

RRDTool: A Round Robin Database for Network Monitoring

¹Manmohan Singh, ²Sweta Dargada, ³Hemant Mewada, ⁴Meenu Tahilyani, ⁵Jitendra Malviya, ⁶Romsha Sharma and ⁷Shiv Shakti Shrivastava

¹Department of Computer Science Engineering, IES College of Technology, Bhopal, India

²Department of Computer Science and Engineering, Symbiosis Skills and Professional University, Pune, India

³Department of Computer Science, CSA Govt. P.G Lead College, Sehore, India

⁴Department of Computer Science, Sant Hirdaram Girls College, Bhopal, India

⁵Department of Life Sciences and Biological Sciences, IES University, Bhopal, India

⁶Department of Computer Science, Career College, Bhopal, India

⁷Department of Computer Science Engineering, RNTU University, Bhopal, India

Article history

Received: 24-04-2022

Revised: 25-06-2022

Accepted: 08-07-2022

Corresponding Author:

Manmohan Singh
Department of Computer
Science Engineering, IES
College of Technology Bhopal,
India
Email: kumar.manmohan4@gmail.com

Abstract: RRD tool is a circular database that stores time series data. RRD tool assumes time-variable data in intervals of a certain length. An interval is named as a step, and specified upon creation of an RRD file which cannot be changed afterward. Because data may not always come at just the right time, RRD tool will automatically interpolate any submitted data. This is done to fit its internal timestamps. If we compare linear databases with RRD then we can say that RRD is non-linear like a circular queue. Usually, in a Round-Robin Database (RRD), time-series data like network bandwidth, memory usage, and CPU load are stored. The data is stored in a non-linear way so that the system storage footprint remains constant over time. This avoids resource-expensive purge jobs which reduce complexity. MySQL does not provide any such kind of storage engine. RRD tool has also the ability to create and feed this database. RRD tool stores data; that makes it a back-end tool. The RRD tool command set allows the creation of graphs which makes it a front-end tool as well. Other databases just store data and cannot create graphs. RRD combined with NET-SNMP can collect data and using scripts we can generate beautiful real-time graphs which can be used for monitoring computer networks.

Keywords: Network Monitoring, RRD Tool, NET SNMP, Computer Network, Graphing, Open-Source, Database, Network Management

Introduction

RRD tool (round-robin database tool) handles time-series data like temperatures, CPU load, network bandwidth, etc. The data is stored in a round-robin database which acts as a circular buffer; thus, the system storage footprint remains constant over time (Shafi *et al.*, 2020). RRD tool is an opensource tool, and it can be integrated into Shell scripts, Python, Ruby, Perl, Lua, or Tcl applications easily. RRD tool assumes time-variable data in intervals of a certain length. This interval, usually named step, is specified upon the creation of an RRD file and cannot be changed later (Fiad *et al.*, 2020).

RRD tool interpolates data to get the time steps. PDP stands for Primary Data Point. It is the value of a specific step that is interpolated. Consolidation Function CF is used to plot Multiple PDPs. This plotting of PDPs forms a Consolidated Data Point (CDP). Typical consolidation functions are

average, minimum, and maximum (Ali *et al.*, 2020). Round Robin Archive (RRA) stores the Consolidated Data Points (CDP).

Total time covered by RRA:

$$\text{Time Covered} = (\#CDPS \text{ Stored}) _ (\#PDPs \text{ Per CDP}) _ \text{Steps}$$

In a Round-Robin Database (RRD) usually, time-series data like network bandwidth, temperatures, CPU load, etc., is stored. To avoid resource-expensive purge jobs and reduces complexity data is stored to form a constant footprint of data (Runsungnoenand Anusas-amornkul, 2020). MySQL doesn't provide this kind of storage facility. Although some people were thinking about it and some prototypes exist. RRD tool re-samples the data and stores this value. Network Monitoring tools gather data after some specific time. But the data is not available all the time because of some query lost

or device down. Data does not arrive exactly in that interval. It takes the actual arrival time under the curve and creates new points in time between these time intervals and makes a curve beneath the real curve (Omotehinwa *et al.*, 2019; Olofintuyi *et al.*, 2019; Rad *et al.*, 2017; Zaidi and Shukla, 2018). Thus traffic latency will be the same but the numbers shown may differ.

