Data Intensive Systems

Big Data

27-Sep-2022

Dr. Sumit Kalra
Assistant Professor
Department of Computer Science & Engineering
Indian Institute of Technology Jodhpur

Slide Credits:
Sneha Prasad
PhD Scholar
IoT IDRP
IIT Jodhpur

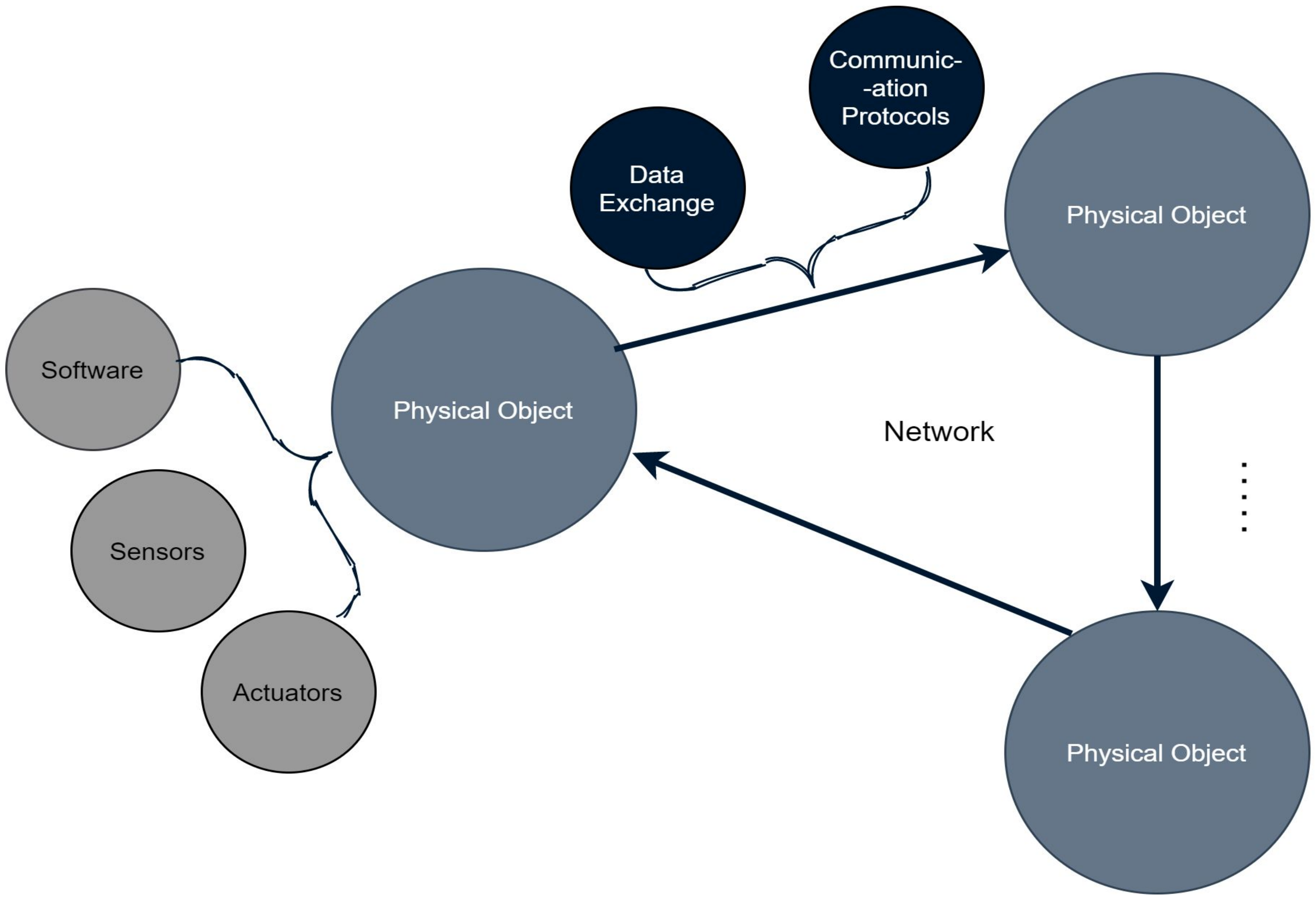
Outline

Motivation

- IoT
- IoT Services
- Big Data
- Big Data Analytics
- Relationship between IoT and Big Data Analytics
- Big Data in IoT

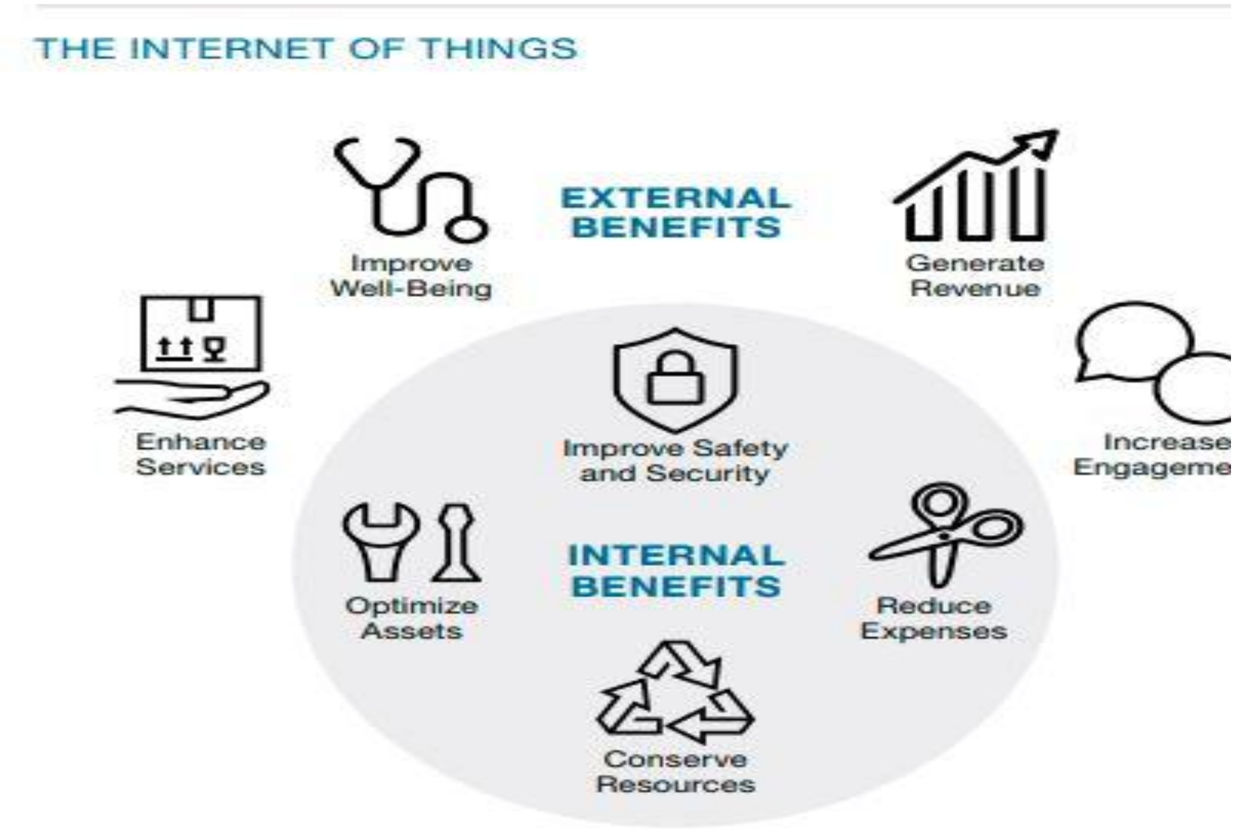
Internet of Things

Definition :

