Infrastructure for Time Critical Applications of Big Data Systems Using Light Weight YARN Architecture

Saravanan A.M. (✉ saravanan.am@yahoo.com)

Muthurangam Government Arts College (Autonomous)

K. Loheswaran

CMR College of Engineering and Technology

G. Naga Rama Devi

CMRIMT: CMR Institute of Technology

Karuppathal R

PSNA CET: PSNA College of Engineering and Technology

C Balakrishnan

Alagappa University

T.Aditya Sai Srinivas

G Pullaiah College of Engineering and Technology

Research Article

Keywords: YARN Architecture, MapReduce, lightweight architecture, Task Trackers, Big Data

Posted Date: April 22nd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-436282/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

INFRASTRUCTURE FOR TIME CRITICAL APPLICATIONS OF BIG DATA SYSTEMS USING LIGHT WEIGHT YARN ARCHITECTURE

¹Dr.A.M.Saravanan, ²Dr.K.Loheswaran, ³Dr.G.Naga Rama Devi, ⁴Karuppathal R,
⁵Dr C Balakrishnan, ⁶T.Aditya Sai Srinivas

¹Assistant Professor of Computer Science, PG & Research Department of Computer Science, Muthurangam Government Arts College (Autonomous), Otteri Road Vellore - 632 002 Tamil Nadu, saravanan.am@yahoo.com

²Associate Professor, Department of Computer Science and Engineering, CMR College of Engineering & Technology, Kandlakoya, Medchal Road, Hyderabad-501401, loheswaran.k@gmail.com

³Professor, Department of Computer Science and Engineering, CMR Institute of Technology Medchal, Kandlakoya, Hyderabad-501401, ramadevi.abap@cmritonline.ac.in

⁴Professor, Department of Information Technology, PSNA College of Engineering and Technology,Dindigul-624622, karuppathal@psnacet.edu.in

⁵Assistant Professor, Alagappa Institute of Skill Development, Alagappa University, Karaikudi – 63003, balasjc@gmail.com

⁶Assistant Professor, Department of Computer Science and Engineering, G. Pullaiah College of Engineering and Technology, Kurnool-518002, taditya1033@gmail.com

Abstract:

Increasing of humanity and development of Internet resources, storage size is growing with each day, whereby digital records are accessible in clouds of an exploratory format. The immediate future of Big Data is coming shortly for almost all other sectors. Big data can aid in the metamorphosis of significant company operations by offering a recommended and reliable overview of available data. Big data has also figured prominently in the detection of violence. Present framework for designing Big data implementations is capable of processing vast quantities of data through Big data analytics using collections of computing devices together to execute complex processing. Furthermore, existing technologies have not been built to fulfil the specifications of time-critical application areas and are far more oriented on real applications than on time-critical ones. This paper proposes the lightweight architecture called Yet Another Resource Negotiator (YARN), which focuses on the concept of a time-critical big-data system from the perspective of specifications and analyses the essential principles of several common big-data implementations. YARN as the normal computational framework to help MapReduce and another application instances within that Hadoop cluster. YARN requires multiple programs to execute concurrently on a constitutive common server and assent programs to delegate services depending on need. The final evaluation is accompanied by problems stemming from infrastructure and services that serve applications, recommend framework and provide preliminary efficiency behaviours that often contribute system impacts to implementation reliability.

Keywords: YARN Architecture, MapReduce, lightweight architecture, Task Trackers, Big Data

Introduction:

Latest developments in computer science apply to big data structures that often represent massive and complicated data-centric programmes that could be operate properly using present information resources [1-6]. Big Data systems is rapidly being a trend-setting activity which really produces an immense volume of information and offers a fresh platform that allows to make appropriate decisions. Breakthroughs in Big Data Analytics offer a new framework and applications for massive data repositories, processing, and innovative technology [7]. The big dataaffords a holistic interpretation of large-scale data creation and perspectives into how that can potentially generate utility for both the organization and the consumer.

