Research Article

Designing Energy Management Aware Task Scheduling Algorithm And Model For Cloud Data Centers

Dr. Deepak Kumar Sinha[1], Dr. Garima Sinha[2]
[1]Professor, IIMTU, Meerut dipu_sinha@yahoo.co.in
[2]Professor, IIMT Engineering College, Meerut mailatgarima@yahoo.co.in

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract
In cloud computing many tasks come to servers from different sites for computation and storage which makes data centers consume large energy and become less responsive. Task scheduling is one of the best ways to utilize cloud data center resources efficiently, lower power consumption and latency, and empowers available bandwidth use. Task scheduling algorithms and data center network models employed previously are not much efficient from energy consumption perspective due to the features that they haven't included. To solve this problem an appropriate and efficient proximity based task scheduling technique and model are designed and implemented for cloud computing data centers.
This research work proposes and implements a new proximity based task scheduling algorithm and scalable data center model that can minimize power consumption and increase scalability of cloud computing data centers. The evaluation results, found from the experiment are presented and showed more efficient outcome in terms of energy efficiency for both the proposed scheduling algorithm and data center network model.

Keywords: cloud computing, power management, datacenter energy efficiency, proximity, task scheduling

1. Introduction

The task scheduling algorithms previously employed to minimize data center power consumption are not much efficient, because in those algorithms, the proximity of computing resources was not considered during scheduling and if free resources is found anywhere in the cloud data center (access layer, core layer), the tasks could be assigned a resource.

In this research we have designed an efficient proximity based task scheduling algorithm which first checks the availability of a resource near to the task to assign host, and assign a task to remote resource only if nearby resources are occupied. This feature can reduce the latency in task execution and in turn minimize the energy waste (extra energy consumption) that cloud be lost for the latency time. Additionally, due to the scalability and energy efficiency limitations of current data center models, a scalable cloud data center model that can support many computing machines with limited number of switches is proposed to reduce the energy consumption.

In cloud computing data center modeling, different models have been proposed but the popular and commonly used data center architecture this days is three tier fat tree model, which comprises of three layers of network switches, namely core layer found on the top level and responsible for central data center networking, core data exchange and external communication, aggregation layer aimed to connect core and access network layers of the data center network, and access layer which connects the servers arranged in racks to aggregation layer network and give direct access to client requests. The arrangement is done in logical order from the central computing node to lower level components in a hierarchy.

Fat-tree data center network design is a switch-centric data center model that uses a tree like topology with commodity equipment and solves the oversubscription problem at some extent faced by traditional three-tier data center topology. As the traditional three tier design, it follows a hierarchical arrangement of core, aggregation and access network layers with internal working difference. The numerical relationship between core, aggregation and access (edge) layer networking switches is stated in Section 4.3 to compare with the proposed model. This data center network architecture solves most of the limitations of three-tier data center network design, but has scalability, network congestion and energy efficiency problems which needs to be solved in this research. Fat-tree data center network model is done based on the fact that, if the data center is designed to have n pods, then the data center will support $(n/2)^2$ core layer switches, and $n^2/2$ aggregation and access layer networking switches .

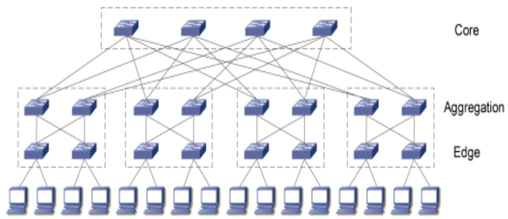

Figure 1:Cloud computing fat-tree data center network model

2. Proposed Data Center Network Model

As we have shown in Section 1, the most commonly used data center network models these days, have their own advantages and limitations. By using those data center network models as a background, in this research we have tried to modify and improve the most popular and recently widely used fat-tree cloud computing data center network architecture. This model tries to solve the previous models flexibility, network congestion, and scalability problems and can reduce energy consumption. The data center network model is designed to support many servers and proposes high bandwidth communication links to overcome the oversubscription problems occurring at aggregation network layer.