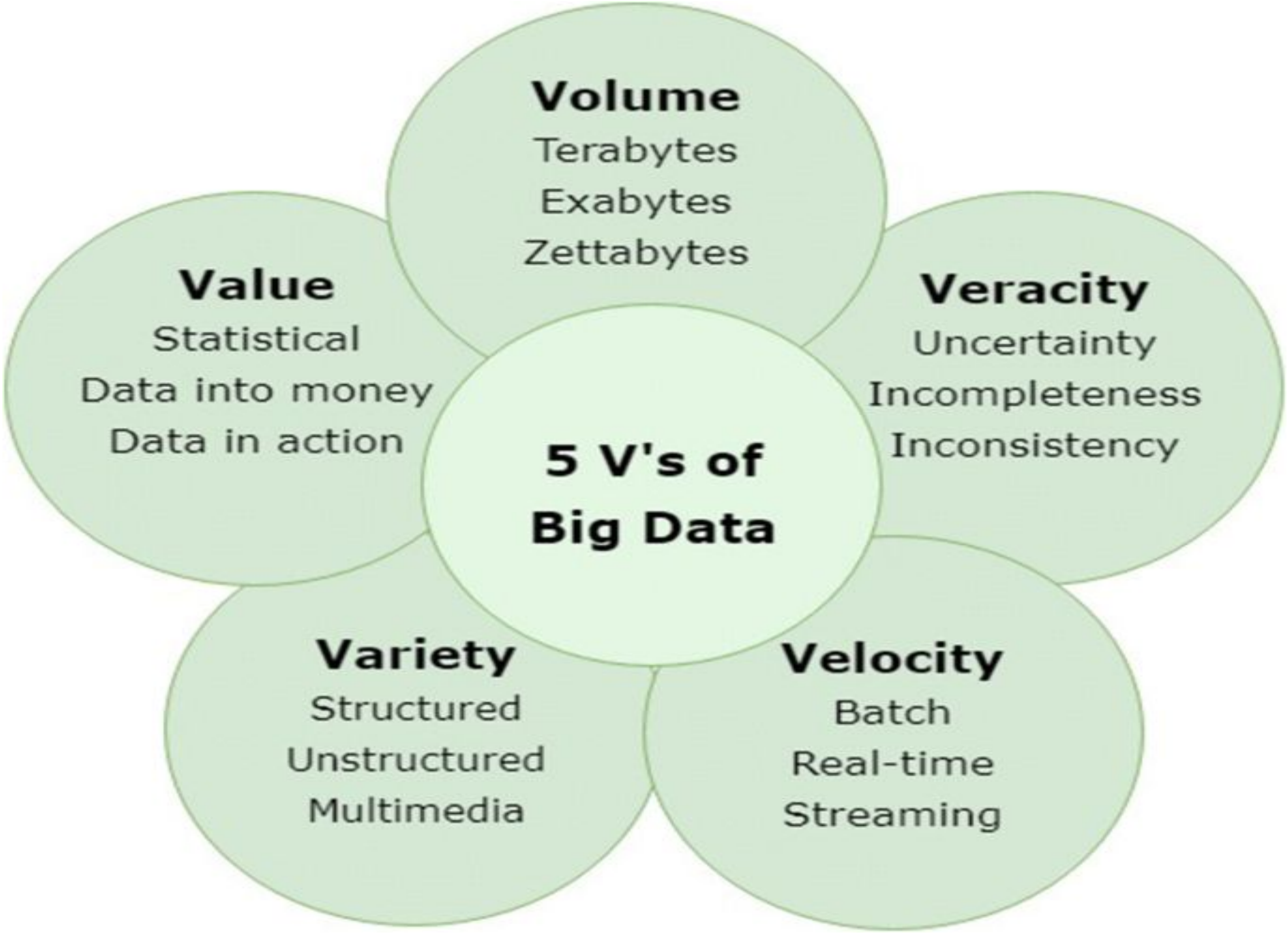


IoT Services

The Internet of Things (IoT) will do just that by producing unprecedented volume, velocity and variety of data. This will force organizations to re-architect their data and analytics capabilities, adopt new data management technologies and platforms, and create new data governance policies and practices to act upon all of this data.



Big Data



Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big Data*, *6*(1), 44

Big Data Analytics

BIG DATA PHASE 1	BIG DATA PHASE 2	BIG DATA PHASE 3
Period: 1970-2000	Period: 2000-2010	Period: 2010-present
DBMS-based, structured content: • RDBMS & data warehousing • Extract Transfer Load • Online Analytical Processing • Dashboards & scorecards • Data mining & statistical analysis	Web-based, unstructured content • Information retrieval and extraction • Opinion mining • Question answering • Web analytics and web intelligence • Social media analytics • Social network analysis • Spatial-temporal analysis	Mobile and sensor-based content • Location-aware analysis • Person-centered analysis • Context-relevant analysis • Mobile visualization • Human-Computer-Interaction

Relationship between IoT and Big Data Analytics

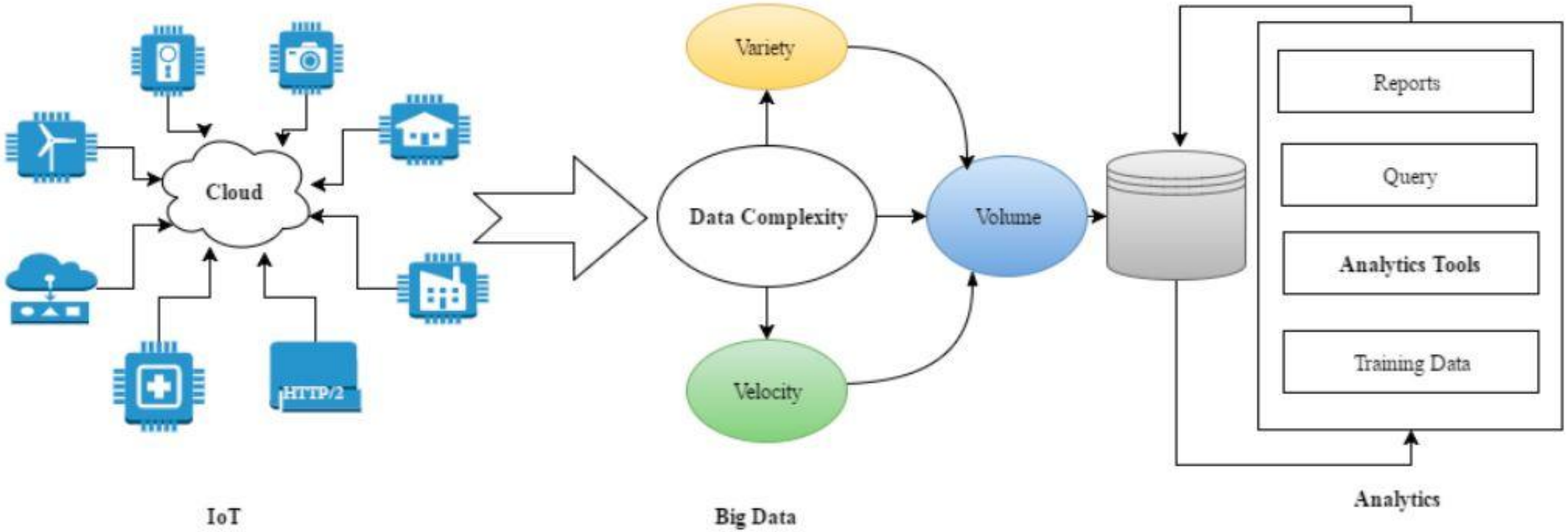


Fig 1. Relationship between IoT and Big Data Analytics

Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob, I. (2017). Big IoT data analytics: architecture, opportunities, and open research challenges. *IEEE Access*, *5*, 5247-5261.

Big Data in IoT

Time and Space correlation in the sensor data,

Real-Time
Unstructured
Large Amount

Big Data in IoT

High level of noise and redundancy in datasets

Context-driven nature of data sensing and processing

Taherkordi, A., Eliassen, F., & Horn, G. (2017, April). From IoT big data to IoT big services. In *Proceedings of the Symposium on Applied Computing* (pp. 485-491).

Data Intensive System

- Data Intensive System
- Software Architecture of Data Intensive System
- Features of Data Intensive System

Data Intensive System

Big Data

Parallel Processing

Distributed Architecture

Software Architecture of Data Intensive System

Software Architecture : Logical Organization of Software Components and their relationships