Background and Related Work

In recent years network developers have used relational databases for storing network data persistently (Preeth *et al.*, 2015). Data with uniform characteristics are organized in tables linked by relationships table monitoring. Each table is logically divided into columns and rows wise, where a row is uniquely identified using a primary key (Zeng and Wang, 2009). Data stored in the database can be modified and deleted. Unfortunately, network data is not characterized by among relationships, it is usually unchangeable and the same data is repeated over time at every measurement interval or after some time, making relational databases not convenient for handling this type of data (Zeng and Wang, 2009).

- Every measurement interval is populated with fresh data that increases cardinality. The consequence is that cardinality as well the space taken on disk increases with the number of measurements continues Perkins and McGinnis (1997)
- As soon as indexes become large enough to prevent themselves to be cached, data retrieval becomes significantly slow (Zeng and Wang, 2009; Xiaogao, 2007) thus jeopardizing the performance of applications sitting on top of the database

A partial solution for avoiding these slowdowns is the use of Binary Large Objects (BLOB) for storing time series. In this case, the drawback is that the database is unable to search directly on blobs. As it can just read/store raw data, it must delegate to third-party applications the implementation of such retrieval, and data management facilities. To address these issues with relational databases, time series database servers (Zukowski *et al.*, 2006) have been created. They have been designed for enabling efficient data retrieval within some defined date/time ranges of the system, as well as for handling date and timezone conversions. Mostly used in the industry, and the only open-source alternative (Abadi *et al.*, 2006) has a pretty complex architecture of database managed with several components interacting over a network.

Acquisition, Consolidation, and Archival

While the man pages talk of command line switches you have to set to make the RRD tool work it is

important to note that the RRD tool can be remotely controlled through a set of pipes. This saves a considerable amount of startup time when we plan to make the RRD tool do a lot of things quickly. We need to check the section on Remote Control further down (Zeng and Wang, 2009).

Data Acquisition

When monitoring the state of a system, it is convenient to have the data available at a constant time interval. Unfortunately, we may not always be able to fetch data exactly at the time we want (Peng *et al.*, 2008). RRD tool can update the log file when the user specifies. RRD tool will interpolate the data from the Data Source and write it in the log. Mauro and Schmidt (2005).

Consolidation Function

We may log data at a 1 min interval, but what to do if we want the data for a whole year? For this, we might simply store the data in a fixed interval but this will take up so much disk space. Time to interpolate such large data will also be more. Using CF this problem is solved. We can define interval and Consolidation function which could be average, minimum, maximum, and total, last to build our RRD (Xiaogao, 2007).

Round Robin Archives (RRA)

Round Robin Archives (RRA) stored Data values of the same consolidation function Rose and McCloghrie (1990) If we like to store some 100 values in some 2 min intervals, RRD tool will allocate the space accordingly. There is a header stored that stores the information about slots new values are written to the round robin archive in a round robin manner (Zeng and Wang, 2009). This will automatically limit the history to the last 100 values. Using RRA we ensure that the RRD does not grow large and old data is eliminated. The consolidation feature will help us to keep data for a very long time. Mocerri (2006). We can use different CF for different requirements like the minimum Traffic on the Lan, the maximum temperature of the nuclear reactor, the total min of downtime, etc. Unknown RRD stores data at a constant interval. Sometimes no new data is available, so with RRDtool, we can handle such situations by storing an "unknown" value into the database. When consolidating a dataset, the amount of "unknown" data values is accounted for and when a new consolidated value is ready to be written to its Round Robin Archive (RRA) (Fiad *et al.*, 2020), a validity check is performed to make sure that the percentage of unknown values in the data point is above a configurable level. If not, an "unknown" value will be written to the RRA Chaparadza (2005).