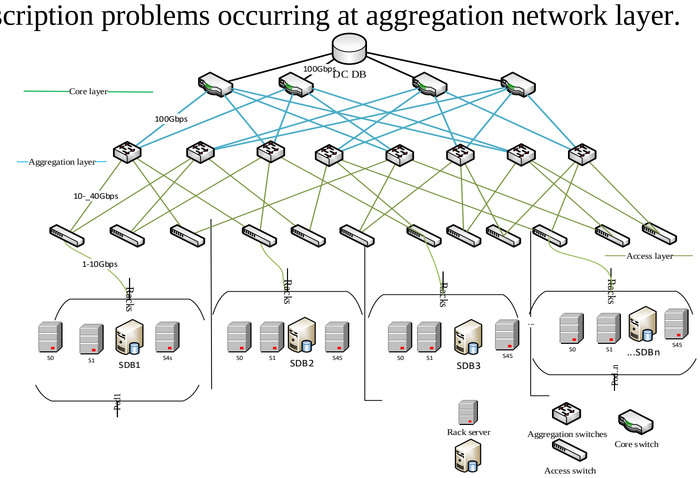

Figure 2: Proposed data center network model

The proposed task scheduling system model and data center network model are designed with Microsoft Visio 2016.The data center network model is composed of three network layers, namely core layer, aggregation layer and access layer each having its own responsibility. To avoid the port deficiency and support maximum access layer server connection, we use the recent most capable networking switches.

The proposed data center network model is composed of n number of pods and $(n/2)^2$ number of core networking switches, which totally comprises of $5(n)^2/4 + (n/2)^2$ networking commodity switches, where $(n)^2/2$ are for aggregation network layer and $3(n)^2/4$ for access network layer.

At access network layer each switch can support 48 server connections, which is the maximum capacity of recent popular commodity networking switches. In this data center model the oversubscription at aggregation layers is solved by increasing the bandwidth of the communication links as shown in Figure 4.1. The number of racks supported at the access network layer as shown in Figure (4.1) is applied for all access layer switches in each pod.

Fat-tree data center network model is done based on the fact that, if the data center is designed to have n number of pods, then the data center will support $(n/2)^2$ core layer networking switches, $n^2/2$ aggregation and $n^2/2$ access layer networking switches.

Table 1 and Table 2 shows the difference between the current popular data center network model and proposed model in terms of the number of networking switches that the models can support.

Architecture Comparison between Fat-tree Data Center Network Model and Proposed Data Center Network Model in terms of scalability as the Data Center Size Grows

Fat-tree data center network architecture scaling trend

Table 1: Fat-tree architecture networking switch support trend

No. of Core layer switches	4	9	16	25	36	64	256	1024
No. of Aggregation Layer switches	8	18	32	50	72	128	512	2048
No. of Access layer switches	8	18	32	50	72	128	512	2048

Proposed data center network architecture scaling trend

Table 2: Proposed architecture networking switch support trend

No. of Core layer switches	4	9	16	25	36	64	256	1024
No. of Aggregation layer switches	8	18	32	50	72	128	512	2048
No. of Access layer witches	12	27	48	75	108	192	768	3072

In table 1 and table 2, we have tried to show the scalability difference between fat-tree data center network model and proposed data center network model. As it can be seen in the tables, the comparison of data center network models is mainly targeted the amount of servers and switches supported, which has a direct relation with energy consumption. Hence compared to fat tree data center network model, the proposed model can support many computing servers with little number of pods and switches, which shows it's more scalable feature. This property helps to reduce the overall energy consumption of cloud computing data centers, because networking switches consume large amount of energy when they are in active state.

In the proposed data center model to reduce oversubscription problem occurred at the aggregation layer, instead of increasing the number of networking switches at the aggregation and access network layers, we have used data communication links with throughput ranging from 10G to 40G. The bandwidth used at aggregation(40G) layer makes the overall network bandwidth comparable to the sum of its access layer links (1-10G) connected to it (Figure 2).

The reason that motivates us to make the above decision is that, networking switches are power hogs that need much energy for their proper operation and functioning. Unlike servers, networking switches can't be turned off even in low load state for the data center's 24/7 active operation, which in turn increase the overall data center energy consumption and conflicts with our energy efficiency goal.