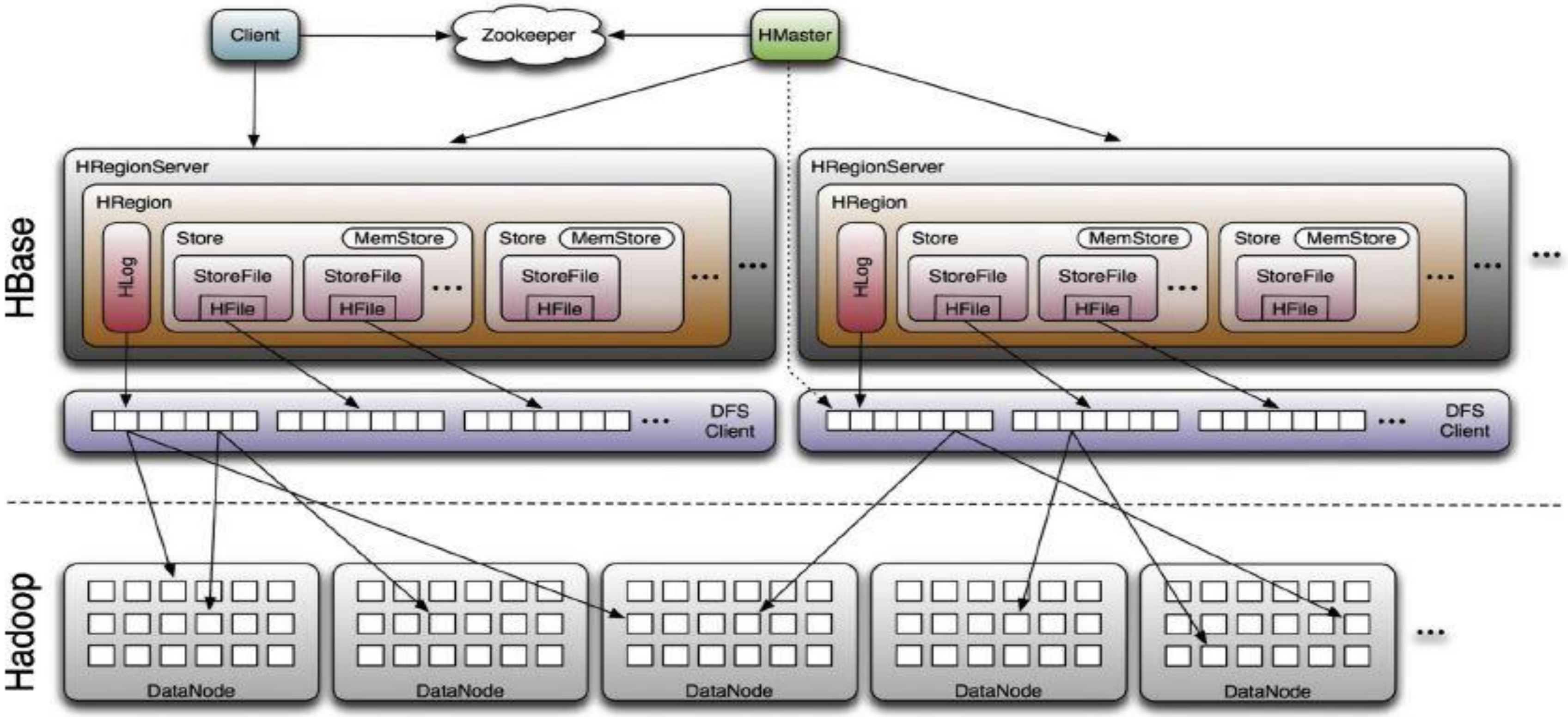
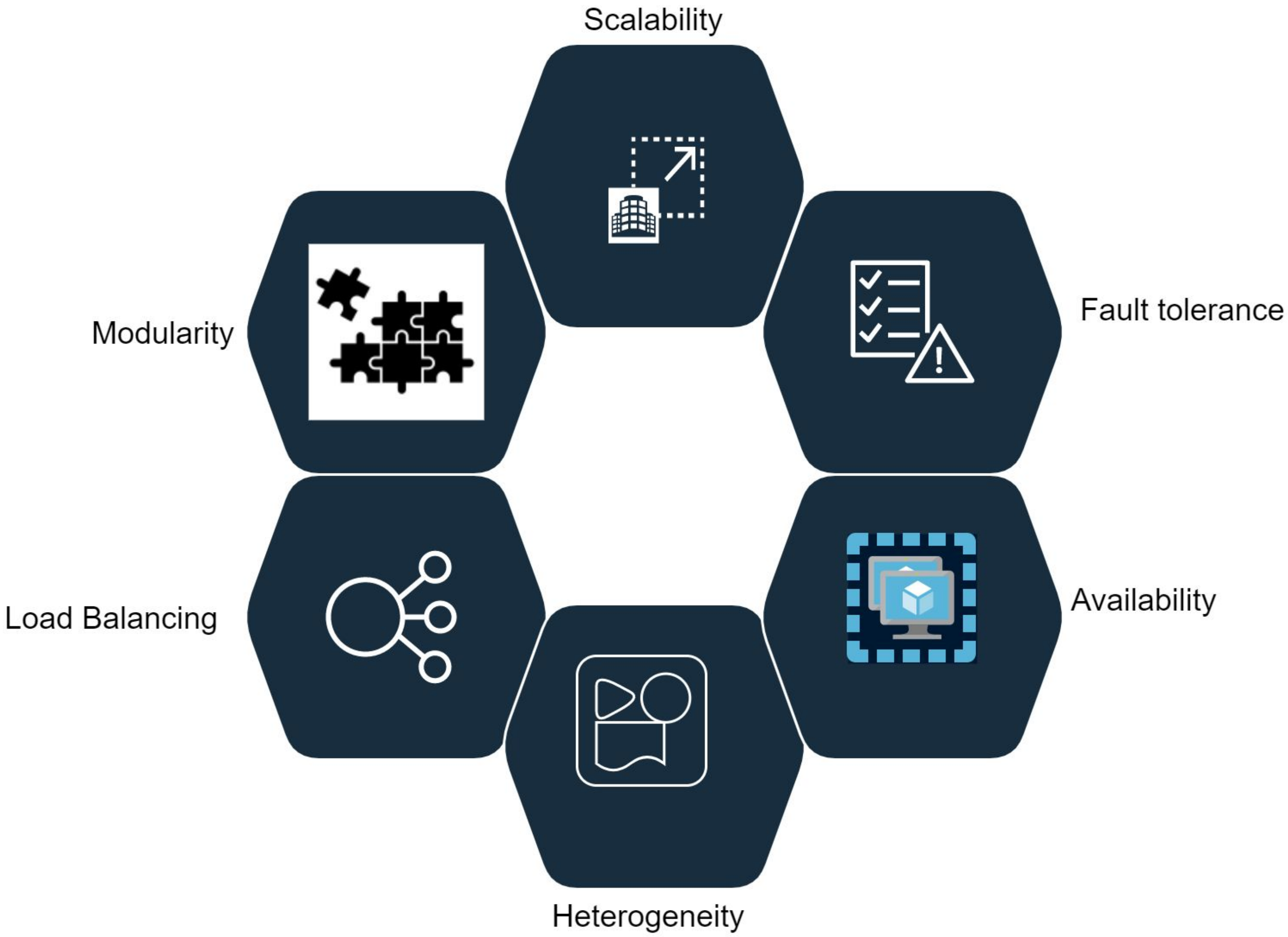


Fig 2. Hadoop Architecture

Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information sciences*, *275*, 314-347..

Features of Data Intensive System



State of The Art Frameworks

- Current Status
- Time Span of Selected Frameworks
- Theory of Evolution of Data Intensive Systems
- Data Processing Category
- Data Analytical Category

Current Status

S.No.	Name of the Tool	1st Version	Latest Version	Software License Category	Maintained By	Categorization
1	Apache Hadoop	2006, (1)	2018, (3.1.1)	Apache License 2.0	Apache Software Foundation	Data Processing
2	Apache Spark	2012, (0.5)	(2.4)	Apache License 2.0	Apache Software Foundation	Data Processing
3	Apache Tez	2014, (0.5.0)	2019, (0.9.2)	Apache License 2.0	Apache Software Foundation	Data Processing
4	Apache Kafka	0.7.0	2.5.0	Apache License 2.0	Apache Software Foundation	Data Processing
5	Apache Samza	2014, (0.7)	2020, (1.4)	Apache License 2.0	Apache Software Foundation	Data Processing
6	Apache Flink	2015, (0.9.1)	2020, (1.10.0)	Apache License 2.0	Apache Software Foundation	Data Processing
7	Apache Storm	2012, (0.8.0)	2019, (2.1.0)	Apache Software License 2.0	Apache Software Foundation	Data Processing
8	Torch	2002	(7.0)	BSD License4	Facebook AI Research Lab	Data Analytics- Deep Learning
9	Keras	2015, (0.0.1)	2019, (2.3.1)	MIT	Google, Microsoft, Amazon and Nvidia	Data Analytics- Deep Learning
10	Tensorflow	2017, (1.0.0)	2020, (2.2.0)	Apache License 2.0	Google Brain Team	Data Analytics- Deep Learning
11	Mxnet	2015	(1.6.0)	Apache License 2.0	Apache Software Foundation	Data Analytics- Deep Learning
12	Caffe2	2017	2018	BSD License	Facebook	Data Analytics – Deep Learning
13	BigDL	2016	2019, (1.10.0)	Apache License 2.0	Intel	Data Analytics- Deep Learning
14	RapidMiner	(5.0.0)	(9.6.0)	AGPL 3.0	RapidMiner Inc	Data Analytics – Deep Learning

