INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Designing Architecture for Handling Data Locks in Cloud Computing Environment

Kamini Kanchan
Student, Department of Computer Science, Apex Institute of Engineering and Technology, Jaipur, Rajasthan, India
Preeti Thakur
Assistant Professor, Department of Computer Science,
Apex Institute of Engineering and Technology, Jaipur, Rajasthan, India

Abstract:
The recent trend in cloud computing sees the usage of agents for service providing. The problem remains in assigning data locks to users in case of conflicts. This paper deals with a way of handling data locks by providing a novel architecture where conflicts occurs between the users. The architecture uses multiple agents including mobile agents and fixed agent. This paper proposes a scheme that increases the speed of write lock assignment in the SaaAS (Software as an Agent Service) Environment in an efficient manner with the help of few messages passing. It also takes a look onto the portability and interoperability of agents in a Cloud computing environment. This paper also deals with the transaction delay due to agent lost in the middle of the transaction which leads to communication gap.

1. An Introduction to Cloud Computing

When we store our photos online instead of on our computer system, or using webmail or a social networking site, we are using a "CLOUD COMPUTING" service. Cloud computing refers to the delivery of computing resources over the internet. Instead of keeping data on our own device or updating applications for our need, we use a service over the internet, at another location, to store our information or use its applications. Doing so may give rise to certain privacy implication.

Cloud computing is the delivery of computing services over the internet. Cloud services allow individuals and businesses to use software and hardware that are managed by the third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail and online business applications. The cloud computing model allows access to information and computing provides a shared pool of resources, including data storage space, networks, computing processing power, and specialized corporate and user applications

A definition of cloud computing developed by the US National Institute of Standards and Technology (NIST) "Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resource (example networks, servers, storage, applications and services) that can be rapidly provisional and released with minimal management effort or service provider interaction".

The cloud computing consists of three different layers. The lowest layer is IaaS (Infrastructure as a Service) provides basic infrastructure component. Then, intermediate layer is PaaS (Platform as a Service) which provides platform oriented services and allows the use of hosting environment as per specific need. Finally at the top most there is SaaS (Software as a Service) with complete application offered as service on demand [3]. There can both types of communication in cloud computing environment intra and inter. Where intra communications is done with the help of high speed LAN and inter communication is done with the help of low speed WAN [1]. This concludes that they maintain interoperability and portability in cloud computing environment. This paper uses a cloud computing model with mobile agents to reduce the interoperability problems.

The rest of this paper is organized as follows: section 2 provides literature review on past researches which present through the idea on how this paper is being inspired, section 3 gives architecture of agent based cloud computing which solves the problem of access lock, section 4 provides method which gives a summarized review of the complete proposed process, section 5 consist of algorithm of the whole process, section 6 presents a case study related to the identified problem which is to be solved by this paper, section 7 gives transaction delay which helps in understanding the complete process time requirement. Finally a conclusion of the work along with the future scope and references are discussed in section 8.

2. Literature Review

This work on Realization is based on Open Cloud Computing Federation Based on Mobile Agent [4] which through light on the process that provides a uniform resource interface to the user by incorporating the services of multiple cloud computing service providers. That paper tries to explain the realize portability and interoperability between different kinds of cloud computing platform through Mobile Agent. Then we have another paper on Agent based Online Quality Measurement Approach in Cloud Computing Environment [2] which represent the requirement of online evolution during running services which was required to maintain the quality of Service by Cloud Computing Environment and this requirement was fulfilled by the fixed Agent with appropriate architecture. This idea is based on SaaS by Mobile Agent Based Services for Cloud Computing in Internet Environment [1] which explains divided Cloud and Convergence coherence mechanism with data having "Real Lock" and "Write Lock". And this is ensured by the coordination between Main Consistence Server (MCS) and Domain Consistence Server (DCS), where the DCS works on lock requests of all users in its domain.

After this study on SaaAS in [1] it was found that there was lack in information about the process of assigning write lock to users in conflict cases. The paper lacks in priority based assignment of write lock to users. By the study of all this papers [1, 2 and 4] it was found that there was no solution for the damage of MCS and control over the cloud environment. There was also no solution for the case if the agent is lost in the middle of the transaction which leads to communication gap.