Plotting and Graphing

RRD tool allows us to consolidate data as it becomes older. RRD tool stores data in a binary file. It has a dump function, so the data could be dumped in an xml file. Beautiful graphs could be built using some of the following. Functions RRD tool allows reports in numerical and graphical format. The graphing feature is fully configurable. The size, color, and contents of the graph can be defined freely Harrington *et al.* (2002).

In this Fig. 1, the hierarchy formally establishes that any RRD parameter at any Level is a function only of the physical variables at circular queue the same Level or above.

In this Fig. 2 RRD tool is a round robin data management system. It could be used with network monitoring to get graphs of network statistics.

In this Fig. 3, the name RRD is an initialism of round robin database. The “round robin” refers to the fact that the database is always fixed in size, and as a new entry enters the database, the oldest entry is discarded. In practical terms, the database collects data for a fixed period of time, and information that is older than the threshold gets removed.

In this Fig. 4, the structure of an RRD database is different than other linear databases their databases define tables with columns, and many other parameters.

In this Fig. 5, A and B, sometimes users and system administrators have a need to analyze the metrics in more detail of memory utilization of week.

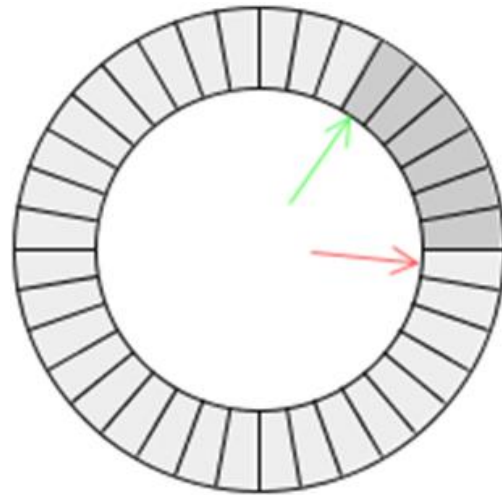


Fig. 1: RRD's circular data structure

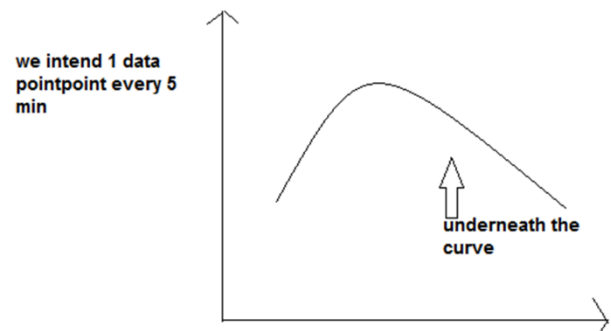


Fig. 2: RRD's data curve

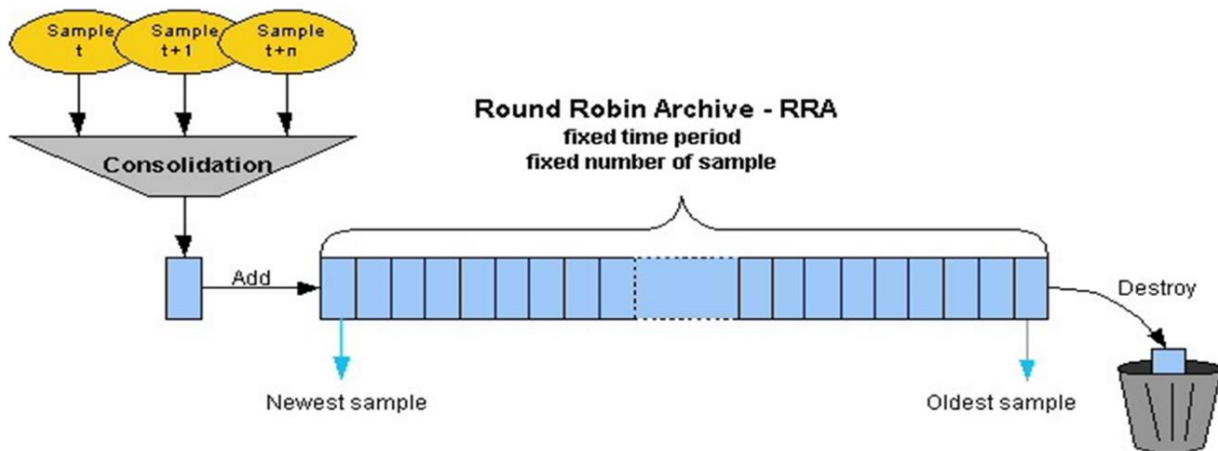


Fig. 3: RRD's Mechanism to plot data points

Sr. No.	Command	Description
1	<code>rrdcreate</code>	Set up a new Round Robin Database (RRD).
2	<code>rrdupdate</code>	Store new data values into an RRD.
3	<code>rrdupdatev</code>	Operationally equivalent to update except for output
4	<code>rrdgraph</code>	Create a graph from data stored in one or several RRDs. Apart from generating graphs, data can also be extracted to <code>stdout</code>
5	<code>rrddump</code>	Dump the contents of an RRD in plain ASCII. In connection with restore, we can use this to move an RRD from one computer architecture to another
6	<code>rrdrestore</code>	Restore an RRD in XML format to a binary RRD.
7	<code>rrdfetch</code>	Get data for a certain time period from a RRD. The graph function uses fetch to retrieve its data from an RRD.

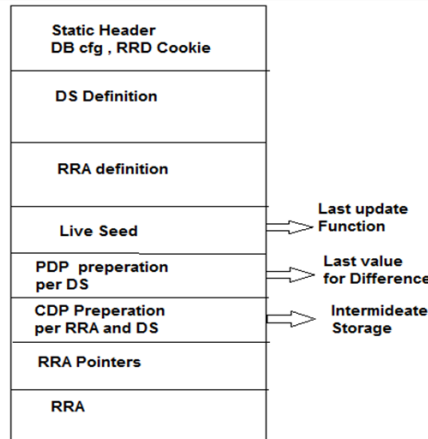
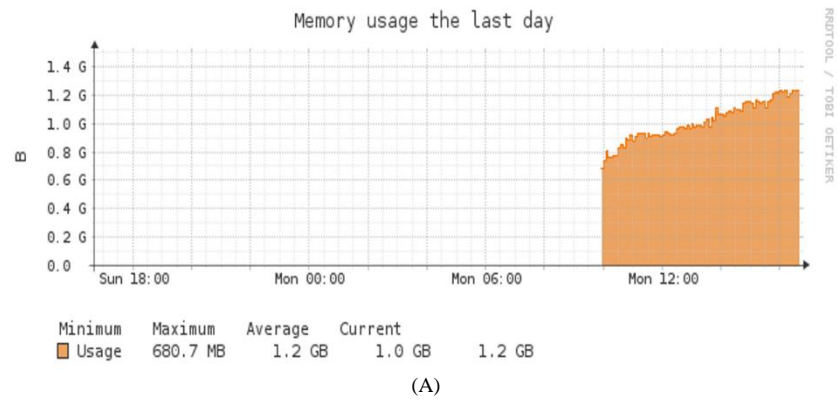
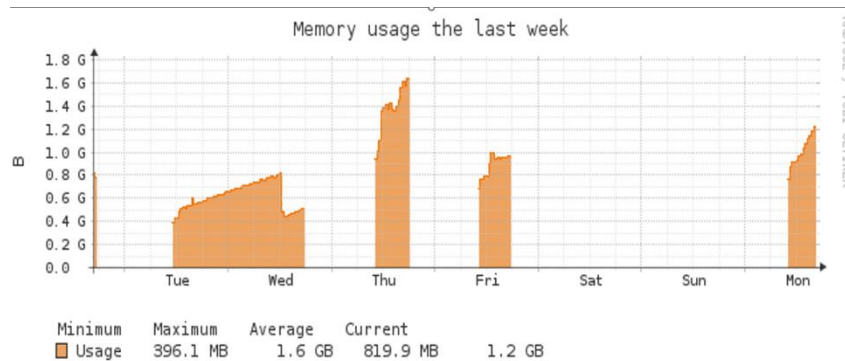


