartoon of the process is depicted in Fig. 5 with the bottom layer being



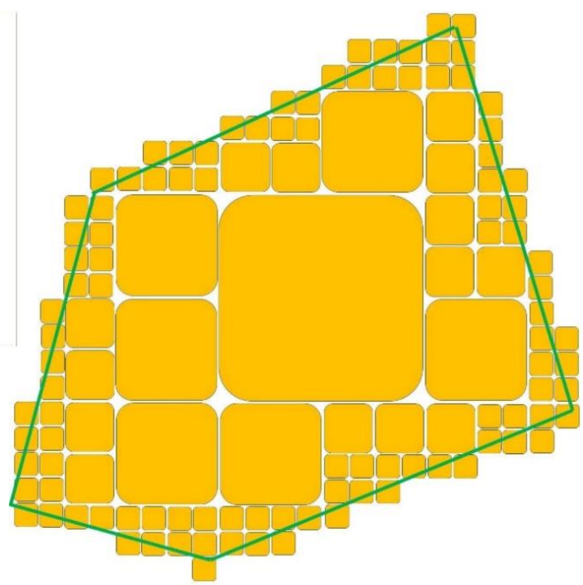


Fig 5 (top): Coarse-grain overview layers.

Fig 6 (right): Quad-Tree decomposition of an area for efficient filtering with overview layers.

[7] installation via e.g. pip install –user git+https://github.com/ibm/ibmpairs

the original data ingested and the overview layers on top, hosting e.g. the minimum or some other statistical function of (power of) 4 neighboring pixels from the layer below. Such data preparation allows rapid cross-filtering of layers at different resolutions. For example, if one is interested in areas of layer A where the values in another layer B are larger than a given value X (e.g., areas in an optical field where radio emission at a certain wavelength exceeds a given value), then a series of overview layers consisting of maximum values in B can exclude big patches where B<X. Thus the high-resolution data do not need to be retrieved in full, but only the relevant parts of the overview layer and the associated parts of the high-resolution layer that satisfy the condition.

In a scenario where one wants to spatially aggregate all pixels in a region to compute some statistic, overview layers can shortcut explicitly retrieving each pixel. According to Fig. 6, overview layer statistics of various types can be generated at the curation stage and then retrieved efficiently by decomposing the region (green polygon) into the largest possible overview cells (yellow squares). Only the region boundary needs to be represented by pixels at the original high resolution.

6.3 Advanced Analytics in PAIRS

Although PAIRS Geoscope uses HBase which does not allow SQL for queries, a capability to use SQL anyway has been built in for convenience. To do this, e.g., results from a PAIRS query can be directly exported into a Spark DataFrame, which is similar to a distributed relational database table. The Spark SQL functionality then allows the use of well-known SQL expressions and user-defined functions on the DataFrame. There can also be multiple result tables with, e.g., different resolutions, which can be joined using SQL commands and the spatio-temporal indexing.

The Spark DataFrame is distributed throughout the memory of the PAIRS computer cluster. Using PySpark, the DataFrame may be accessed via a (local) Python Jupyter notebook. The link between the user’s local machine and PAIRS is established through a RESTful API employing an Apache Livy server (https://livy.apache.org/). No downloading of data to the local machine is required as the notebook instructions operate on the data where they reside in the database.

Efficient data filtering and selection combined with the ability to work with and visualize large volumes of data in a familiar setting will allow for the rapid development of new methods for discovery and insight. Such capabilities will lead to features in PAIRS-A and to methods of advanced analysis that have not been possible before.

7. Cost Estimate and Timescale for PAIRS-A

PAIRS-A can be built with commodity hardware because speed is achieved by scale-out and fault tolerance is achieved by redundancy. Currently, PAIRS Geoscope is a ~3.4 PB system with Hadoop/HBase/Spark on ~65 nodes, each of which has at most 10x10 TB hard disks, 16 CPUs (hyper threading to 32 cores), and 500 GB maximum RAM. Each datacenter rack contains up to 24 data nodes with 10 GB/s network interconnect switches, while the racks are connected with an aggregator switch. The data curation and upload nodes, running collectively at ~10 TB/day, have an additional 30% of CPUs and 25% of RAM for a total of ~85 nodes in the system. No additional resources are needed because the analytics are done on the 65 data nodes, some directly on the HBase tables and some loaded from HBase into Spark DataFrames.

With mid-decade hardware at about twice the density of today's hardware, a 100 PB system of user-ready data would occupy ~27 datacenter racks at full capacity, contain ~650 nodes[9] and be able to curate ~75 TB/day (scaling with node count) at current networking speeds. Considering a likely ramp up to 50 TB/day with an average curation speed of 25 TB/day, a 100 PB system will cover a ten-year period. Such a system would allow the assimilation and inter-comparison of the most important astronomical data, including radio and optical/IR surveys from ground and space. Larger systems are possible with our design.

A 100 PB system for mid-decade, with half the cost per PB of an equivalent system today, including regular hardware replacement, maintenance and facilities expenses, might cost around $6 million per year, placing it in the mid-size ground-based category ($20M-$70M) over a decade. An exact price is not available at this time. Additional costs will depend on the additional features and software developments that are desired.

8. Partnerships

This APC Whitepaper proposes a highly scalable computer system for curation, storage, and advanced analytics on extremely large volumes of astronomical data, including, if desired, most of the reduced and calibrated data from the telescopes of the future. It is based on an existing system in place at IBM Research that is used academically and commercially for Earth-based data via remote log-in ("in the cloud"). For astronomical use, it could be structured and accessed in a similar way with facilities located in a public cloud or an NSF or NASA center, and managed by the proposed data facility at NOAO in Tucson, AZ. Use could be regulated by password; access by individuals to particular data sets could be regulated in a transparent way to be consistent with contractual obligations and other arrangements. The system could be built and serviced by facility staff or a vendor of choice, with provisions for the Intellectual Property already developed and for future IP that will be invented by anyone involved. It is our opinion that a scalable, efficient and user-friendly system like PAIRS will be necessary to get the most science out of the enormous volume of data that the next decade of instruments will generate for the astronomical community.