This paper proposes solution to the problems in the papers [1, 2 and 4]. This research work consist with the solution of priority based assignment of write lock where as it also have the solution to the communication gap due to agent lost in middle of transaction. With this solution it also have solution for damage of MCS and control over cloud computing.

3. Agent Based Architecture of Cloud Computing

After analyzing the paper on SaaAS [1] it is found to lack information regarding the process of assigning write lock to user in case a conflict arises. The past research work failed to provide any method of prioritization of requests for write lock from users. The existing methodologies [1, 2 and 4] do not offer any solution in case the MCS is damaged and the control over the cloud environment is lost. Although agent based techniques are commonly used for cloud computing, the present review work finds no solution in case the agent is lost in the middle of a transaction creating a communication gap.

Here SaaAS architecture consists of working groups which is made up of domains. This domain consists of cloud servers and fixed agents. These on connected to each other with high bandwidth LAN. Each domain is then interconnected with low bandwidth WAN. There are some domain known as DCS (Domain Consistence Server) and other which are located in low populated working groups. The database which is responsible for the maintenance of all the information flowing through network and MCS (Main Consistence Server) is known as known as DBCS (Database Consistence Server). This DBCS provides the user with data lock. In the process of getting data lock from MCS and the fixed agents for scheduling the request, mobile agents are used as key driver. This architecture consists of three levels and they are as follows:

a. User level: This level manages all the users present in different work stations.
b. Database management level: This level manages the main Database Server where records of all the transaction are stored and maintained.
c. Main management level: Main Consistent Server present in this level provides lock to the users.