Fig. 4: On disk structure of RRDtool



(A)



(B)

Fig. 5: Memory Utilization, max and min memory usage of last day and last week

In this Fig. 6 A and B, makes it very easy to read metrics stored in the RRD database format. Reading an archive is very quick, and your resulting data is a for an individual archive, or a list of for the entire file in year or months wise.

In this Fig. 7 RRD and draw an area for the "in" traffic and a line for the "out" traffic. View the image and keep logging data for a few more days starting with band width monitoring tools.

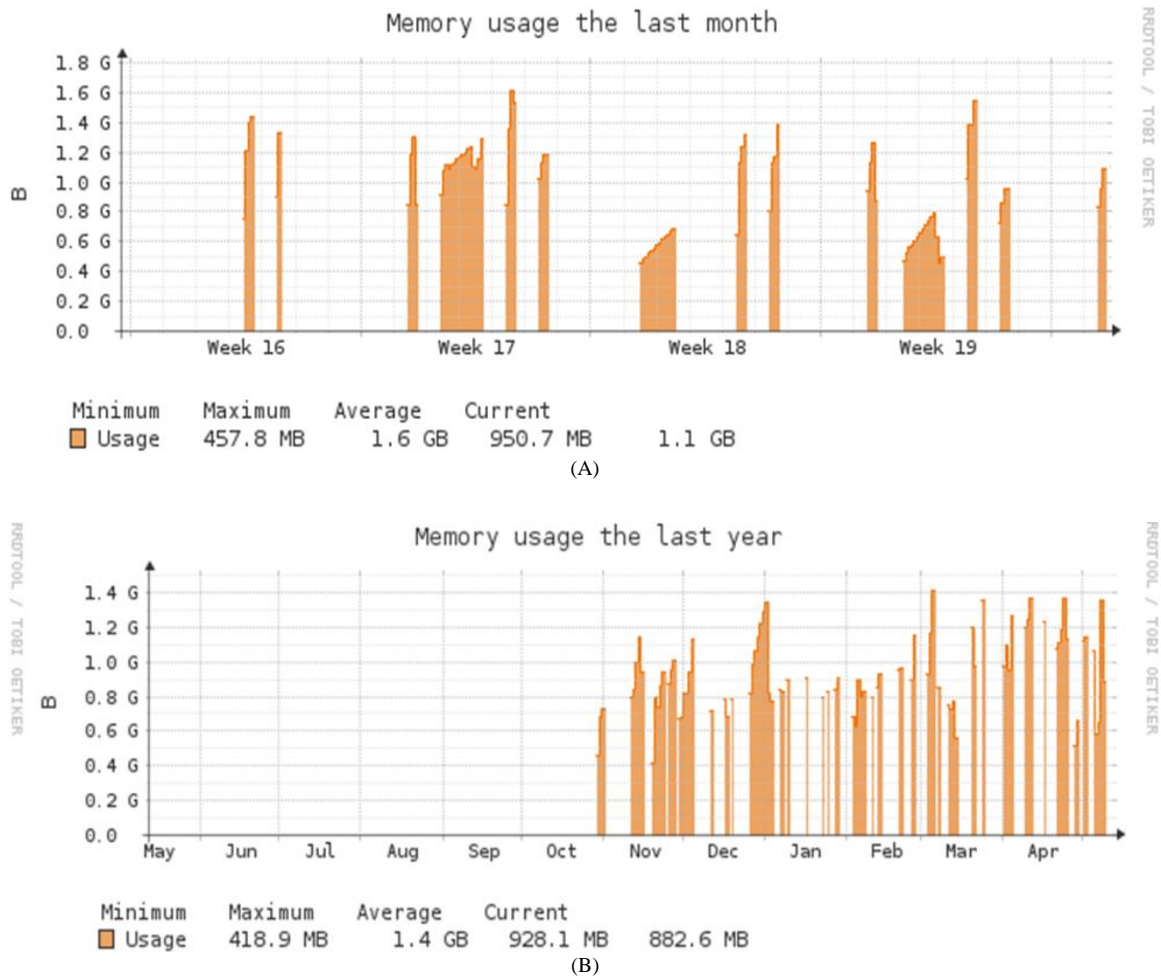


Fig. 6: Interpolated and modified data of memory utilization in the last month and year

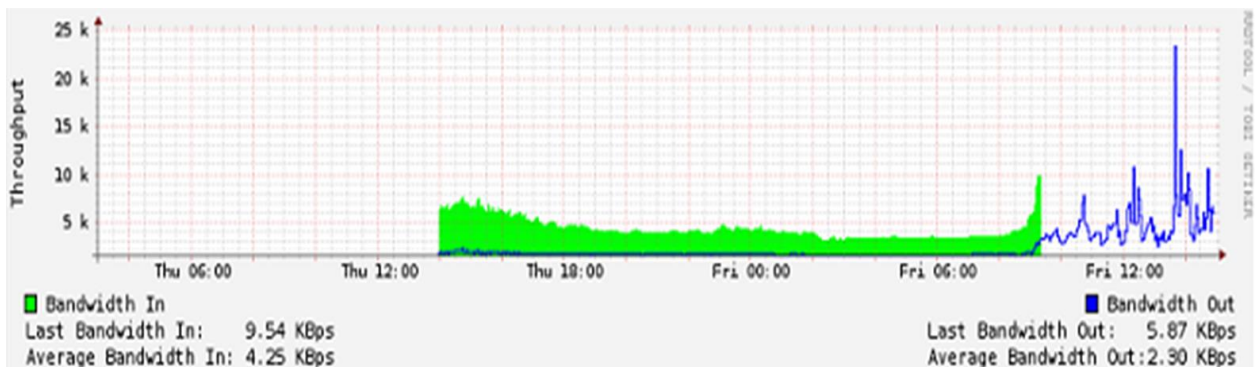


Fig. 7: In/out traffic of a router link

Implementation and Results

Historic information about the network is very important to know the trend of traffic. We can get the base on how the network works in normal and abnormal conditions. We can then compare the change in the network using this baseline. We want to analyze the trend of the network. To estimate the future needs or performance of the network, this metric prepared by the baselining is useful. We can monitor device health and network problems based on the metric. Some proactive measures are required to manage faults and monitor the network.

RRD tool is responsible for generating graphs in layer 3. RRD tool stores the time series data like network bandwidth, temperatures, and CPU load in a circular database. The following steps are used in the account of graphing using the RRD Tool.

Initialize the Database

To create a database we use the `rrdcreate` command.

`Rrdcreate-step60-NDS:pl:GAUGE:120:0:100.`

`DS:rtt:GAUGE:120:0:10000000 RRA:MAX:0.5:1:1500.`

“Step 60”-Then the user has to specify the amount of time after which we want our database to be updated.

“Start N”-this command is used to start the timer Now.

“DS: pl: GAUGE: 120: 0: 100”-DS stands for dataset and PL is Packet Loss. DS says that this is a data set. PL is packet loss. GAUGE is absolute data. 120 is the timeout value in seconds. The accepted value lies between 0 and 100.