9. Summary

We propose a curation, storage and analytics platform for discovery and collaboration that can incorporate nearly all science-ready astronomical data and deliver the following unique features to the researcher: (1) high scalability to accommodate the next decade of calibrated data spanning the dimensions of sky position, wavelength, time and multi-messenger source; (2) ability to compare, combine, filter, sort and display multiple data sets simultaneously for discovery and big-data analyses like machine learning and neural network classification; (3) highly efficient retrieval and use of raster (map), vector, and time-series data in multiple layers with key-value storage; (4) familiar user interfaces involving modern tools and software in a combination of private, collaborative and public work spaces; (5) a fast and highly efficient computing system with computation close to the data to minimize data movement, and (6) a cost envelope in the range of mid-size, ground-based projects.

[9] 650 nodes = [100PB total/100TB per node currently]/[2x density]*[85 total nodes/65 data nodes]); 27 racks = 650 nodes/24 nodes per rack.

REFERENCES

[1] Becla, J., Wang, D., Monkewitz, S. et al. 2017 "Large Synoptic Survey Telescope (LSST) Data Management Database Design," LDM-135, https://github.com/lsst/LDM-135

[2] Dean, J. & Ghemawat, S. 2008, "MapReduce: simplified data processing on large clusters," Communications of the ACM, 51, (1), p. 107

[3] DeCandia, G., Hastorun, D., Jampani, M., et al. 2007, "Dynamo: amazon's highly available key-value store," ACM SIGOPS operating systems review, 41, (6), p. 205

[4] Ekanayake, J., Pallickara, S., & Fox, G. 2008, "MapReduce for data intensive scientific analyses," IEEE Fourth International Conference on eScience, p. 277

[5] Flewelling, H.A., Magnier, E.A., Chambers, K.C. 2019, "The Pan-STARRS1 Database and Data Products," arXiv:1612.05243v3

[6] George, L. 2011, "HBase: The Definitive Guide," O'Reilly Media, 1st ed.

[7] Górski, K.M., Hivon, E., Banday, A.J. et al. 2005, "HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere," ApJ, 622, 759

[8] Heap, S.R., Szalay, A.S. et al. 2016, "The WFIRST Science Archive and Analysis Center," in Astroinformatics, IAUS 325, eds. M. Brescia, S.G. Djorgovski, E. Feigelson, G. Longo & S. Cavuoti, Cambridge, University Press, p. 373

[9] Hughes, J.N, Annex, A., Eichelberger, C.N., Fox, A., Hulbert, A., Ronquest, M. 2015, "GeoMesa: a distributed architecture for spatio-temporal fusion," in Proceedings Volume 9473, Geospatial Informatics, Fusion, and Motion Video Analytics V, p. 94730F

[10] Ivezić, Ž., Kahn, S.M., Tyson, J.A. 2019, "LSST: From Science Drivers to Reference Design and Anticipated Data Products," ApJ, 873, 111

[11] Jurić, M., Kantor, J., Lim, K-T., 2017, "The LSST Data Management System," Astronomical Data Analysis Software and Systems XXV, ASP Conference Series, Vol. 512, eds. N.P.F. Lorente, K. Shortridge, & R. Wayth, Astronomical Society of the Pacific

[12] Klein, L.J., Marianno, F.J., Albrecht, C.M. et al. 2015, "PAIRS: A scalable geo-spatial data analytics platform," 2015 IEEE International Conference on Big Data, https://www.ibm.com/us-en/marketplace/geospatial-big-data-analytics

[13] Komatsu, E., Smith, K. M., Dunkley, J. et al. 2011, "Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation," ApJS, 192, 18

[14] Lim, K.-T., Bosch, J., Dubois-Felsmann, G. 2019, "Large Synoptic Survey Telescope (LSST) Data Management System Design," LDM 148, https://github.com/lsst/LDM-148

[15] Lu, S., Shao, X., Freitag, M. et al. 2016, “IBM PAIRS Curated Big Data Service for Accelerated Geospatial Data Analytics and Discovery,” 2016 IEEE International Conference on Big Data, p. 2672

[16] Mokbel, M.F., Aref, W.G., Kamel, I. 2002, “Performance of Multi-Dimensional Space Filling Curves,” GIS’02, ACM 1-58113-XXX-X/02/0011

[17] Rhodes, J., Nichol, R.C., Aubourg, E., et al. 2017, “Scientific Synergy between LSST and Euclid,” ApJS, 233, 21

[18] Silva, D.R. 2015, “NOAO Strategic Plan A Forward Look,” National Optical Astronomy Observatory

[19] Vagg, D., O'Callaghan, D., O'Hógáin, F. et al. 2016, “GAVIP: a platform for Gaia data analysis,” Proceedings of the SPIE, Volume 9913, id. 99131V 19 pp. (2016).

[20] White, R.A. & Stemwedel, S.A. 1992, “The Quadrilateralized Spherical Cube and Quad-Tree For All Sky Data,” ASPC, 25, 379

[21] Whitman, R.T., Park, M.B., Ambrose, S.A., & Hoel, E.G. 2014, “Spatial Indexing and Analytics on Hadoop,” in Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, p 73

[22] Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S. and Stoica, I., 2010. “Spark: Cluster computing with working sets,” *HotCloud*, *10*(10-10), p.95