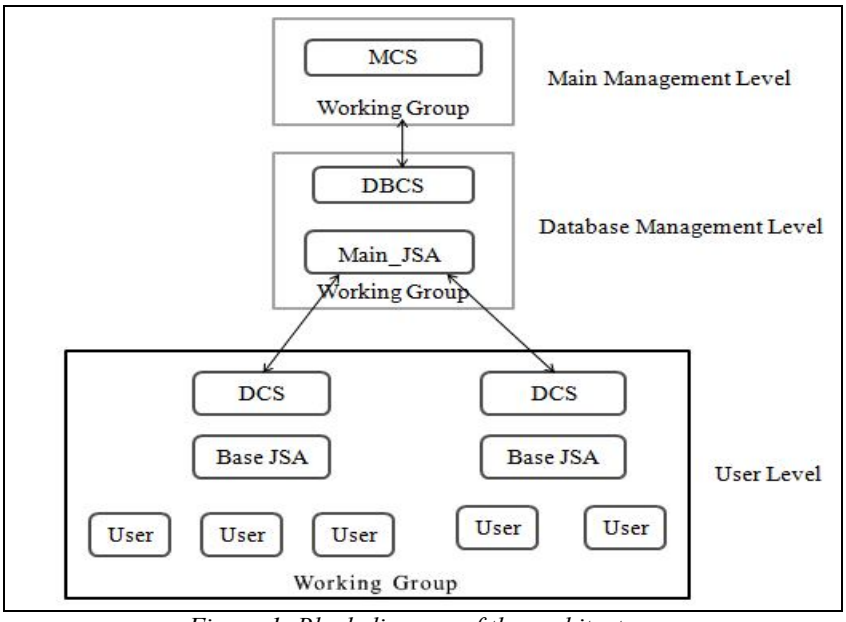

Figure 1: Block diagram of the architecture

Data Structure	Description
IA (Interface Agent) (Source_Id, Termination_Time, forward/receive , Status, message)	This Agent takes the information from the user to the DCS.
WA (Working Agent) (Source_Id, Termination_Time, Forward/receive , Status, message)	This Agent takes the information from DCS to the DBCS.
JSA (Job Scheduling Agent) (Source_Id, Termination_Time, Forward/receive , Status, message)	This Agent communicates with DBCS and MCS.
RA (Reply Agent) (Source_Id, Termination_Time, Forward/receive , Status, message)	This Agent is used to send the reply from MCS or DBCS to user with a Lock_Grant message.
QA (Queuing Agent) (Source_Id, Termination_Time, Forward/receive , Status, message)	This Agent is used to send the waiting message from DBCS to user.
UA (Update Agent) (Source_Id, Termination_Time, Forward/receive , Status, message)	This Agent is used to update the MCS database. This message goes to MCS from DBCS.
Base_JSA(Base_Job Scheduling Agent)	This Agent schedule the request came from users.
Main_JSA(Main_Job Scheduling Agent)	This Agent schedule the request came from DCSs.
DMA (Domain Management Agent)	Manage all the mobile Agents presents in a domain.
FDMA (Fixed Domain Management Agent)	Manage all the fixed Agents in a domain.
GMA (Group Management Agent)	Manage DMA and FDMA in a working group.
MMA (Main Management Agent)	Manage all the GMA.
Req_msg (Data_File_Id, Lock_Req, Completion_Time, Local_Time_Stamp, Authentication_Code)	This message carries the information related to request of lock from USER to DCS.
MUReq_msg (Data_File_Id, Lock_Req, Completion_Time, Lock_Status, User_Id)	This message carries the information from DCS to DBCS if the pre-existing lock is found in DCS.
UReq_msg (Data_File_Id, Lock_Req, Completion_Time, User_Id)	This message carries the information from DCS to DBCS if the pre-existing lock is not found in DCS.
URep_msg (Destination_Id, Lock_acquire_time, Lock_Grant, User_Id)	This message carries the information from DBCS to USER if the lock is already acquired by another user.
Lock_msg (Data_File_Id, Lock_Req, User_Id)	This message carries the information from DBCS to MCS if the lock is not acquired by another user.
Rep_msg (Destination_Id, Data_File_Id, Lock_Grant, User_Id)	This message carries the information from MCS to USER when a lock is first time granted.
WRep_msg (Destination_Id, Data_File_Id ,Lock_Grant , User_Id)	This message carries the information from DBCS to USER when DBCS grant the lock request to waited user.
Update_msg (Data_File_Id)	This message updates the data base of MCS. So it carries the information from DBCS to MCS. MCS update its database repeatedly.
DCS_Main(Data_File_Id, Lock_Present, Completion_Time)	This table maintains the information of Data_File_Id, Lock_Present and Completion_Time of the user.
DCS_Back (Data_File_Id, User_Id, Authentication_Code)	This table maintains the information of Data_File_Id, User_Id and Authentication_Code of the user.
DBCS_Main (Data_File_Id, Lock_Present, Completion_Time)	This table maintains the information of Data_File_Id, Lock_Present and Completion_Time of the user coming from DCS.
DBCS_Back (Data_File_Id, User_Id, Lock_acquire_time, DCS_Id)	This table maintains the information of Data_File_Id, User_Id Lock_acquire_time and DCS_Id of the user coming from DCS.
DBCS_Wait (Data_File_Id, Completion_Time, User_Id, Lock_Req, DCS_Id)	This table maintains the information of Data_File_Id, Completion_Time, User_Id, Lock_Req and DCS_Id of the user coming from DCS when a conflict occurs between two requests.
MCS_Main (Data_File_Id, Lock_Present, User_Id)	This table maintains the information of Data_File_Id, Lock_Present and User_Id of the user coming from DBCS.

Table 1: Detail description of the keywords used in this Paper

4. Methodology Used to Handle Data Locks

The complete methodology can be divided into 6 sections:-

4.1. Section 1: *Process of generating User_Lock_Request*

For this IA is created and Req_msg is required, then these both are concatenated. Then send IA to Base_JSA for getting the lock. After that read RA with Rep_msg/URep_msg/WRep_msg from Base_JSA regarding lock access. If Lock_Grant is equal to URep_msg then

wait for Lock_acquire_time duration. Then read WRep_msg from Base_JSA else directly get the Rep_msg/WRep_msg. With the help of this section user will retrieve the data of Lock_Time_Stamp from the remote server and save the data in local buffer.