Time Span of Processing and Analytical Frameworks

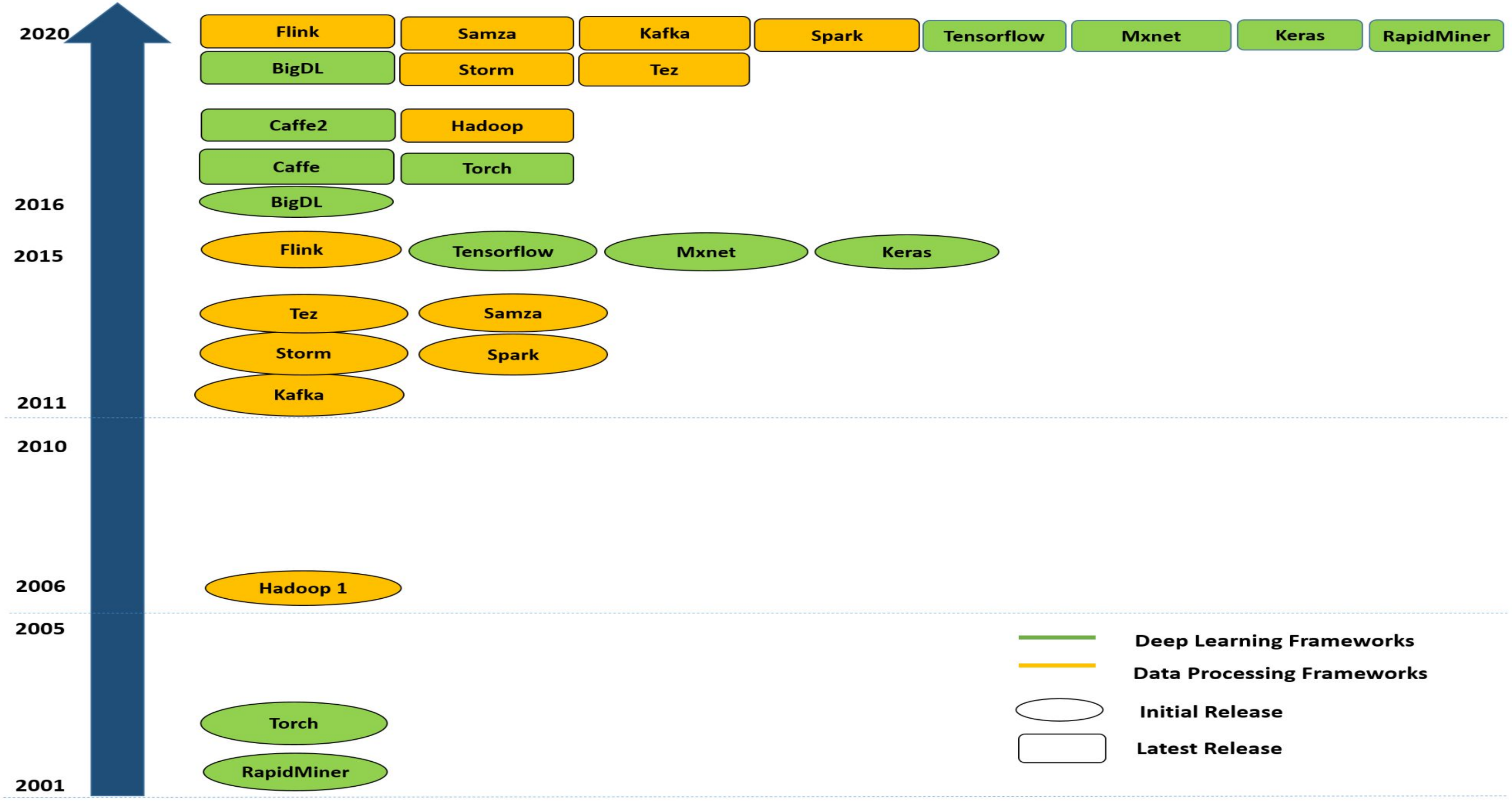


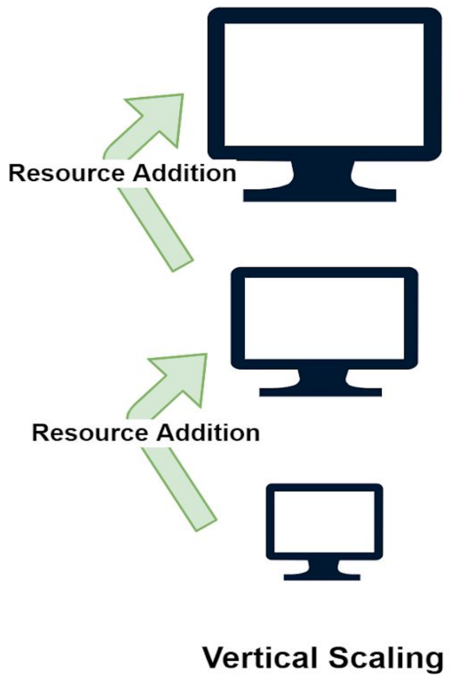
Fig 3. Time Span of Processing and Analytical Frameworks

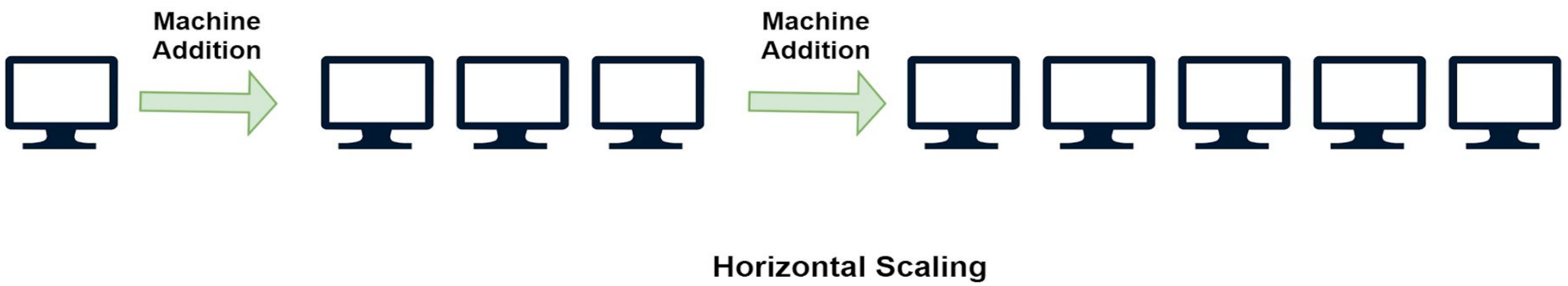
Technical Approaches

- Scalability
- Master Slave Architecture
- Parameter Server Architecture
- Data Parallelism and Model Parallelism
- Fault Tolerance
- Replication
- Check-pointing
- RDD Lineage

Scalability

Transformation of System Size based on Demand of Resources





Inclusion of more power and memory resources to a single machine.