Big data systems [8-9] could also appeal to technologies which conduct any form of an optimization that retrieve useful knowledge from the data, explore different patterns, predict market dynamics, or catch criminals. Big data constraints are distinguished by the presence of an immense quantities of content that ought to be analysed to accomplish a target. In predefined cases, the Internet, which offers cost-effective storage of large information analytical power technology through the cloud computing topology might be used as a cost-effective solution.

The fields of big data applications range across aerospace to healthcare [10-11] and, relying on the deployment environment, the functional and non-functional issues vary accordingly, which influences both the technical requirements and the implementation of big data technologies.

Time-critical systems pertain to devices applicable to such perceptual restraints, that further normally composed of optimum time constraints for an input event to be analysed and or an outcome to be obtained. These optimum timelines are distinguished from the properties of a physical conditions which enforces physical standards on application fields. Processing times might be greater effectively in time-critical systems in the order of milliseconds or microseconds [12-13]. Standard time-critical applications had already gained enormously with a variety of general-purpose algorithmic technologies, along with precise optimization models which use implementation interpretation to quantify, a priori, the optimum positional accuracy [14-18].

Figure 1: Time-critical Big data applications

Figure 1 demonstrates the architecture for time-critical Big data applications which are capable of performing time-critical analytics [19]. These analyses are facilitated by an architecture capable of running numerous programming heuristics on a wide network of computers, configured as clusters.

Hadoop provides solutions to the problem of structured and notification knowledge archives. Hadoop is not suggested for small datasets in addition to the fact that there's very little utilisation of circulated slight analysis in more categories of individuals or that it would often demand more development and expense. Hadoop has two sections for HDFS and MapReduce. The Hadoop Record Framework offers a gateway in between autonomous machine and the client's implementations. HDFS is conscientious of big asset distribution and constructive analysis within the Hadoop [20].

The main significance of the proposed architecture called Yet another resource negotiator (YARN) is to focus on Hadoop Optimizing Engineering and is called H2Hadoop. The H2Hadoop is responsible for detaching the immense flexibility of JT into two different sections in the spite of the fact whether JT has more volume or incremental payload in the MRv1. The proposed H2Hadoop (YARN) methodology is organised to be blame-tolerant, that further presumes that perhaps the endeavour will be accurately completed, and that certain work will be carried out. At the point whenever the density of the communities increases the behaviour or assessment dissatisfaction seeks to maintain up that essentially reduces the operating time. The series in the demand for statistical knowledge is comprised of four main features being used for the representation of Big Data.

This paper is structured as follows: section II explains the relevant research methodologies and summarizes the reviews.

Section III explains the number of various issues and technical challenges. Section IV presents the proposed YARN architecture which are equipped with time-critical applications of Big

data. Section V demonstrates the evaluation results and section VI summarizes the results and upcoming work.

II. Related works

In Hadoop release 1.0, also alluded to as MRV1, MapReduce implemented both computing and resource management functions. It constituted of a Job Tracker, as a sole master. Job Tracker dedicated money carried out arranging and tracked handling tasks. It allocated a map and reduced assignments to a set of hierarchical process known Task Trackers. Project Trackers have regularly posted their improvement to the Job Tracker.

Using massive data from SAP is one way to operate, practice and survive. It's the way to be consistently learned along by capturing and analysing of the indications within the computerised commotion. Huge information end-up included essentially any aspect of life from retail and detectors to medical care and natural recharge.

The functional limitations of this kind of model are met by a cluster of 5000 nodes and 40,000 processes functioning simultaneously. Besides this constraint, the usage of computing resources in MRV1 is wasteful. Also, the Hadoop architecture was constrained to the MapReduce computing model. To solve all these problems, Yahoo and Hortonworks released YARN in Hadoop version 2.0 in 2012. The fundamental concept underlying YARN is to ease MapReduce by carrying away responsibilities for inventory control and task scheduling. YARN began offering Hadoop the opportunity to execute non-MapReduce jobs inside the Hadoop system.