4.2. Section 2: Base_JSA Scheduling

This section shows the Base_JSA scheduling request that is received by it. For that first it gets the IA with Req_msg from the user then there will be two cases of forward and receive. First, if the request is forward then compare the status to the previous status, when the status is same then compare Local_Time_Stamp, assign priority and send IA to DCS. Else if the request is received then read RA from DSC then check status for assigning priority and send RA to User.

4.3. Section 3: *Domain Server Process*

This section shows how DCS modifies its table and generates requests. First of all IA is received with Req_msg from Base_JSA. Then the status is checked. If status of IA is high, then Authentication code is checked and if the match is not found, then the status is assigned as low priority. After that check DCS_Main table for pre-existing Write lock. Then duplicate of IA is created to WA and check Lock_Present Status. If it is present, then there are two cases, first create the URep_msg and concatenated with WA. Second create MUReq_msg and concatenate with WA. Then these two are sent to Main _JSA for updating. In Main_JSA Data_File_Id, Completion_Time, User_Id, and Lock_Present are updated. After that RA is sent to Base_JSA.

4.4. Section 4: Main _JSA Scheduling

This section shows the scheduling in Main_JSA According to the request received by it. First of all get WA with MUReq_msg from DCS. Then check for forward or receive request. If there is forward request, compare status with previous request status and if the status is same then compare the Completion_Time and assign the priority and send WA to DBCS. Else if there is receive request, Read RA or QA FROM DBCS. Then check status for assigning priority and send RA or QA to DCS.

4.5. Section 5: DataBase_Server Processing

This section shows how DBCS modifies its table and generates requests. For this first get the WA from Main_JSA, and then check for Destination_Id, Data_File_Id, Lock_Grant and User_Id. If Lock status is not present then search DBCS_Main table for pre-existing write lock. For each request there can be two cases, first update and the second create JSA duplicate WA and Lock_msg. Then concatenate these both and send to MCS. Else if lock status is present update Data_File_Id, User_Id, Lock_acquire_time and DCS_Id. After this Lock_acquire_time is calculated and DBCS_Back is updated. Then URep_msg and QA is created duplicating WA and then QA along with URep_msg is sent to Main_JSA. Now RA is received from MCS and if Data_File_Id of DBCS_Wait is equal to Data_File_Id of Completion_Time then update DBCS else send RA with Rep_msg to Main_JSA again check for Destination_Id, Data_File_Id, Lock_Grant and User_Id. Then Grant the lock to user in DBCS_Back and create Destination_Id and sent it to Main_JSA. Now check DBCS_Main table for Data_File_Id and corresponding Lock_Present and if it is not present create both UA and Update_msg, concatenate each other and send it to MCS.

4.6. Section 6: *Main server Processing*

This process shows how MCS grant the lock. For this first get Lock_msg and update MCS_Main table. Then create RA duplicating JSA and Rep_msg and concatenate each other and send to DBCS. Then receive UA and update MCS_Main table.

5. Algorithm to Handle Data Locks In Cloud Computing Environment

Further, the complete algorithm for the methodology can be divided into 7 segments:

5.1. User to DCS: When request a lock

- Step 1: Start. \\User
- Step 2: Create IA and generate Req_msg for Data_File_Id, Lock_Request, Completion_Time, and Lock_Time_Stamp.
- Step 3: Go to Base_JSA and check for request forward is no go to step 4 else go to step 5
- Step 4: Go to Main_JSA to User segment.
- Step 5: Check for the status high? If yes go to step 7 else go to step 6. If same status then check Local_Time_Stamp is low, again if it is yes go to step 7 else go to step 6.
- Step 6: Set low priority and go to step.
- Step 7: Set high priority and go to step 8.
- Step 8: Schedule job and send to DCS.

5.2. DCS to DBCS

- Step 1: In DCS if status check for authentication code. If yes go to step 3 else go to step 2.
- Step 2: Set lowest status.
- Step 3: Set same.
- Step 4: Now check for pre-existing write lock. If there is no lock then go to step 6 else go to step 5.