Addition of resources is not in the machine but by the machines

Master Slave Architecture

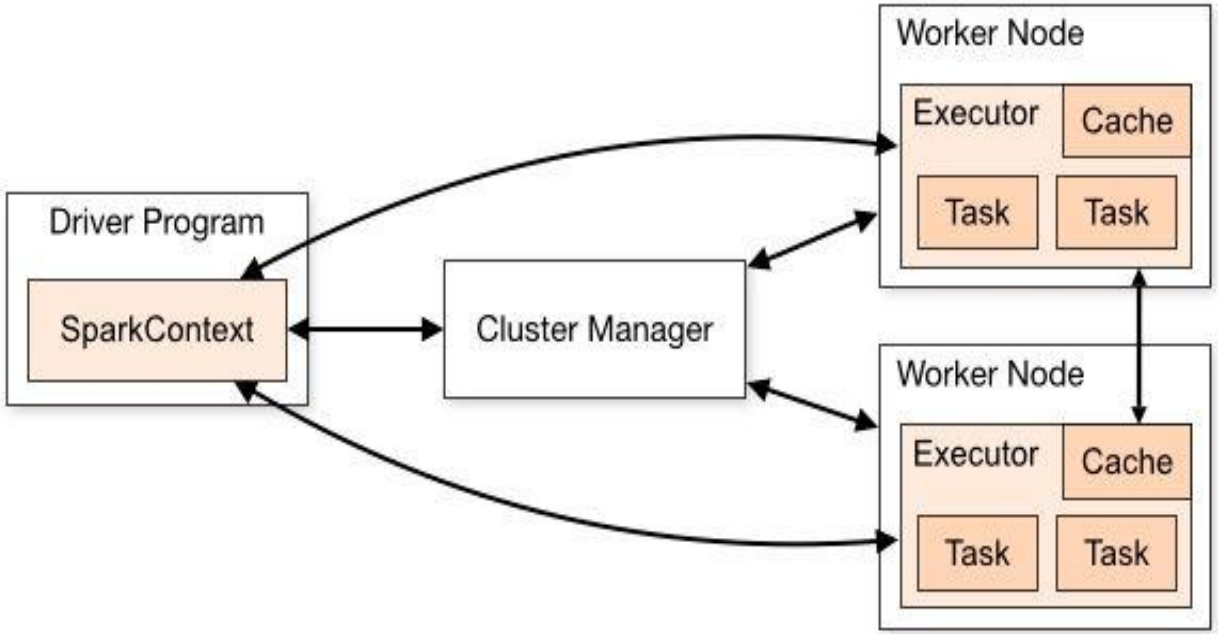


Fig 7. Spark Cluster Architecture

Parameter Server

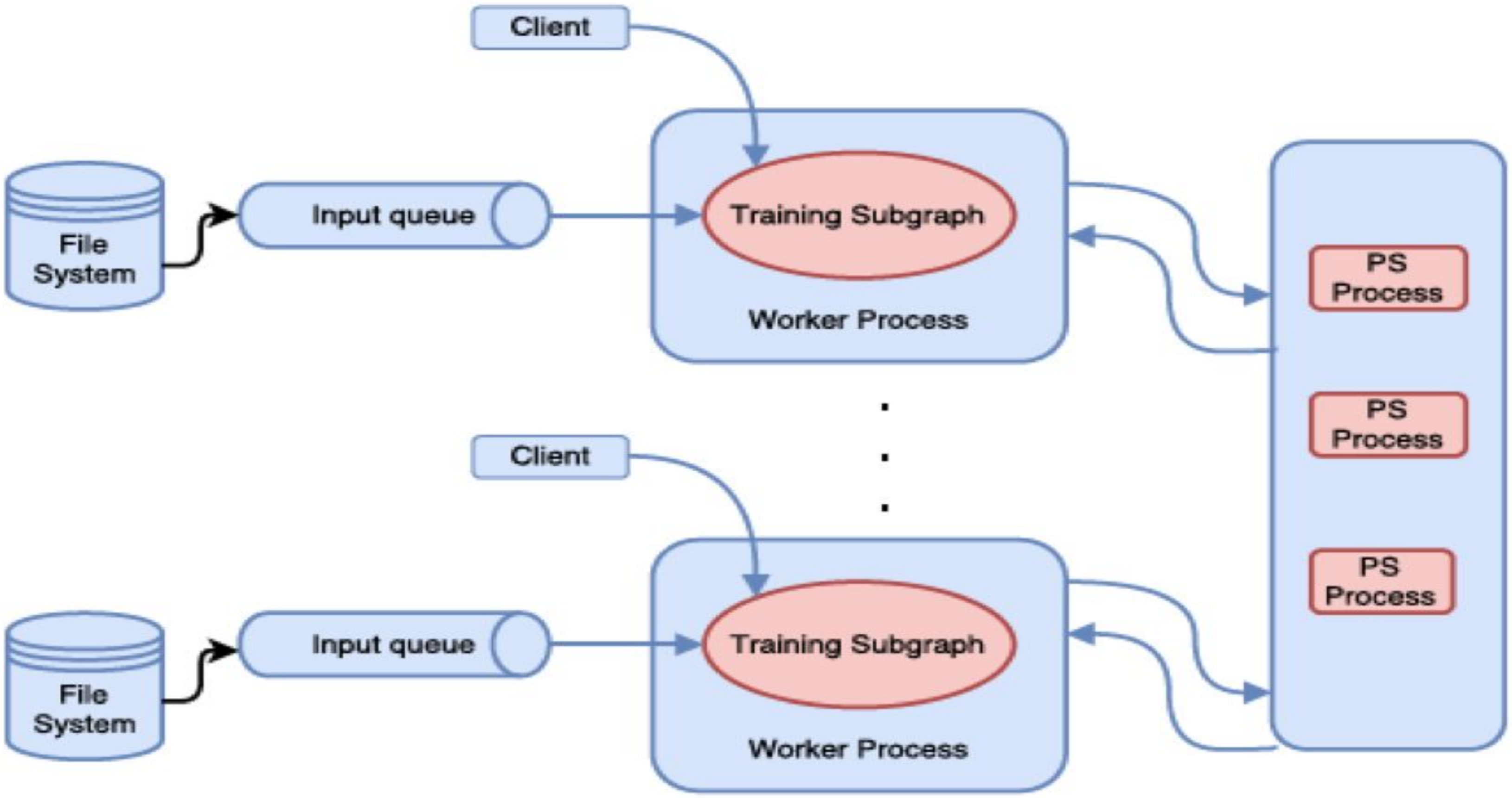


Fig 8. TensorFlow Between-Graph Replicated Training

Zhang, K., Alqahtani, S., & Demirbas, M. (2017, July). A comparison of distributed machine learning platforms. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-9). IEEE.

Data Parallelism and Model Parallelism

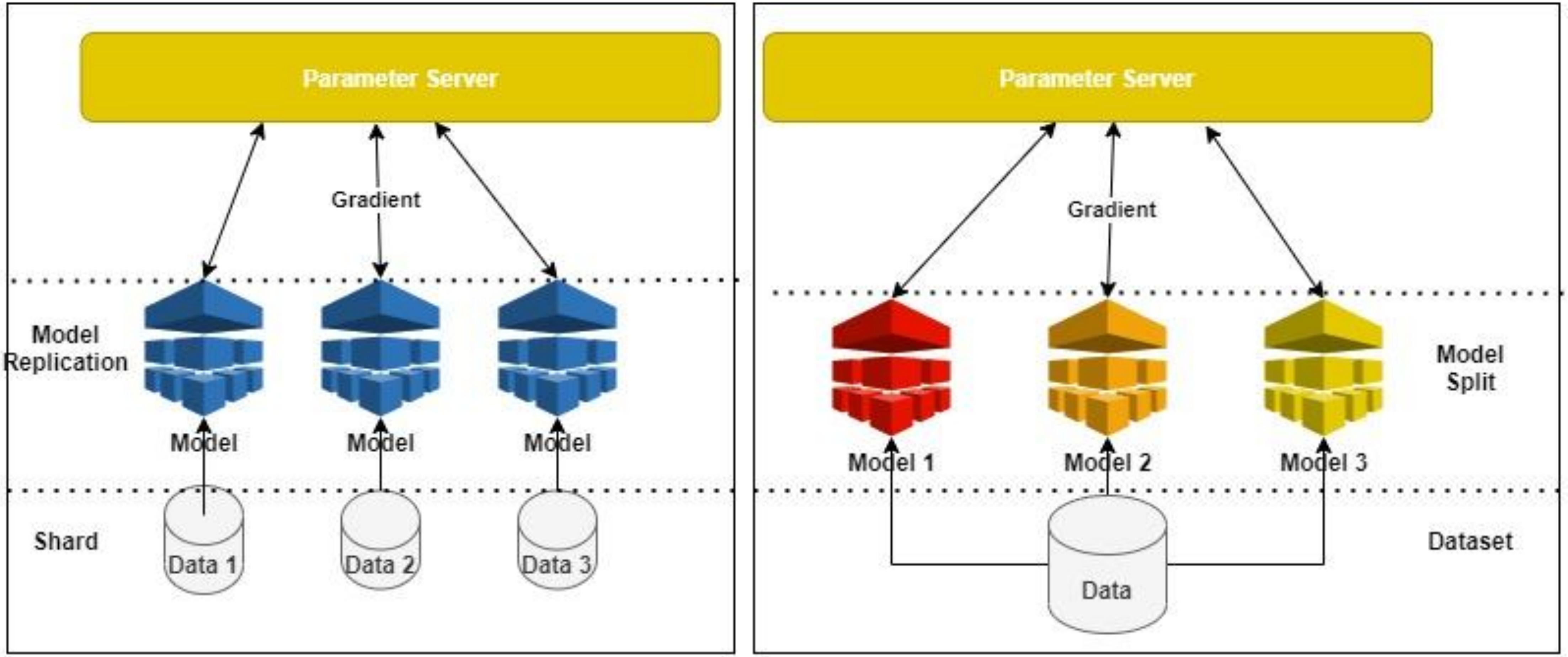


Fig 9. Data Parallelism and Model Parallelism

Fault Tolerance

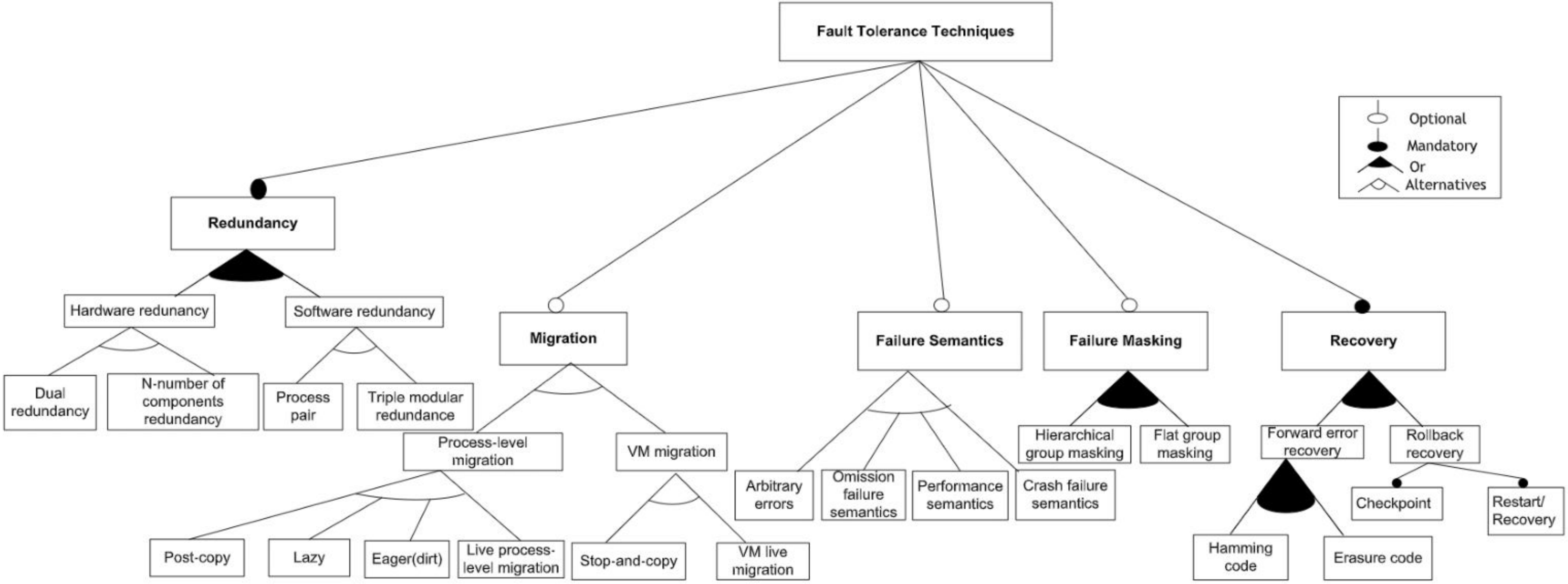


Fig 10. Fault Tolerance Techniques

Egwutuoha, I. P., Levy, D., Selic, B., & Chen, S. (2013). A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *The Journal of Supercomputing*, *65*(3), 1302-1326.