The YARN framework has disparate functionality for managing distinct operations and it solves the Hadoop1 or MapReduce shortfall. YARN has a centralised resource provider that manages resources and distributes resources on request. The Yarn is proposed to overcome the downside about too much strain on Task Tracker in Hadoop 1[21]. Yarn also endorses the multi-tenancy strategy. YARN introduces more common interfaces to execute non-Hadoop jobs within the Hadoop framework.

R.S. Raj and G. P. Raju addressed plenty of the community developing designs and leading most of the valuable contraptions for Apache Hadoop. In spite of the effectiveness of Hadoop in splintering and restoring huge data, its efficiency can be related to non-stop activities and scattered knowledge [22]. Their analysis found that Hadoop follows the computing paradigm of MapReduce that is used for the implementation of endeavours in the decentralized context.

C. Yang, C. Yen, C. Tan, S. R. Chafe,[23] based on the need for equivalent bookings, to the degree that they're being done on a distinctive day and generation beyond the boundaries of the client. About the same way, the two functionalities are constrained and fair that are liable for locating the supplier of machinery of multiple allegiances.

Longbin, et al [24] addressed the constraints of Hadoop when a substantial majority of obligations exist within a solitary system. This creates problems of flexibility in communities where Job Tracker needed to monitor a huge proportion of Task Trackers, function flexibility and facilitate implementation, and minimise workload. The study showed how YARN has advanced as Hadoop's demonstrate to monitor and display activity, sustain a cross platform and conduct security requirements.

III. Technical issues and Challenges

The solution to overcome time-critical big-data applications is to examine the infrastructures already presented to professionals, researching the technological problems that emerge from attempting to develop time-critical implementations employing particular technologies. In some circumstances, these technologies have never been sufficiently adapted to the conditions of time-critical applications, contributing to the emergence of advanced regulations for the execution of such methodologies. Both significant development mechanisms focused on Hadoop[25] and map-reduce aim HPC platforms and do not embrace the concept of specifying time constraints. Although in some limited cases, such as the Apache Storm decentralized stream processor[26], do they appear to target online computing. However, because in these situations, most infrastructure projects ought to be based on complex models but instead on time-critical reliability.

Many large-data developments are decentralized technologies in which a variety of associated servers exchange knowledge and insights to carry out a task, usually to undertake large data analysis. As a result, several infrastructural facilities are capable of spreading processing between various cluster members, leveraging various guidelines. Besides that, in most situations, the techniques encountered do not incorporate the time-critical complexities of the operation and generate settings that are non-optimal mostly from point of view of the task.

Remote availability of information continues to result in a reaction points deduction of over an enormous margin in certain big data implementations, in terms of an increase jitter [27]. Remote networking, furthermore, offers the solution for concurrent and shared processing to mitigating the risks amount of time that can be considered significantly decrease the worst-case responsiveness of the submission.

With respect to infrastructure inadequacies, still exists to be a major shortfall in time-critical big data applications. Present prevailing equipment for massive information were being developed for development purposes and are ambiguous on the discovery of particular enhancements for time-critical systems. These techniques must also be generalised with the time-critical properties needed for the creation of time-critical big data systems.

IV YARN Architecture with its components and time-critical applications based big data system

The Yarn connotes a forum for the creation and implementation of decentralized computing systems. Perhaps it improves performance and the capacity to share resources. The Yarn framework is planned in figure 2 to accommodate more data[26] computing templates including the Apache Giraph, Apache HAMA, Apache Spark, Apache Wind, and several others than MapReduce and Hadoop 1. Our Yarn architecture follows a master-slave modelling approach in which the ResourceManager is the master and node-specific slave NodeManager (NM). Global ResourceManager and NodeManager develop a far more flexible, standard and simple framework for remote performance optimization. As per Yarn design and architecture, the ResourceManager program is administered on the network master node. The Yarn user submits a request to the ResourceManager. The task may be a single MapReduce work, a cyclic job graph, a Java application, or another shell script. The user also discusses the ApplicationMaster and the order to launch the ApplicationMaster on a server. The programme manager managing of the system administrator will verify and approve the request from its customer.