In cloud computing network models, the central cloud computing database is assumed as the home of all the cloud computing data requests. Components connected to central database through routers represents cloud computing data centers found in different areas and work in collaboration with other data centers through networks. Routers that connect data centers with one another and with central database are the means for data and information exchange

In the proposed data center model, data center DB represents the cloud computing data center database, in which all data center data is assumed to be here, and requests that didn't responded at access layer servers and databases get response. Core layer switches are high power networking devices that helps for efficient communication with aggregation network layer. Aggregation layer networking switches are used as a middleware between access and core network of the cloud computing data center, and access layer switches are used for direct rack server connection and data transmission to aggregation network layer.

In each rack, computing servers, which are responsible for managing and responding to client requests are found and arranged properly. The servers are arranged inside racks for safety, management and other related issues, and racks in turn are arranged in pods. In each pod a rack database is found for frequent tasks request storage. If the data requested is not here the request will be sent to data center database and

if not, to the central cloud database. For each layer data exchange, communication links from and to computing, switching and storage devices are found at each layer. Their variety is shown in Figure 2.
In the proposed data center network model very high bandwidth (40-100G) links are used for the core-aggregation connection and medium bandwidth links, usually 10G Ethernet links for aggregation-access connection. For external data center and central cloud computing database connection 100Glinks are used to improve response time and performance of the overall system.
Access layer is a data center network layer where high computing power servers are directly attached and many computation operations are performed. Additionally at each pod, server databases are found to manage the local data management and improve response time of the client requests. The server database is important for easy data replication, management and to work efficiently with the whole databases that are connected with it, which in turn contribute for energy efficiency. Racks are used in cloud computing data centers specifically at this layer for server devices protection, proper arrangement and easy management. The servers are arranged inside racks as 1U, 2U, 4U… where U represents server rack units with 1.75 inch height, 19 inch width and varying depth.

3. Proposed Task Scheduling System Model

The proposed overall task scheduling system model works as follows. First various tasks from different sites (users, groups, other actors) come to cloud computing data centers for computation or storage. To perform such activities, since the resources we have in cloud computing data centers are limited, all the tasks may not be executed and assigned a resource at a time, So the tasks should be identified for importance and urgency to set priority and put them in an execution and storage queue. After the tasks are identified, we set priority, and group them as urgent or non-urgent, toassign them a resource for execution and storage in their proper order. The priority will be set and calculated by counting the frequency/occurance of each incoming tasks.
Based on the algorithm we implement, task preemption and queueing will also be done to avoid long time and infinite waiting of execution and storage resource for the queued tasks. The task scheduling activity will be done in iteration, because new tasks come and executed tasks will leave continuously from the cloud computing data centers.
The best way to perform task execution in this research is executing them at data center access layer if enough and nearby resource (storage, computing, communication) is available. We also compare for proximity of a computing and storage machines at this layer. If not,we go to data center database servers found at core layer of the data center model (see Figure 2) as second option, and if it is still imposible, we schedule task execution and storage (assign resources for tasks) at central cloud computing servers.
We propose this mechanism because if the task computation is possible at access layer of cloud computing data centers (nearby resources to incoming tasks), the time letency and power consumption will be minimized significantly, and we select the rest options only if the above option is imposible.The scheduler checks proximity of resources even at a single data center layer. The mechanism can reduce energy consumption of the whole data center, because if task execution is completed early, the computing and storage resources will be free of work, and can become in idle state to save energy.