Replication

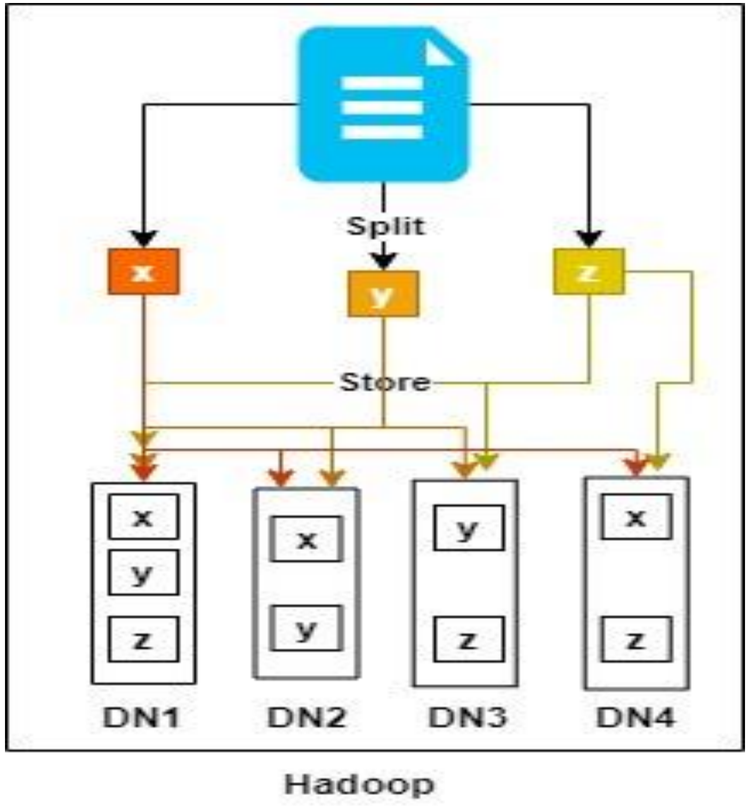


Fig 11. Fault tolerance Architecture of Hadoop

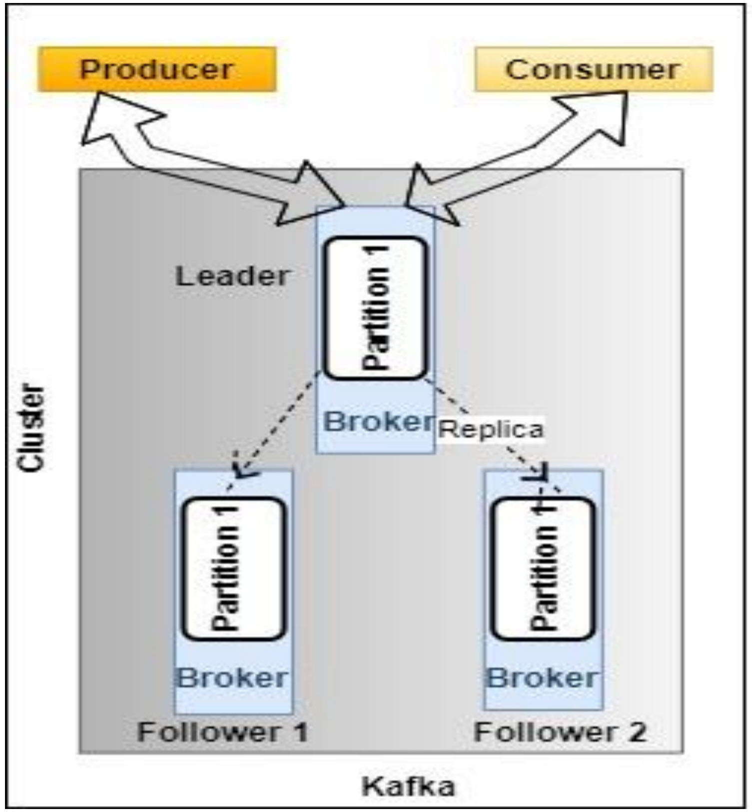


Fig 12. Fault tolerance Architecture of Kafka

Check-pointing

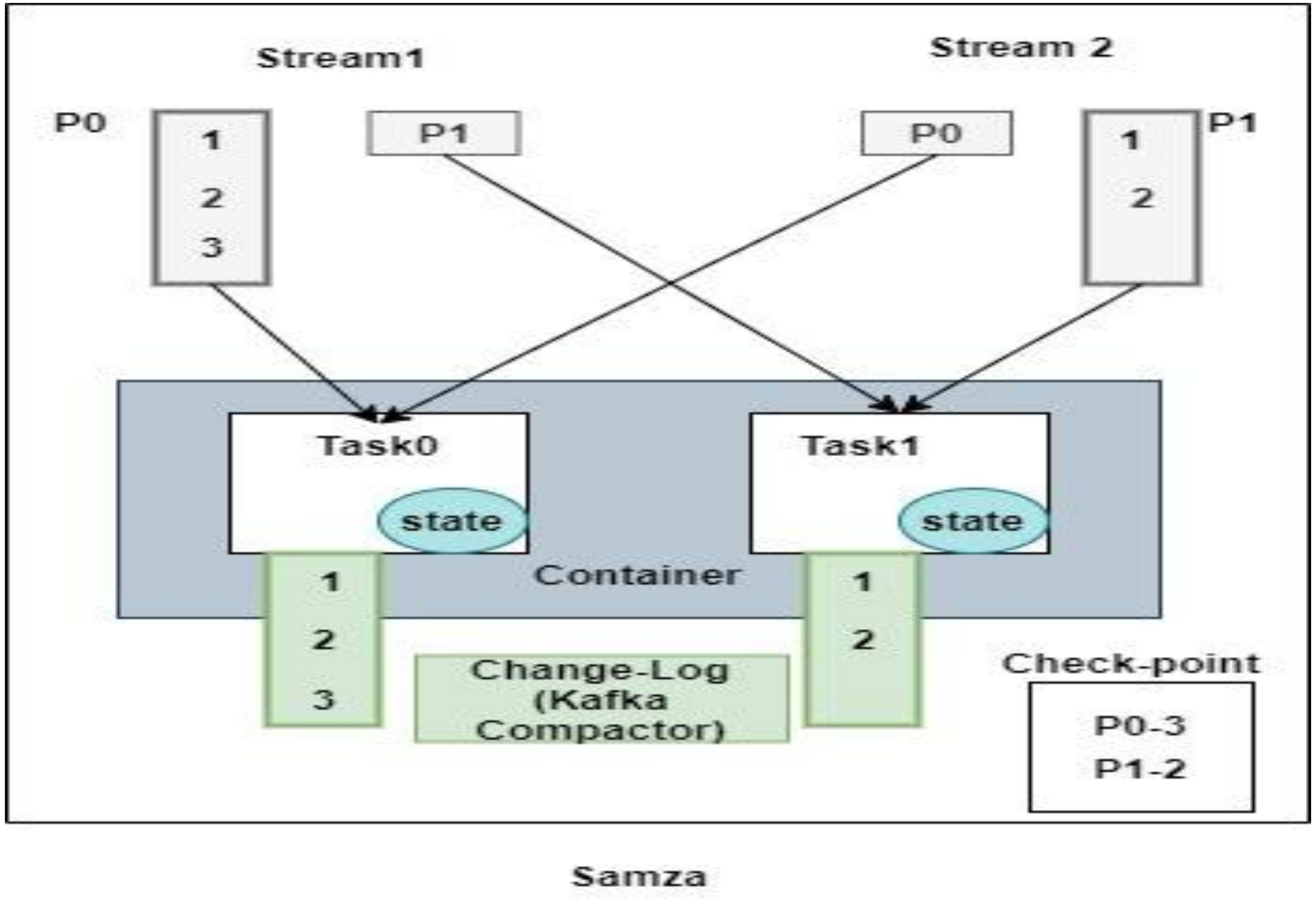
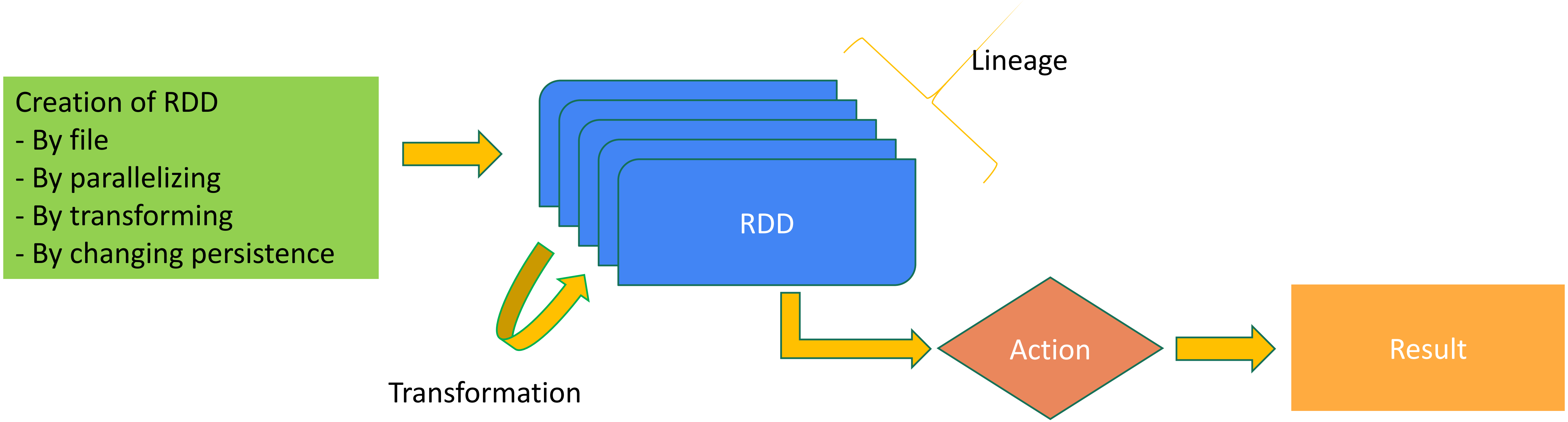


Fig 13. Architecture of Apache Samza

Resilient Distributed Dataset

A read-only collection of objects partitioned across a set of machines that can be rebuilt if a partition is lost.