During that resource manager configuration process, the ApplicationMaster container would be allocated to the server and the NodeManager administration will employ the instruction to launch the ApplicationMaster administration [28]. Each Yarn programme has a distinguishable container called the ApplicationMaster. The ApplicationMaster Container is the first pool of the Latest NodeManager server application operating on any slave in the Yarn cluster. It is responsible for overseeing the containers of the programme. The services defined for the container are obtained from the resources of the NodeManager. At specified periods, each NodeManager upgrades the ResourceManager for a collection of available resources.

Figure 2: YARN architecture for Time-critical based Big data applications

Resource Manager:

The Resource Manager allocates all accessible clustered services and therefore controls the clustered systems operating on the Yarn machine. Resource Manager serves as a significant resource scheduler that is liable for resource control and organizing in compliance with the ApplicationMaster's order for the resource requirements of the application. It is responsible for the stock of comfortable tools and operates a wide range of essential utilities, the most important however is the scheduler. The Scheduler portion of the Yarn ResourceManager delivers services for the execution of requests.

Resource Tracker Service:

Early Validation ensures that the Node Manager signals to the Resource Manager that while it is possible to accept requests and to offer suggestions on its Processor cores and memory capacity. In particular, dual forms of resources are offered, initially, to remind the Resource Manager of its welfare via the Resource Tracker Program and, furthermore, to apprise the Node Manager of the current condition of the repositories at site. This knowledge is used by the

optimizer to change its broad perspective of the cluster resources in such a manner that everything effectively disburses themselves to future applications from the Application Master.

Application master service:

The Master Application Service is responsible for the method of importing of Yarn resources per application platform. It is also responsible for ensuring the usage of services for a runtime environment with the Resource Manager. Application Masters requests are processed by the Application Master Provider. All requests are in the standard layout and which comprise details including the quantity of containers[10] required, processor and Memory capacity per container, location option and requirement preference from inside the request.

Scheduler:

The scheduler should be responsible for determining that activities are to be carried out and when and when to be carried out. The Scheduler is the prime requisite of the Resource Manager. The scheduling algorithm has a plug-in approach that is responsible for segmenting network infrastructure across several queues, processes, etc. Scheduler is accountable for the allocation of activities which is not concerned with the management of progress and the investigation of services.

NodeManager:

NodeManager operates as a per-machine administrator and is responsible for controlling the lifespan of the device and tracking the usage of its services. It deals more with the RM and the AM and distributes the state of currently operating containers and allocated memory and Processor services on its computer by comprehending heartbeats to the RM. It is therefore responsible for the closure of containers on the grounds of a request submitted by either the RM or the AM. NodeManager is a node server application that is responsible for the enforcement of node-based containers.

Application Master:

The Application Master shall be at the per-application stage and shall be responsible for the control of the lifespan of the request and shall pay attention on the necessary tools of the dispatcher, their performance and the tracking of development. Yarn serves as a third-party repository receptive to consort tools from the ResourceManager Scheduler and operates with NodeManager to execute operations. It also demands the distribution of services from the planner and sends its reputation back to it at predetermined times. It is responsible for talking to the Resource Manager relates to appropriate resource bins, recording their location and measuring performance.

Container:

A container is a feature of the Yarn atomic commodity. A container can be acknowledged as a pragmatic reserve of resources to be used by the activity assigned in that container. At the primitive level, the container is an aggregation of hardware facilities including CPU cores, RAM,[18] and discs on a single node. The container in YARN is described about how the job

component becomes in the context of a job. Jobs are devoid of jobs, but every work is performed in a container with a fixed number of resource utilization. Users can recognise the specifications of the container commodity so if clients approve tasks to [19] RM and implement such a myriad of features.