“DS: rtt: GAUGE: 120: 0: 10000000”-rtv is "round trip time value". 120 is the timeout time in seconds. The accepted value lies between 0 and 10000000.

“RRA: MAX: 0.5: 1: 1500”-RRA is used for archival of the data points. MAX accepts the maximum value if multiple values are available over multiple "steps". 0.5 is an internal resolution value, and "1" specifies how many steps should be averaged before storing. One step is equal to one database value. 1500 is how many "steps" we will store in the database.

Collect the Data Sets Over Time

We update the rrd after every specified interval. A cron job can be used to run in the background and it will be used to enter the data periodically into the database. We wrote a shell script to update the database from time to time where the cron job was called. It is stored in `crond` as it is running daily.

Create the Graph

We want to create a graph for a graphical representation of the collected data. We also want to run this step using a cron job. RRD tool aims to handle time-series data like network bandwidth, temperatures,

CPU load, etc. So we used a shell script to generate the following graphs using a few shell scripts. The graphs below are the graphs of memory utilization on the workstation. The graphs are daily, weekly, monthly, and yearly memory utilization. These graphs can be saved in PNG format.

The user can get graphs of other information like bandwidth, CPU utilization, and memory utilization of network devices. The user has to just give the IP Address and the community string of his SNMP, if it is SNMP v1 and for SNMP Perkins and McGinnis (1997). V2 and V3 he is supposed to give the Security name, Security level, authentication protocol, authentication key, and encryption key. Later the information on traffic on the interface of the network devices will be logged by the RRD tool.

Conclusion and Future Work

We get the information of the network in a raw form when we are just querying the network devices. RRD tool can be used for storing the statistics received from the querying. A low-cost tool like RRD tool can be used for real-time visualization of network status in a graphical format. We can plot disk usage, memory utilization; spam mails, etc., using RRD Tool in graphical format. The data is stored in a way that the system storage footprint remains constant over time. This avoids resource-expensive purge jobs and reduces complexity. MySQL does yet provide this kind of storage engine. RRD tool stores the resampled version of the data rather than storing it directly.

A round robin database can be used in network monitoring. Using this tool, the gap can be bridged between the network administrator and the problems lying in the network. So many monitoring systems are being implemented using RRD Tool. Open Source and well-known network monitors like CACTI, NAGIOS, PRTG, and MRTG are using it for graphing and storing purposes.

Acknowledgment

To our families and all those who matter in our lives.

Author's Contributions

Manmohan Singh and Sweta Dargada: Carrying out the experiment, collecting and verifying the analyzed data; prepared the draft of the manuscript and approved the final manuscript.

Hemant Mewada: Member of the laboratory experimental/implementation monitoring and approved the field data.

Meenu Tahilyani: Correction of the translation of the manuscript in English, experimental monitoring, member of the laboratory/implementation and approved the field data.

Jitendra Malviya: Member of the laboratory, preparation of the nursery, implementation monitoring, and collection of data.

Romsha Sharma and Shiv Shakti Shrivastava: Designed the research plan, and supervised this study, and approved the final manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