RDD Lineage

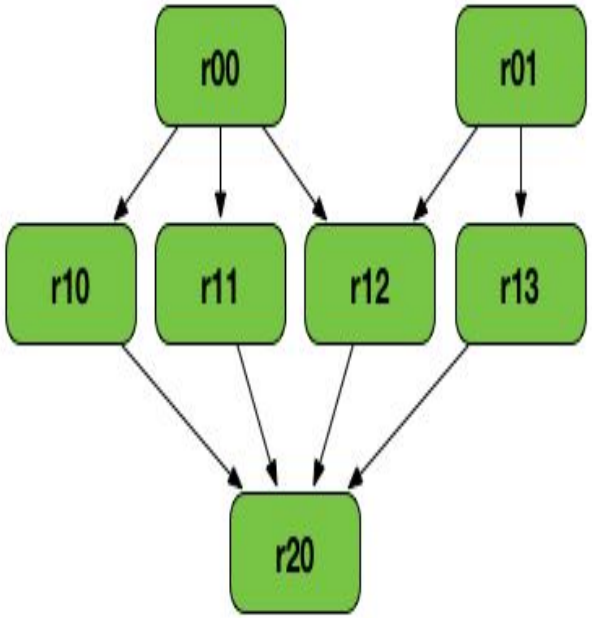


Fig 14. RDD Lineage

```
val r00 = sc.parallelize(0 to 9)
val r01 = sc.parallelize(0 to 90 by 10)
val r10 = r00 cartesian r01
val r11 = r00.map(n => (n, n))
val r12 = r00 zip r01
val r13 = r01.keyBy(_ / 20)
val r20 = Seq(r11, r12,
r13).foldLeft(r10)(_ union _)
```

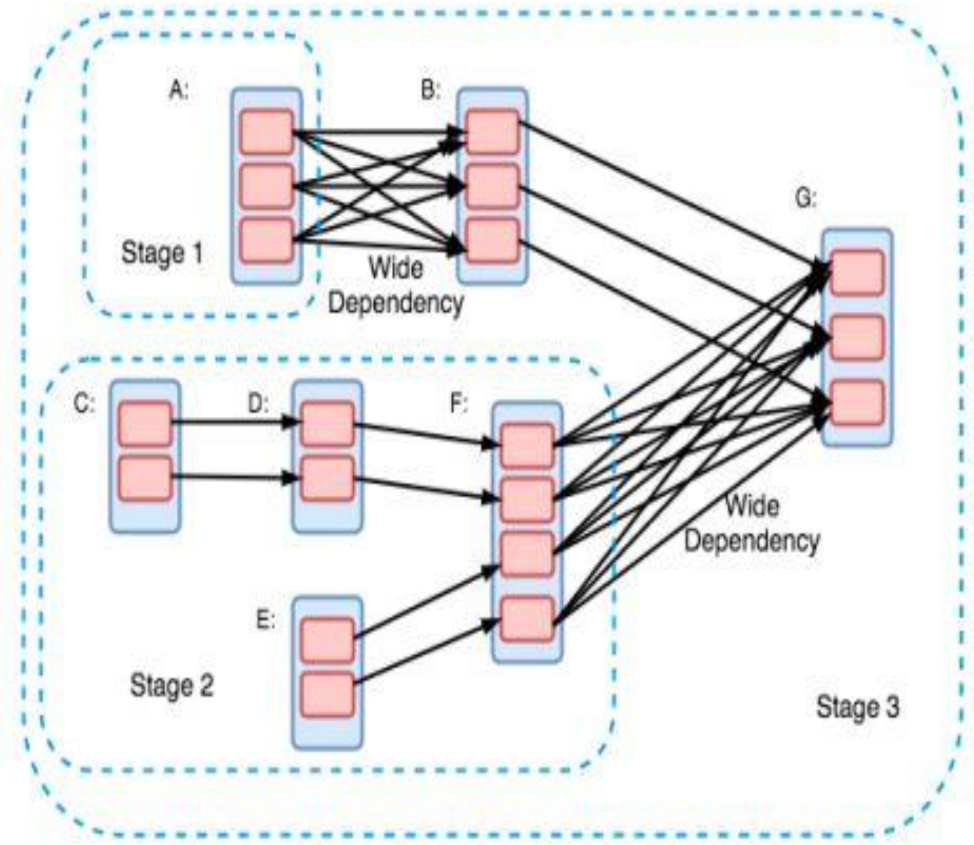


Fig 15. RDD Stages

Job Agents and JVM

Scalability – Job Agents

Fault tolerance - JVM

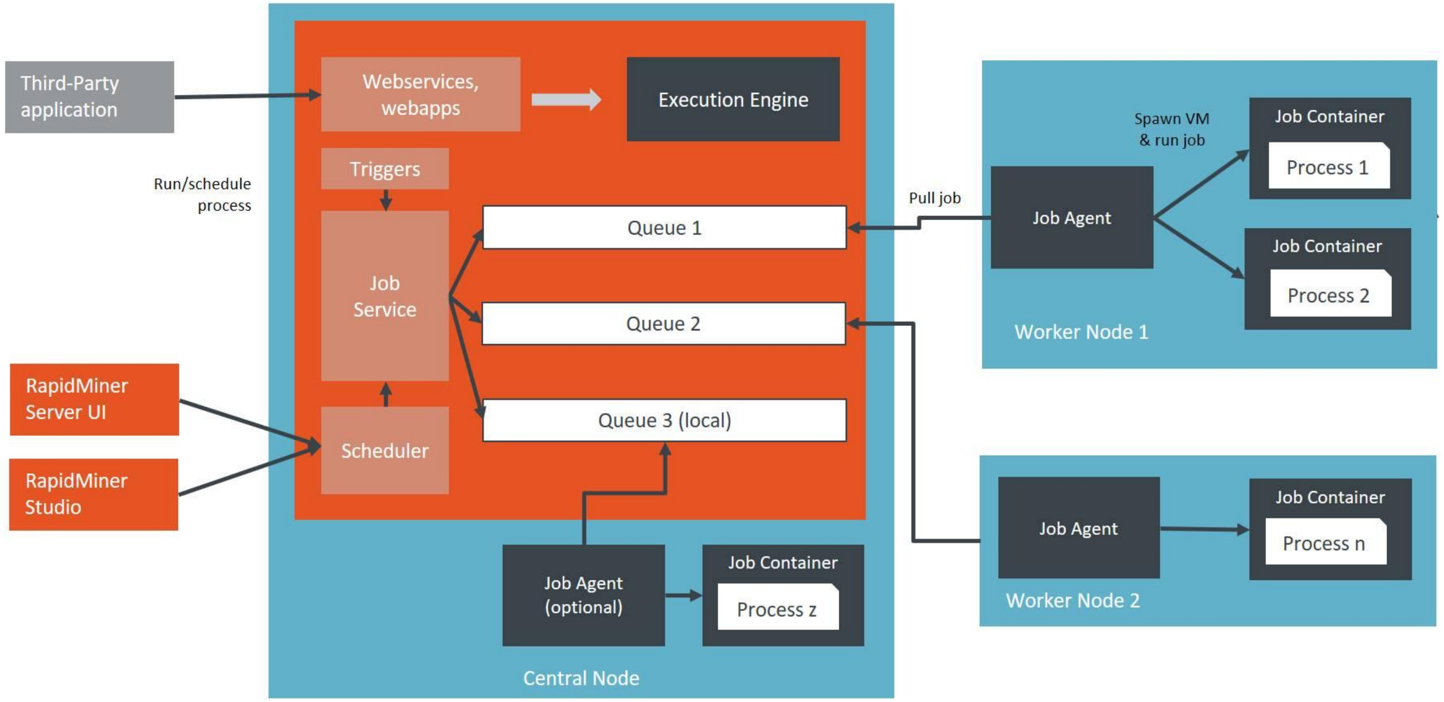


Fig 16. RapidMiner Architecture

Open Issues

Event Stream Ingestion
Data Retention
Distributed Learning
Data Privacy
Interoperability
Scalability
Fault tolerance

Conclusion

The state-of-the-art methodologies involved in achieving scalability and fault-tolerance in data processing and data analytical architectural frameworks, are surveyed to accomplish robustness in sustaining reliability, availability and maintainability of a large scale distributed system.