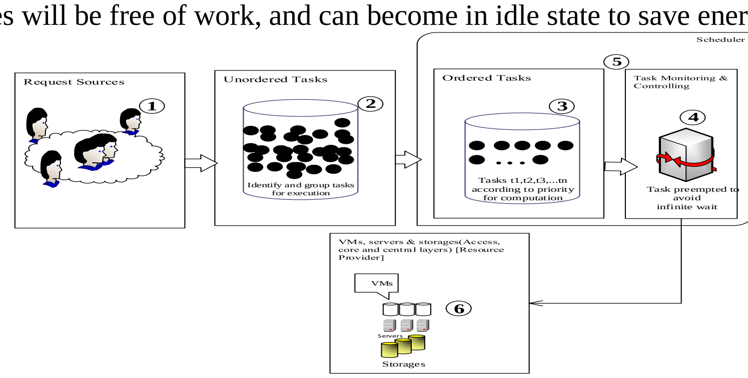

Figure 3: Proposed task scheduling system model

The proposed task scheduling system model mainly contains six components as shown by the circled numbers in Figure 3. The component identified by (1), describes many possible sources of tasks (individual users, groups, others) that come to cloud computing servers and storage machines. The tasks are cloud tasks since they are computed and stored in the cloud computing environment. As tasks come from different sources, they are random and in unordered manner as shown in (2). The section assigned by (2) shows tasks coming from different sources before getting into a scheduler and are not assigned a resource at data center computing machines.
At stage (3), tasks are ordered for computation and resource assignment based on their priority such as request rate of the task, and is enabled to identify the nearby server by a scheduler. Number (4) shows the task preemption which helps to protect infinite waiting. Sections identified by (3) and (4) are parts of subsystem (5) where the scheduling algorithm is implemented and nearby resource is selected by the scheduler. The resources shown in (6) comprises of various computing and storage components of acloud computing data center including access and core layer data center database storages and computing servers. Storage components (grey), computing servers (white-blue) and corresponding virtual machines (white) are found in the subsystem. This portion (6) of the task scheduling system model shows the resource provider component, which gives tasks a resource for storage and computation.

4. Energy Model

The energy consumption of a cloud computing data center is the sum of energy consumption of each constituting parts, including computing servers, storage devices, networking switches, communication links and cooling devices. For this research we have mainly focused on energy consumption of computing servers and networking switches.

4.1. Overall Energy Model for Cloud Computing Data Centers

Energy modeling is very crucial task in computing systems. Cloud computing data centers are one of the most sensitive areas where large amount of energy is consumed and very high volume of greenhouse gases are released to the environment.
The overall energy consumption model for cloud computing data centers can be formulated as:

$$E_{data\ center} = E_{servers}+E_{storages}+ E_{core} + E_{aggregation}+ E_{accesss} + E_{other\ devices} \ldots\ldots\ldots\ldots\ldots(4.1)$$

Where; $E_{data\ center}$ is the total cloud computingdata center energy consumption, $E_{servers}$is energy consumption of computing servers, $E_{storages}$is the energy consumption of storage devices found at each pod and data center database, E_{core} is the power needed for core layer networking components, $E_{aggregation}$ is the energy consumption of aggregation network layer switches, E_{access} is the power consumption of access layer switches and $E_{other\ devices}$ is the power needed for other devices such as communication links and cooling .

4.1.1. Energy Model of Switching Materials in Data Center Networks

The energy consumption of the switch components used in cloud computing data centers is highly dependent on the number of active line cards used during communication that provide interface for the network, the port data transfer rate and type of switch used for the data center infrastructure. The total energy consumption can be summarized as:

$$E_{switch=}K_{w(t).}(E_{chasis(t)} + E_{linecard(t)} +\gamma_{(t).} E_{Port\ (t)})\ldots\ldots\ldots\ldots\ldots(4.2)$$

Where;
$K_{w\ (t)}$ = usually lie [0, 1], 1 indicates the switch is running, while 0 shows the switch is turned off
$E_{chasis\ (t)}$ = energy consumed by the switch base hardware
$E_{linecard\ (t)}$ =the energy consumed by active line card

$\gamma_{(t)}$ = number of active ports
$E_{port\,(t)}$ = varies according to the communication rate of the switch [48].
The above mathematical formula applies for all network layer switches in the data center (core, aggregation and access).

4.1.2. Energy Model of Computing Servers inside Racks

P_{Serers}=$P_{chassis}$ + C*P_{unit}……………………………………………………………………….. (4.3)

Where;
$P_{chassis}$ denotes static power used to make memory and I/O devices live,
P_{unit} represents unit server CPU energy consumption and C is the CPU utilization rate [68].

$P_{Unit} = p_s(l) *(2* d_{db} + d_{exce}) *r_{(a)} * t$ ………………………………………………….............. (4.4)

Where;
$p_s(l)$ is energy needed for the server that execute workloads,
d_{db} is the time required for both querying and receiving requests to and from the nearby database which is dependent on the database location and scheduling strategy employed,
d_{exce} is the workload time,
$r_{(a)}$ is the average data access rate/threshold of the unit database, and "t" is the workload or task total execution time [16].