- Step 5: Create WA and prepare to generate MUReq_msg (Data_File_Id, Lock_Req, Completion_Time, User_Id) and move to Main_JSA.
- Step 6: Again check for request for forward if no stop else go to next step.
- Step 7: Check for status high and if same status then check for Completion_Time low. If status is high set high priority and schedule the job else first set low status and then schedule the job.
- Step 8: Move to DBCS.

5.3. DBCS to URep_msg or MCS

- Step 1: In DBCS check for Lock_Status high in DBCS_Wait table. If yes go to step 6 and if no go to next step.
- Step 2: Check for pre-existing write lock if yes go to step 6 else go to next step.
- Step 3: Create JCS and Lock_msg for Data_File_Id, Lock_Req and User_Id and then move to MCS and Grant lock.
- Step 4: Create Rep_msg for Destination_Id, Data_File_Id, Lock_Grant, User_Id and move to MCS to Main_JSA segment.
- Step 5: Now check if status_Indicator is 0, if yes go to step 6 else go to step 1.
- Step 6: Update DBCS_Back table and calculate Lock_acquire_time.
- Step 7: Generate URep_msg for Destination_Id, User_Id, Lock_acquire_time, and Lock_Grant.

5.4. MCS to Main_JSA

- Step 1: In MCS create RA and Rep_msg for Destination_Id, Data_File_Id, Lock_Grant and User_Id.
- Step 2: Move to DBCS and update DBCS_Main table.
- Step 3: Check if Data_File_Id of Rep_msg and Data_File_Id of DBCS_Wait is same. If yes go to next step else go to step 5.
- Step 4: Set Status_Indicator to 0 and move to DBCS to URep_msg segment.
- Step 5: Set RA and Rep_msg status indicator to 1 and move to Main_JSA.

5.5. Main_JSA to User

- Step 1: In Main_JSA check for request receive. If yes go to step 3 else go to next step.
- Step 2: Move to DCS to DBCS segment.
- Step 3: Check for status high. If yes set high priority and go to next step, else set low priority and go to next step.
- Step 4: Schedule the job and move to DCS.
- Step 5: Update table and move to Base_JSA and check for the request received. If yes go to step 7 else go to next step.
- Step 6: Move to User to DCS segment.
- Step 7: Check for high status. If yes set priority high and go to next step, else set low priority step 7 else go to next step.
- Step 8: Schedule the job and move to User.
- Step 9: Stop.

5.6. DBCS (WRep_msg) to User

- Step 1: In DBCS_Back table set WRep_msg for Destination_Id, Data_File_Id, Lock_Grant, User_Id.
- Step 2: Move to Main_JSA check for request receive. If yes go to step 3 else go to next step.
- Step 3: Move to DCS to DBCS segment.
- Step 4: Check for status high. If yes set high priority and go to next step, else set low priority and go to next step.
- Step 5: Schedule the job and move to DCS.
- Step 6: Update table and move to Base_JSA and check for the request received. If yes go to step 7 else go to next step.
- Step 7: Move to User to DCS segment.
- Step 8: Check for high status. If yes set priority high and go to next step, else set low priority step 7 else go to next step.
- Step 9: Schedule the job and move to User.
- Step 10: Stop.

5.7. DBCS to MCS (For Database Update)

- Step 1: In DBCS check for DBCS_Main Table for Data_File_Id and corresponding Lock_Present. If yes go to step 3, else go to next step.
- Step 2: Create UA and Update_msg for Data_File_Id, then move to MCS and delete the entry.
- Step 3: Stop.

6. Case Study

6.1. Case: - When one of the users is author and there is a pre-existing write lock.

- if User 1 is the author
  - Let us consider that User1 and User8 want the Write lock on a Data file D that is User3 already acquired simultaneously. For that User1 creates an Agent 'IA1' and User8 creates an Agent 'IA8', and Req_msg.
  - Agent 'IA1' is dispatched to Base_JSA1 and Agent 'IA8' is dispatched to Base_JSA4 for scheduling the requests.