References

- Abadi, D., Madden, S., & Ferreira, M. (2006, June). Integrating compression and execution in column-oriented database systems. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (pp. 671-682).
<https://doi.org/10.1145/1142473.1142548>
- Ali, K. F., Marikal, A., & Kumar, K. A. (2020). A hybrid round-robin scheduling mechanism for process management. *International Journal of Computer Applications*, 975, 8887.
- Chaparadza, R. (2005, November). On designing SNMP-based monitoring systems supporting ubiquitous access and real-time visualization of traffic flow in the network, using low-cost tools. In *2005 13th IEEE International Conference on Networks was Jointly held with the 2005 IEEE 7th Malaysia International Conf on Communic* (Vol. 2, pp. 6-pp). IEEE.
<https://doi.org/10.1109/ICON.2005.1635630>
- Fiad, A., Maaza, Z. M., & Bendoukha, H. (2020). Improved version of round robin scheduling algorithm based on analytic model. *International Journal of Networked and Distributed Computing*, 8(4), 195-202.
<https://doi.org/10.2991/ijnkc.k.200804.001>
- Harrington, D., Presuhn, R., & Wijnen, B. (2002). *An architecture for describing Simple Network Management Protocol (SNMP) management frameworks* (No. rfc3411).
- Mauro, D., & Schmidt, K. (2005). *Essential SNMP: Help for System and Network Administrators*. " O'Reilly Media, Inc."
- Moceri, P. (2006). SNMP and Beyond: A Survey of Network Performance Monitoring Tools.
- Olofintuyi, S., Omotehinwa, T., Oyekanmi, E., & Olajubu, E. (2019). An Improved Time Varying Quantum Round Robin CPU Scheduling Algorithm. *Achievers Journal of Scientific Research*, 2(2), 31-38.
- Omotehinwa, T., Azeez, I. & Olofintuyi, S. (2019). Simplified Improved Dynamic Round Robin CPU Scheduling Algorithm. *International Journal of Information Processing and Communication*. Vol. 7 No. 2, pp. 122140
- Peng, H., Jianlin, Q., & Xiang, G. (2008). Design and implementation of remote host monitoring system based on SNMP. *Computer Engineering and Design*, 29(13), 3303-3312.
- Preeth, E. N., Mulerickal, F. J. P., Paul, B., & Sastri, Y. (2015, November). Evaluation of Docker containers based on hardware utilization. In *2015 international conference on control communication & computing India (ICCC)* (pp. 697-700). IEEE.
<https://doi.org/10.1109/ICCC.2015.7432984>
- Perkins, D., & McGinnis, E. (1997). *Understanding SNMP MIBs* (p. 528). Englewood Cliffs: Prentice Hall PTR.
- Rad, B. B., Bhatti, H. J., & Ahmadi, M. (2017). An introduction to docker and analysis of its performance. *International Journal of Computer Science and Network Security (IJCSNS)*, 17(3), 228.
- Rose, M. T., & McCloghrie, K. (1990). *Structure and Identification of Management Information for TCP/IP-based internets* (No. rfc1155).
- RRD Tool Home Page: <http://oss.oetiker.ch/rrdtool>.
- Runsungnoen, M., & Anusas-amornkul, T. (2020). Round Robin scheduling based on remaining time and median (RR_RT&M) for cloud computing. In *Smart Trends in Computing and Communications* (pp. 21-29). Springer, Singapore.
https://doi.org/10.1007/978-981-15-0077-0_3
- Shafi, U., Shah, M. A., Wahid, A., Abbasi, K., Javaid, Q., Asghar, M. N., & Haider, M. (2020). A novel amended dynamic round robin scheduling algorithm for timeshared systems. *Int. Arab J. Inf. Technol.*, 17(1), 90-98.
- Xiaogao, Y. (2007). The research on the program technique of implementing NMS in win SNMP API. *Computer Applications and Software*, 24(1), 121-122.
- Zaidi, T., & Shukla, S. (2018). Variable time quantum-based Round Robin policy for the cloud computing environment. *International Journal of Computer Sciences and Engineering*, 6, 1075-1081.
- Zeng, W., & Wang, Y. (2009, April). Design and implementation of server monitoring system based on SNMP. In *2009 International Joint Conference on Artificial Intelligence* (pp. 680-682). IEEE.
<https://doi.org/10.1109/JCAI.2009.34>
- Zukowski, M., Heman, S., Nes, N., & Boncz, P. (2006, April). Super-scalar RAM-CPU cache compression. In *22nd International Conference on Data Engineering (ICDE'06)* (pp. 59-59). IEEE.