5. Flow chart for Scheduling and Execution of Tasks

The following diagram shows the proposed task scheduling and execution sequential flow in cloud computing data centers.

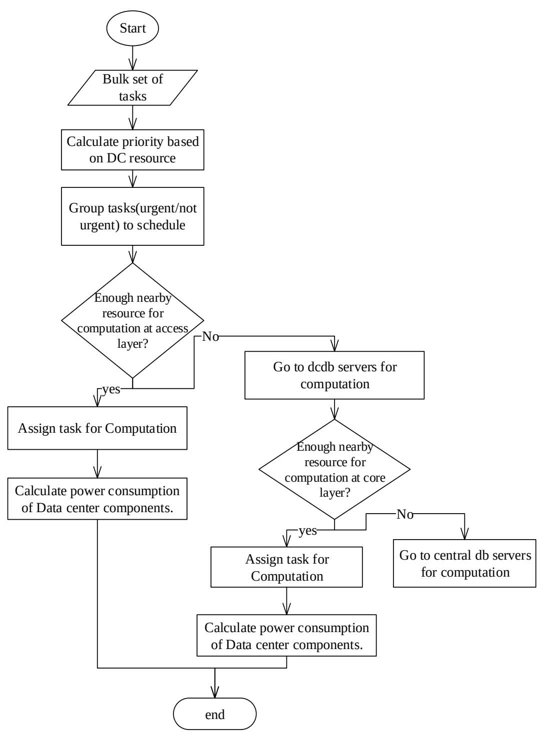

Figure 4:Data flow diagram for task scheduling process

Algorithm Design

The proposed proximity based task scheduling algorithm is based on request rate comparison and resource proximity, and takes the following assumptions and considerations into account during its implementation.

Considerations while designing the task scheduling algorithm.

- Identifying the task for assigning resource
- Considering the update task of all the databases and servers every time after every task scheduling operation is performed
- Bandwidth of networking links in the system model
- Devices used for the cloud computing datacenter have low cost and are easy to find

Assumptions made for proposed task scheduling

- All tasks coming for computation and storage get resource at either cloud data center or central database servers
- The tasks coming for scheduling are selected based on the temporal and special locality (priority based), and assigned to the nearby computing resource.
- During task scheduling activity the task is directed to the appropriate host by the scheduler

Table 3: Notations used in the algorithm

Notations	Description
P_{cenDB}	Power consumption of central database servers
P_{dcDB}	Power consumption of datacenter database servers
P_{rack}	Power consumption of rack level database servers
Thr_{dc}	Throughput of datacenter databases links
Thr_{rack}	Throughput of rack level databases links
$DC_{thr(i)}$	Throughput value of datacenter i
$T_{ar(i)}$	Access rate of task i
$DC_{(i)}$	Datacenter node i
$T_{i \rightarrow} DC_i$	Assign task i at cloud datacenter i
$T_{i \rightarrow} R_i$	Assign task i at rack i

Inputs

- Tasks coming from different sites for computation and storage
- Virtual machines, servers and storage components

Outputs

- Scheduled tasks, uplink and downlink loads of communication link resources
- Energy consumption and throughput of the cloud computing datacenter components

Overall Method

For each $T_{(i)}$ coming from different source
  If $T_{(i)}$ €$DC_{rack\ (i)}$ //*task T is coming frequently and found in the cloud databases*
    If ($P_{rack\ (i)<}P_{Dc\ (i)}$ &&$thr_{rack\ (i)\ <}thr_{DC\ (i)}$)
If (nearby free resource at rack)
      If (Check computing, storage and networking capacity is enough)
      End if
End if
$T_{(i)\ \rightarrow}rack_{\ (i)}$ //*First recommended scheduling option ->at rack db servers (access layer)*
End if
    Calculate $P_{DC\ (i)}$, $thr_{DC\ (i),}$ $thr_{rack\ (i)}$
  End if
Elseif