Client and Admin interfaces:

Yarn endows both the database and the administrator command prompt software. For the identification of yarn elements, there is the REST API and also the MBeans for the registry mechanism

Time Critical big data systems:

Figure 3: Architecture for Time-Critical Big-Data

The architecture introduces a multidisciplinary architecture to big data, merging conventional time-critical middleware[30] with big data architecture [29]. Our conceptual template comprises of four primary layers, including one that deals with various facets of the big data configuration. Applications that are considered analytic tools are at the forefront of the design. They are structured as a directed acyclic graph structure, facilitating implementation in a wide collection of servers. The key prerequisite for this level is the presence of time-critical specifications. Standard criteria are timelines ranging from sub-seconds to seconds, days and weeks. These analytical targets are necessary in order to decide the quantity of computers needed to conduct the analysis. Enhancing software are instruments that describe another round of layout. These methods fulfil the various facets of an empirical method. In the special case of a time-critical device, these provide time-limit assistance and any other form of specifications. To meet objectives, tools provide precise control of the available resources to computers that need to be carefully managed and programmed. The characteristics of resources

that are attempted involve disc throughput, memory, Core processor, and communication services.

Figure 4: framework for generation of time-critical evaluation

The design defined in Figure 4 was being formalised leveraging real-time conceptual framework to generate time-critical evaluation, which is beneficial for mapping-reducing and decentralized data aggregation. The nature of the methodology resides in the separation of various phases in systems which have been programmed specifically with some other operations so this communicate with most other concurrent (||) and simultaneous phases(->).

Through a structured point of view, the time-critical big-data platform (TC BDS) comprises a series of n analysis (TC Ai):

$$TC_BDS \triangleq (TC_A_1, \dots, TC_A_n) \quad (4.1)$$

Where each analytic (TC Ai) is expressed by the implementation graph (TC DAGi) and the reliability criteria set (TC RQi):

$$TC_A_i \triangleq (TC_RQ_i; TC_DAG_i) \quad (4.2)$$

The simple prerequisite is a time limit for the entire study (D). Time limits are described by analytics and may involve training, writing, blocking of information. Similarly, each direct acyclic graph (DAGi) consists of a series of levels (i). At each phase, the description is determined by a limited inter-arrival time (Ti j), a partial time limit for each step (Di j) and a worst-case implementation (Ci j) aligned with the point.

$$TC_DAG_i = \begin{pmatrix} [T_i^1, \dots, T_i^{stages(i)}] \\ [D_i^1, \dots, D_i^{stages(i)}] \\ [C_i^1, \dots, C_i^{stages(i)}] \end{pmatrix} \quad (4.3)$$

Every device provides a standardised maximum usage (Uk) of the device and also imposes a cumulative blocking time (Bk), that usually influences the programme and corresponds to the time duration the implementation might be anticipating a resource.

$$TC_CLS \stackrel{\text{def}}{=} (\pi_1, \dots, \pi_m) \quad (4.4)$$

$$\pi_k \stackrel{\text{def}}{=} (U_k, B_k) \quad (4.5)$$

In order to be better defined, each phase of the request should indeed be allocated to a cluster system ($\Delta_i j$). In order to support the programme, the device must announce a priority ($P_i j$) to be honoured in all endpoints. In turn, when selecting a node, the process analytic often struggles a blockage ($B_i j$) from infrastructure.

$$TC_DAC_i = \begin{pmatrix} [T_i^1, \dots, T_i^{stages(i)}] \\ [D_i^1, \dots, D_i^{stages(i)}] \\ [C_i^1, \dots, C_i^{stages(i)}] \\ [\pi_i^1, \dots, \pi_i^{stages(i)}] \\ [P_i^1, \dots, P_i^{stages(i)}] \\ [B_i^1, \dots, B_i^{stages(i)}] \end{pmatrix} \quad (4.6)$$

Blocking (B_i) decreases the output of the request, which is improved by blocking (see Eq. 8). They may not, furthermore, have an effect on the number of services of a system dependent on consumption, so it just imposes an interruption. Some other fascinating consequence of the methodology would be that a secure limit may also be derived for the highest amount of processing elements (m) needed for the implementation of the method. As seen in the format, the report focuses on the limited = application time and the optimum implementation costs.