- After scheduling based on Status and Timestamp Base_JSA1 and Base_JSA8, dispatch the Agents to DSC1 and DCS2.
- DCS1 and DCS2 check their respective tables for the presence of any pre-existing Write lock on the file 'D'. If no such Write Lock is found in their respective tables, then a UReq_msg and WA1 & WA4 are created and send the request by Both DCSs to Main_JSA.
- On receiving the request Main_JSA schedule the request based on Status and Completion_Time, send it to DBCS_Wait table.
- DBCS check its DBCS_Main table for pre-existing Write Lock.
- Here DBCS found that User 3 already have the Write lock on Data File 'D', In the DBCS there is a provision of checking Completion_Time of author with a fixed value.
- If Completion_Time is greater than the fixed value no other changes takes place in the database, on the other if it is lesser then the fixed value the author get the priority and get the next execution position and the author's Completion_Time is added to all the other user's execution time.
- DBCS calculates the Lock_acquire_time for both User1 and User8. DBCS creates URep_msg and an agent QA1 and QA2. DBCS sends the message to the User1 and User8 through QA1 and QA2 respectively.
- On completion of the task of User3, first User1 and then User8 gets the Write Lock and updates DBCS_Main table.

7. Transaction Delay Time Calculations

The transaction delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is send out from the source.

Transaction Delay = Propagation Time + Transmission Time + Queuing Time + Processing Delay.

Where,

Propagation time is the time required for a bit to travel from source to the destination. So it can be define as $\frac{\text{distance between source and destination}}{\text{propagation speed.}}$

Propagation speed depends on the medium and on the frequency of the signal.

Transmission time is defined as**:** $\frac{\text{Size (Message)}}{\text{Total bandwidth of communication path}}$

Queuing time is the time needed for each intermediate or end device to hold the message before it can be processed.

Processing Delay can be defined as $\frac{\text{No. of entries}}{\text{Speed of the CPU}}$

Here Transaction Delay can be define as a combination of the Latency for IA, Latency for WA, Latency for JSA and Latency for RA.

Latency for IA defines how long it will take for an entire message to completely arrive at the DCS from the time first bit is send out from User. It is the combination of propagation time (Pt_{User}), transmission time (Tt_{User}), queuing time (Qt_{User}) and processing delay (Pd_{User})

Propagation time for IA is the time required for a bit to travel from User to the DCS. So it can be expressed as $\frac{\mathbf{Distance\ (D_{DCS} - D_{User})}}{\mathbf{Propagation\ speed\ (PS_{User})}}$

Transmission time for IA can be expressed as $\frac{\mathbf{Message\ size\ (M_{User})}}{\mathbf{Bandwidth\ (BW_{DCS-User})}}$

Queuing Time for IA is the time needed for each Intermediate or end device to hold the message that is for the fixed agent (Base_JSA) it is **(n-1) (S_{User}+T_{User})**

Processing Delay for IA can be expressed as $\frac{\mathbf{No.\ of\ entries\ (E_{User})}}{\mathbf{Speed\ of\ the\ CPU\ (SC_{DCS})}}$.

Latency for WA defines how long it will take for an entire message to completely arrive at the DBCS from the time first bit is send out from DCS. It is the combination of propagation time (Pt_{DCS}), transmission time (Tt_{DCS}), queuing time (Qt_{DCS}) and processing delay (Pd_{DCS})

Propagation time for WA is the time required for a bit to travel from DCS to the DBCS. So it can be expressed as $\frac{\mathbf{Distance\ (D_{DBCS} - D_{DCS})}}{\mathbf{Propagation\ speed\ (PS_{DCS})}}$

Transmission time for WA can be expressed as $\frac{\mathbf{Message\ size\ (M_{DCS})}}{\mathbf{Bandwidth\ (BW_{DBCS-DCS})}}$

Queuing Time for WA is the time needed for each Intermediate or end device to hold the message that is for the fixed agent (Main_JSA) it is **(n-1) (S_{DCS}+T_{DCS})**

Processing Delay for WA can be expressed as $\frac{\mathbf{No.\ of\ entries\ (E_{DCS})}}{\mathbf{Speed\ of\ the\ CPU\ (SC_{DBCS})}}$