($T_{ar\,(i)} > T_{ar\,(j)}$)&& (free resource)
$T_{(i)} \rightarrow$ rack $_{(i)}$, Calculate $P_{DC\,(i)}$ &&$thr_{DC\,(i)}$
End
If (P_{dcDB}<P_{cenDB}) && ($thrdc_{DB}$<thr_{cenDB})
If (! nearby free resource at rack)
T_i->dc_{DB}// *Second scheduling option if option one do not meet -> at dcdb servers (core layer)*
Calculate $P_{DC\,(i)}$, $thr_{DC\,(i)}$
End if
Else
T_i->cenDB*// Final scheduling option ->at the central cloud database servers*
End if
End for

6. Conclusion

In this research we have proposed and implemented proximity based task scheduling algorithm and scalable cloud data center model that can minimize power consumption and waste in cloud computing data centers. The proposed proximity based task scheduling algorithm have solved, previously existing holes by including proximity of computing and storage machines in the system model and during implementation. In this algorithm, the scheduler enables incoming tasks to detect the nearby resource. The proposed data center model also improves scalability of cloud computing data centers to cope with the growing technology and helps to use electric power efficiently. The proposed data center network model best fits with the current growing technology compared to fat tree data center model both in scalability and energy efficiency. The main contributions of this research work are enhanced scalability and new proximity based task scheduling algorithm that can both minimize energy consumption of cloud computing data center servers and the whole cloud computing data center at large.

References:

[1] T. a. O. A. a. C. H. a. B. I. a. P. J.-M. a. V. A. V. Mastelic, "Cloud computing: Survey on energy efficiency," Acm computing surveys (csur), vol. 47, 2015.
[2] M. G. a. D. K. a. P. Bouvry, "HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers," 2015 IEEE 8th International Conference on Cloud Computing, pp. 742-749, 2015
[3] S. a. Y. S. a. E. M. a. A. M. Atiewi, "A review energy-efficient task scheduling algorithms in cloud computing," 2016.
[4] N. K. W. J. kaur, "An Enhancement of DVFS Algorithm for Cloud Computing," International Journal of Computer Science Trends and Technology (IJCS T), vol. 4, Jul - Aug 2016
[5] Y. a. S. A. Mhedheb, "Energy-efficient Task Scheduling in Data Centers," 2016.
[6] G. S. a. I. N. C. S. Shanmugam, "Effort of load balancer to achieve green cloud computing: A review," International journal of multimedia and ubiquitous Engineering, vol. 11, pp. 317--332, 2016.
[7] N. a. A. I. Horner, "Power usage effectiveness in data centers: overloaded and underachieving," vol. 29, pp. 61--69, 2016.
[8] B. a. R. A. M. a. R. A. Rahmati, "Data replication-based scheduling in cloud computing environment," Journal of Advances in Computer Engineering and Technology, vol. 3, pp. 75--80, 2017.
[9] E. a. F. S. a. H. R. a. T. Z. a. Z. A. Y. Baccour, "PTNet: An efficient and green data center network," Journal of Parallel and Distributed Computing, vol. 107, pp. 3--18, 2017.
[10] T. a. P. H. a. L. W. a. Z. A. Y. Yang, "An energy-efficient virtual machine placement and route scheduling scheme in data center network," Future Generation Computer Systems, vol. 77, pp. 1--11, 2017.
[11] P. R. Padala, Virtualization of Data Centers: study on Server Energy Consumption Performance, 2018.

[12] T. a. M. H. a. T. T. a. H. N. a. H. K. a. F. T. a. A. H. Yoshida, Power supply unit, Google Patents, 2018.

[13] "Difference Between |Description analysis and Comparision," 2018. [Online]. Available: http://www.differencebetween.info/difference-between-rack-and-blade-servers. [Accessed 10 December 2018].

[14] S. K. Kappagantula, Virtualization of Data Centers: Case Study on Server Virtualization, 2018.
R. a. M. I. a. S. M. Rahmani, "A complete model for modular simulation of data centre power load," arXiv preprint arXiv:1804.00703, 2018.

[15] Garima Sinha and Deepak Kumar Sinha. 2020. Enhanced Weighted Round Robin Algorithm to Balance the Load for Effective Utilization of Resource in Cloud Environment. EAI Endorsed Transactions on Cloud Systems 6, 18 (2020).