V. Results and Discussion:

The basic aims of the objective scientific portion are as follows:

- Determine the quantity of nodes needed to fulfil those requirements falsifiable. That it will be meant to have output trends.
- Equate the method with each other, in situations where it is conceivable.

Figure 5: Time results

Figure 6: Speed results

Figure 7: Efficiency

Time: The total time taken to execute each analysis is proportional to the quantity of knowledge possessed. It also reduces as the volume of cores improves (Figure 5). To reach the time limit, the device needs 20 cores: with 16 cores, it allows 2.2 hours to determine the maximum document; and with 32 processors, it involves 1.6 hours.

Volume: The speed, expressed in figure 6 as the number of tweets per second, declines as the number of processors. It also depends on the information gathered however the key constraint is the number of cores accessible for processing information.

Efficiency: The efficiency determined by the speed calculated by the total of cores needed to execute the device improves as the number of components declines (figure 7).

Conclusion:

Recently, there is an increasing need for computer passive components designed to process unimaginable streams of information. Big data applies to any information that originates around the worldwide at an unparalleled pace, since this information may be whether organised or unorganised. YARN technique enables Hadoop to establish enterprise-level options, consequentialist organisations in state-preferred resources development. Many obstacles lie beneath of us in the time-critical big-data frontier of delivering a standardized infrastructure capable of reaching diverse analytical requirements in a consistent manner. This paper suggests the forum for the development of cooperative technologies, integrated encryption and data processing across the full scope of the Hadoop cluster. The YARN is a totally rewritten Hadoop cluster framework. Hadoop YARN handles the capital very professionally. On demand, it dispenses the same with every request. Yarn provides a strong edge in performance, robustness and stability that varies from the classic MapReduce engine in the original implementation of Hadoop. YARN distributes cluster services in a complex and effective way, allowing usage of them is even easier than the previous iterations of Hadoop.

Declaration

Funding

Not Applicable

Conflicts of interest/Competing interests.

The authors declare that they have no conflict of interest.

Availability of data and material

Not Applicable

Code availability

Not Applicable

REFERENCES:

- [1] V. N. Gudivada, R. Baeza-Yates, V. V. Raghavan "Big Data: Promises and Problems". Computer, 48(3) pp.20-23, (2015).
- [2] N. Marz, J. Warren "Big Data: Principles and best practices of scalable real-time data systems". Manning Publications Co., (2015).
- [3] H. V. Jagadish et al. "Big data and its technical challenges". Communications of the ACM 57(7) pp.86-94, (2014).
- [4] M. Chen, M. Shiwen, and L. Yunhao "Big data: A survey". Mobile Networks and Applications 19(2), pp.171-209, (2014).
- [5] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. Byers "Big data: The next frontier for innovation, competition, and productivity" (2011).
- [6] J. Forsyth, L. Boucher "Why Big Data Is Not Enough". Research World 2015. 50, pp. 26-27 (2015).
- [7] Dr. Yusuf Perwej, "An Experiential Study of the Big Data," for published in the International Transaction of Electrical and Computer Engineers System (ITECES), USA, ISSN (Print): 2373-1273 ISSN (Online): 4(1), pp. 14-25, (2017).
- [8] Apache Accumulo. <http://accumulo.apache.org>
- [9] T. C. Bressoud, Q. Tang, Analysis modeling and simulation of Hadoop YARN MapReduce.
- [10] M. Chen, M. Shiwen, and L. Yunhao "Big data: A survey". Mobile Networks and Applications 19(2), pp. 171-209, (2014).
- [11] H. V. Jagadish et al. "Big data and its technical challenges". Communications of the ACM 57(7), pp.86-94, (2014).