Latency for JSA defines how long it will take for an entire message to completely arrive at the MCS from the time first bit is send out from DBCS. It is the combination of propagation time (Pt_{DBCS}), transmission time (Tt_{DBCS}) and processing delay (Pd_{DBCS}).
Propagation time for JSA is the time required for a bit to travel from DBCS to the MCS. So it can be expressed as $\frac{\textbf{Distance } (D_{MCS} - D_{DBCS})}{\textbf{Propagation speed } (PS_{DBCS})}$.
Transmission time for JSA can be expressed as $\frac{\textbf{Message size } (M_{DBCS})}{\textbf{Bandwidth } (BW_{MCS-DBCS})}$
Processing Delay for JSA can be expressed as $\frac{\textbf{No. of entries } (E_{DBCS})}{\textbf{Speed of the CPU } (SC_{MCS})}$.
Latency for RA defines how long it will take for an entire message to completely arrive at the User from the time first bit is send out from MCS. It is the combination of propagation time (Pt_{MCS}), transmission time (Tt_{MCS}), queuing time (Qt_{MCS}) and processing delay (Pd_{MCS}).
Propagation time for RA is the time required for a bit to travel from MCS to the User. So it can be express $\frac{\textbf{No. of entries } (E_{DCS})}{\textbf{Speed of the CPU } (SC_{DBCS})} + \frac{\textbf{No. of entries } (E_{User})}{\textbf{Speed of the CPU } (SC_{DCS})}$
Transmission time for RA can be expressed as
$\frac{\textbf{Message size } (M_{MCS})}{\textbf{Bandwidth } (BW_{MCS-DBCS})} + \frac{\textbf{Message size } (M_{DBCS})}{\textbf{Bandwidth } (BW_{DBCS-DCS})} + \frac{\textbf{Message size } (M_{DCS})}{\textbf{Bandwidth } (BW_{User-DCS})}$
Queuing Time for WA is the time needed for each Intermediate or end device to hold the message that is for the fixed agent (Main_JSA, Base_JSA) it is $\mathbf{2(n-1)\ (S_{User}+T_{User})\ (S_{DCS}+T_{DCS})}$
Processing Delay for WA can be expressed as
$\frac{\textbf{No. of entries } (E_{DCS})}{\textbf{Speed of the CPU } (SC_{DBCS})} + \frac{\textbf{No. of entries } (E_{User})}{\textbf{Speed of the CPU } (SC_{DCS})}$

8. Conclusions and Future Scope

Cloud computing is the latest trend in the computing industry. In spite of so many solutions for the problem in cloud computing environment, still there are challenges for interoperability of cloud services. This paper consists of architecture of cloud computing which is based on multiple agents' services. The load on MCS is considerably reduced as all the users' requests are not directed, but are rescheduled before MCS. The idea behind the lock on a data file in this architecture can be extended also to acquire lock and reschedule. The job can be completed in more efficient and effective manner as there are n numbers of fixed and mobile agents. Depending on the bandwidth available, the size of the message can be controlled which reduces the overall communication cost. The algorithm in the paper can be extended for read lock access with better conflict management. This architecture can also be modified to reduce number of messages passing.

9. References

i. Gaoyun Chan, Jun Lu, Jian Huang, Zexu Wu "SaaAS - The Mobile Agent based Service for Cloud Computing in Internet Environment"-Natural Computation (ICNC), 2010 Sixth International Conference on 10-12 Aug. 2010.
ii. Zhenyu Liu, Tiejiang Liu, Tun Lu, Lizhi Cai, Genxing Yang "Agent-based Online Quality Measurement Approach in Cloud Computing Environment"- 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology
iii. Meiko Jensen, Jorg Schwenk, Nils Gruschka, Luigi Lo Iacon "On technical Security Issues in Cloud Computing"- 2009 IEEE International Conference on Cloud Computing.
iv. Zehua Zhang and Xuejie Zhang "Realization of Open Cloud Computing Federation Based on Mobile Agent"- Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on 20-22 Nov. 2009.
v. "Introduction to cloud computing architecture" –white paper, Sun Microsystems. June 2009
vi. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, Above the Clouds: A View of Cloud Computing, Communications of the ACM, April 2010.