References

[1] Hung, M. (2017). Leading the iot, gartner insights on how to lead in a connected world. *Gartner Research*, 1-29.

[2] Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big Data*, *6*(1), 44.

[3] Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob, I. (2017). Big IoT data analytics: architecture, opportunities, and open research challenges. *IEEE Access*, *5*, 5247-5261.

[4] Iannacci, J. (2018). Internet of things (IoT); internet of everything (IoE); tactile internet; 5G–A (not so evanescent) unifying vision empowered by EH-MEMS (energy harvesting MEMS) and RF-MEMS (radio frequency MEMS). *Sensors and Actuators A: Physical*, *272*, 187-198.

References

[5] Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information sciences*, *275*, 314-347.

[6] Zhang, K., Alqahtani, S., & Demirbas, M. (2017, July). A comparison of distributed machine learning platforms. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-9). IEEE.

[7] Egwutuoha, I. P., Levy, D., Selic, B., & Chen, S. (2013). A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *The Journal of Supercomputing*, *65*(3), 1302-1326.

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

References

[9] Laskowski, J. (2017). RDD Lineage-Logical Execution Plan. *URL: https://jaceklaskowski. gitbooks. io/mastering-apache-spark/spark-rdd-lineage. Html*.

[10] ReleaseNotes- RapidMiner Documentation https://docs.rapidminer.com/8.0/server/releases/ Accessed: 2020-12-26

[11] Hao Zhang, Zeyu Zheng, Shizhen Xu, Wei Dai, Qirong Ho, Xiaodan Liang, Zhiting Hu, Jinliang Wei, Pengtao Xie, and Eric P Xing. 2017. Poseidon:An efficient communication architecture for distributed deep learning on{GPU}clusters. In2017 USENIX Annual Technical Conference USENIXATC 17). 181–193.

[12] Kuo Zhang, Salem Alqahtani, and Murat Demirbas. 2017. A comparison of distributed machine learning platforms. In2017 26th InternationalConference on Computer Communication and Networks (ICCCN). IEEE, 1–9.

References

[13] Zhenyun Zhuang, Tao Feng, Yi Pan, Haricharan Ramachandra, and Badri Sridharan. 2016. Effective multi-stream joining in apache samzaframework. In2016 IEEE International Congress on Big Data (BigData Congress). IEEE, 267–274.

[14] Yanzhao Wu, Ling Liu, Calton Pu, Wenqi Cao, Semih Sahin, Wenqi Wei, and Qi Zhang. 2019. A comparative measurement study of deep learningas a service framework.IEEE Transactions on Services Computing(2019).

[15] Wolfram Wingerath, Norbert Ritter, and Felix Gessert. 2019. General-Purpose Stream Processing. InReal-Time & Stream Data Management.Springer, 57–74.

[16] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S. Rellermeyer. 2020. A Survey on DistributedMachine Learning. 53, 2, Article 30 (March 2020), 33 pages. https://doi.org/10.1145/3377454

References

[17] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, et al.2019. Tensorflow. js: Machine learning for the web and beyond.arXiv preprint arXiv:1901.05350(2019).

[18] Rupinder Singh and Puneet Jai Kaur. 2016. Analyzing performance of Apache Tez and MapReduce with hadoop multinode cluster on Amazoncloud.Journal of Big Data3, 1 (2016). https://doi.org/10.1186/s40537-016-0051-6

[19] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, et al.2019. Tensorflow. js: Machine learning for the web and beyond.arXiv preprint arXiv:1901.05350(2019).

[20] S Surshanov. 2015. Using apache storm for big data.Computer Modelling & New Technologies19 (2015), 14–17.Facebook Open Source. 2020. Caffe2 Tutorials Overview | Caffe2. https://caffe2.ai/docs/tutorials.html. (Accessed on 08/31/2020).

References

[21] Facebook Open Source. 2020. Caffe2 Tutorials Overview | Caffe2. https://caffe2.ai/docs/tutorials.html. (Accessed on 08/31/2020).

[22] S. Shi, Q. Wang, P. Xu, and X. Chu. 2016. Benchmarking State-of-the-Art Deep Learning Software Tools. In2016 7th International Conference onCloud Computing and Big Data (CCBD). IEEE, 99–104.

[23] Ce-Kuen Shieh, Sheng-Wei Huang, Li-Da Sun, Ming-Fong Tsai, and Naveen Chilamkurti. 2017. A topology-based scaling mechanism for A pache Storm.International Journal of Network Management27, 3 (2017), e1933.

[24] Rishika Shree, Tanupriya Choudhury, Subhash Chand Gupta, and Praveen Kumar. 2017. KAFKA: The modern platform for data management andanalysis in big data domain. In2017 2nd International Conference on Telecommunication and Networks (TEL-NET). IEEE, 1–5.Jonathan Samosir, Maria Indrawan-Santiago, and Pari Delir Haghighi. 2016. An evaluation of data stream processing systems for data drivenapplications.Procedia Computer Science80 (2016), 439–449.

References

[25] Jonathan Samosir, Maria Indrawan-Santiago, and Pari Delir Haghighi. 2016. An evaluation of data stream processing systems for data drivenapplications.Procedia Computer Science80 (2016), 439–449.

[26] Sanhita Sarkar. 2020. Platform and software framework for data intensive applications in the cloud. US Patent 10,810,220.

[27] Bikas Saha, Hitesh Shah, Siddharth Seth, Gopal Vijayaraghavan, Arun Murthy, and Carlo Curino. 2015. Apache Tez: A Unifying Framework forModeling and Building Data Processing Applications. InProceedings of the 2015 ACM SIGMOD International Conference on Management of Data(SIGMOD ’15). Association for Computing Machinery, New York, NY, USA, 1357–1369. https://doi.org/10.1145/2723372.2742790

[28] Navina Ramesh. 2015. Apache Samza, LinkedIns Framework for Stream Processing.thenewstack. io(2015). Shadi A Noghabi, Kartik Paramasivam, Yi Pan, Navina Ramesh, Jon Bringhurst, Indranil Gupta, and Roy H Campbell. 2017.

References

[29] Shadi A Noghabi, Kartik Paramasivam, Yi Pan, Navina Ramesh, Jon Bringhurst, Indranil Gupta, and Roy H Campbell. 2017. Samza: stateful scalablestream processing at LinkedIn.Proceedings of the VLDB Endowment10, 12 (2017), 1634–1645.

[30] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learningapplications and challenges in big data analytics.Journal of Big Data2, 1 (2015), 1–21.

[31] Dhabaleswar K. (DK) Panda, Ammar Ahmad Awan, and Hari Subramoni. 2019. High Performance Distributed Deep Learning: A Beginner's Guide.InProceedings of the 24th Symposium on Principles and Practice of Parallel Programming (PPoPP '19). Association for Computing Machinery, NewYork, NY, USA, 452–454. https://doi.org/10.1145/3293883.3302260

[32] Dariusz Mrozek. 2018. Foundations of the Hadoop Ecosystem. InScalable Big Data Analytics for Protein Bioinformatics. Springer, 137–150. John Nickolls, Ian Buck,

References

[33] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. 2008. Scalable parallel programming with CUDA.Queue6, 2 (2008), 40–53. https://doi.org/10.1145/1365490.1365500

[34] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. 2018. Deep learning for IoT big data and streaming analytics: A survey.IEEE Communications Surveys & Tutorials20, 4 (2018), 2923–2960.

[35] Sparsh Mittal and Jeffrey S. Vetter. 2015. A Survey of CPU-GPU Heterogeneous Computing Techniques.ACM Comput. Surv.47, 4, Article 69 (July2015), 35 pages. https://doi.org/10.1145/2788396

[36] Ruben Mayer and Hans-Arno Jacobsen. 2020. Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques, and Tools.ACMComput. Surv.53, 1, Article 3 (Feb. 2020), 37 pages. https://doi.org/10.1145/3363554

Sparsh Mittal and Jeffrey S. Vetter. 2015. A Survey of CPU-GPU Heterogeneous Computing Techniques.ACM Comput. Surv.47, 4, Article 69 (July2015), 35 pages. https://doi.org/10.1145/2788396

References

[37] Sparsh Mittal and Jeffrey S. Vetter. 2015. A Survey of CPU-GPU Heterogeneous Computing Techniques.ACM Comput. Surv.47, 4, Article 69 (July2015), 35 pages. https://doi.org/10.1145/2788396

[38] Sina Gholamian, Wojciech Golab, and Paul AS Ward. 2017. Efficient incremental data analytics with apache spark. In2017 IEEE InternationalConference on Big Data (Big Data). IEEE, 2859–2868.

[39] Leonidas Fegaras. 2016. Incremental query processing on big data streams.IEEE Transactions on Knowledge and Data Engineering28, 11 (2016),2998–3012.

[40] Sara Landset, Taghi M. Khoshgoftaar, Aaron N. Richter, and Tawfiq Hasanin. 2015. A survey of open source tools for machine learning with bigdata in the Hadoop ecosystem.Journal of Big Data2, 1 (2015), 1–36. https://doi.org/10.1186/s40537-015-0032-1