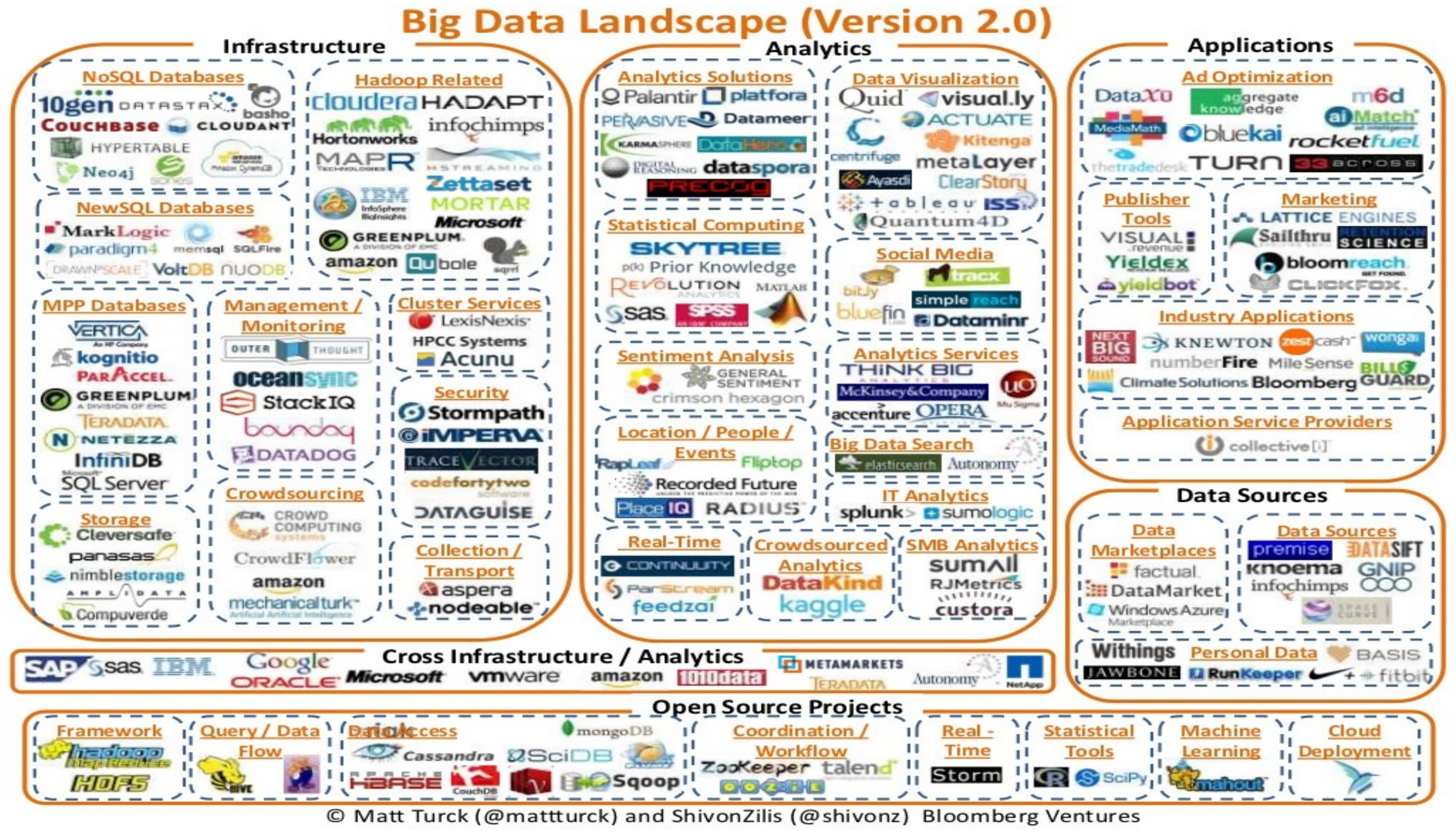
AN COMPARISON OF DATA STORAGE TECHNOLOGIES FOR REMOTE SENSING CYBER-INFRASTRUCTURES

RAJASEKAR KARTHIK, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TN, USA. karthikr@ornl.gov
PRANAB KANTI ROY CHOWDHURY, UNIVERSITY OF TENNESSEE, KNOXVILLE, TN, USA. proychow@vols.utk.edu

INTRODUCTION

- With latest generation EO systems, **Remote Sensing** data is being generated in very large volumes across multiple formats, around the clock, such as LiDAR, SAR, and Hyperspectral (in global level and high resolution).
- Advances in data acquisition techniques in Remote Sensing have resulted in introduction of multitude of operating payload having low GSD, higher revisit frequency, capable of operating round the clock and under any weather conditions. The datasets have become rich in higher spatial and spectral resolution, complex in structures and metadata, and diverse in applications areas.
- For example, NASA EOSDIS, had 8292 unique data products, summing up to 9.1 Petabytes (PB) and growing at 6.4 Terabytes (TB) daily during the period from Oct 1, 2013 to Sept 30, 2014. These datasets were used by about 2 million users with an average end user distribution volume of 27.9 Terabyte each day [1].
- Such recent trends in data-driven analysis necessities the need for **Cyber-infrastructures** capable of high-performance, scalable, or real-time computing that can efficiently handle "**Big Data**" workloads [2].
- In this poster, our focus will be on "**Data Storage**", and technologies being developed and used in other areas that can be applied in Remote Sensing Cyber-infrastructures.