- [12] Dong, Xin Luna, and Divesh Srivastava. "Big data integration." Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, (2013).
- [13] Chen, Hsinchun, Roger HL Chiang, and Veda C. Storey. "Business Intelligence and Analytics: From Big Data to Big Impact." MIS quarterly 36(4), pp.1165-1188, (2012).
- [14] Yuxiao Dong, Reid Johnson, Nitesh Chawla: "Can Scientific Impact Be Predicted?", IEEE Transactions on Big Data, 1, pp. 1, (2016)
- [15] Jules White, Brian Dougherty, Richard E. Schantz, Douglas C. Schmidt, Adam A. Porter, Angelo Corsaro: "R&D challenges and solutions for highly complex distributed systems: a middleware perspective". J. Internet Services and Applications 3(1), pp. 5-13 (2012).
- [16] Sha, Lui, et al. "Real time scheduling theory: A historical perspective." Real-time systems 28.2-3 (2004): 101-155.
- [17] Kopetz, Hermann. Real-time systems: design principles for distributed embedded applications. Springer Science & Business Media, (2011).
- [18] Davis, Robert I., and Alan Burns. "A survey of hard real-time scheduling for multiprocessor systems." ACM Computing Surveys (CSUR) 43(4), pp.35, (2011).
- [19] P. Basanta-Val, N. C. Audsley, A. J. Wellings, I. Gray, N. Fernández,"Architecting Time-Critical Big-Data Systems", preprint (2020).
- [20] Dr. Yusuf Perwej, "An Experiential Study of the Big Data," for published in the International Transaction of Electrical and Computer Engineers System (ITECES), USA, 4(1) , pp. 14-25, (2017).
- [21] T. C. Bressoud, Q. Tang, "Results of a Model for Hadoop YARN MapReduce Tasks", IEEE International Conference on Cluster Com-nutine, (2016).
- [22] R. S. Raj and G. P. Raju, "An approach for optimization of resource management in Hadoop", In: proc. of International Conf. on Computer and Communications Technologies (ICCCT), pp. 4532-4544, (2014).
- [23] C. Yang, C. Yen, C. Tan, and S. R. Madden, "Osprey: Implementing mapreduce-style fault tolerance in a shared-nothing distributed database", In Proc. of International Conf. on Data Eng pp. 657–668, (2010).
- [24] L. Longbin, et al. ShmStreaming, "A Shared Memory Approach for Improving Hadoop Streaming Performance," In: proc. of International Conf. on Advanced Information Networking and Applications (AINA), pp.12-23, (2013).
- [25] K. Shvachko, H. Kuang, S. Radia, R. Chansler. "The hadoop distributed file system". In Mass Storage Systems and Technologies (MSST), IEEE 26th Symposium on pp. 1-10 (2010).
- [26] Storm. "Distributed and fault-tolerant real-time computation". Available on <https://storm.incubator.apache.org/>, (2014).
- [27] P. Basanta-Val, M. García-Valls."Resource management policies for real-time Java remote invocations". J. Parallel Distrib. Comput. 74(1), pp.1930-1944 (2014).
- [28] Yusuf Perwej, BedineKerim, Mohmed Sirelkhtem, "An Empirical Exploration of the Yarn in Big Data",International Journal of Applied Information Systems (IJAIS),12(9),pp.19-29,(2017).

[29] Sha, Lui, et al. "Real time scheduling theory: A historical perspective." *Real-time systems* 28.2(2), pp. 101-155, (2014).

[30] Jules White, Brian Dougherty, Richard E. Schantz, Douglas C. Schmidt, Adam A. Porter, Angelo Corsaro: "R&D challenges and solutions for highly complex distributed systems: a middleware perspective". *J. Internet Services and Applications* 3(1), pp. 5-13 (2012).

Figures

Figure 1

Time-critical Big data applications

Figure 2

YARN architecture for Time-critical based Big data applications

Figure 3

Architecture for Time-Critical Big-Data

Figure 4

Framework for generation of time-critical evaluation

Figure 5

Time results

Figure 6

Speed results

Figure 7

Efficiency