BIG DATA LANDSCAPE



DATA STORAGE

- There exits variety of databases, ranging from traditional relational database management systems (RDBMS) to emerging NoSQL systems.
- Each database system provides its own benefits and is designed to suit different needs of cyber-infrastructures in remote sensing.
- In this poster, we will focus on three such needs and discuss a database system suited for each one:
  - Integration with existing architecture: RDBMS has been a conventional preference for many organizations and typically prefer not to replace it – but, at the same time, are interested in having the flexibility of evolving data sets and formats that are supported in NoSQL systems [3].
  - Scalability: Horizontal (scale out) or linear (throughput)
  - Real time analytics & batch processing

DATA STORAGE TECHNOLOGIES

RDBMS - PostgreSQL

(Integration with existing architecture)

- An widely used RDBMS due to its support for spatial and geographic object types via PostGIS [4].
- Prior to PostgreSQL 9.4, few major disadvantages made it unfit for emerging trends, that includes:
  - Adherence to rigid structure made it difficult to support non-relational data.
  - Poor support for JSON, one of the widely used intermediate data formats, which in-turn negatively affected the performance for querying and indexing GeoJSON [4].
- With 9.4, these challenges are being solved with the integration of
  - hstore: schema-less key-value pairs that enables efficient storage of semi-structured data.
    - Provides fast lookup and various indexing methods such as GIN, BTree, and Hash.
  - JSONB: A new data type that supports binary representation of JSON with fast access operations [4].
- Thus, cyber-infrastructures can use one database to support both relational and non-relational data models.

NoSQL - MongoDB

(Scalability)

- One of the leading NoSQL database systems that uses document-oriented data model [5].
- In general, NoSQL systems enjoy the power of scalability, as they follow the BASE (**B**asically **A**vailable, **S**oft state, **E**ventual consistency) model instead of ACID (**A**tomicity, **C**onsistency, **I**solation, **D**urability) model (that is used in RDBMS) [3, 5]:
- MongoDB's uniqueness in NoSQL lies in its easy and massively scalable nature, while providing rich set of features as well.
- Scalability:
  - Scale up or scale out horizontally
  - High availability & agile
  - Automatic sharding
  - Support for various sharding types such as range, tag-aware, hash [5, 6]
- Key features include:
  - Flexible data model: Data and images can be easily ingested regardless of their format and types.
  - Text search, analytics and cloud capable.
  - Supports variety of spatial indexes and query methods (though not as comprehensive as PostGIS) [5, 6].
  - Support for features that exists in RDBMS such as secondary indexes and sorting.

In-memory - Apache Spark

(Real time analytics + Batch Processing)

- A framework for fast, large-scale and parallel data processing [7, 8].
- Spark differentiates itself from the other two systems, by unifying both batch and real-time processing in one framework using In-memory based architecture
  - Resilient Distributed Dataset (RDD): one common abstraction shared between different components in Spark, and thus enabling it to support both types of processing.
  - Data is cached in memory.
  - Analysis can be run directly on cached data.
  - Uses RAM as much as disk
  - Write ahead log [7, 9].
- Unified core combines SQL, streaming, machine learning and analytics.
- Ideally suited for streaming, and iterative and interactive applications [7, 8].
- Can run either standalone, or on top of Hadoop or cloud.
- Other notable benefits include
  - Fault Tolerance
  - Functional programming
  - Well-defined APIs
  - Can be easily integration with other Apache projects like Hive, Cassandra an HBase [7, 8].

CONCLUSION

- In this poster, we have discussed three technologies that can be applied in Remote Sensing Cyber-infrastructures for application integration, scalable, and real-time computing:
  - RDBMS – PostgreSQL 9.4 brings the power of flexibility and high-performance of NoSQL, while yet supporting its relational data model.
  - NoSQL – MongoDB provides easy scalability and rich set of features.
  - In-memory – Apache Spark provides one framework for fast and large-scale data processing for both batch and real-time.
- Our next step is collect wide variety of large datasets, evaluate each of these systems on top of the same, and explore various optimization measures.
- We also plan to explore newer and proposed technologies in "Big Data" landscape

REFERENCES

1. EOSDIS System Performance. URL https://earthdata.nasa.gov/about-eosdis/performance
2. Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., and Jie, W. (2014). Remote sensing big data computing: Challenges and opportunities. *Future Generation Computer Systems*.
3. Karthik, R., and Lu, W. Scaling an Urban Emergency Evacuation Framework: Challenges and Practices. 2nd workshop on Big Data and Urban Informatics (BDUIC), 2014.
4. Karthik, R. SAME4HPC: A Promising Approach in Building a Scalable and Mobile Environment for High-Performance Computing. 3rd ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (MobiGIS), 2014.
5. Banker, K. (2011). *MongoDB in action*. Manning Publications Co.
6. MongoDB. URL http://mongodb.org
7. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., Mccauley, M., ... & Stoica, I. (2012). Fast and interactive analytics over Hadoop data with Spark. *USENIX; login*, *37*(4), 45-51.
8. Apache Spark. URL https://spark.apache.org/
9. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., ... & Stoica, I. (2012, April). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation* (pp. 2-2). USENIX Association.