CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 Cloud Computing

Cloud computing is the delivery of computing as a service rather than a product, where by shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet).

Cloud computing at the present is an open research field which has not been explored much. Cloud computing is a comprehensive solution that delivers IT as a service. It is an internet based computing solution where resources are shared.

Cloud computing provides computation, software applications, data access, data management and storage resources without requiring cloud users to know the location and other details of the computing infrastructure.

End users access cloud based applications through a web browser or a light weight desktop or mobile app while the business software and data are stored on servers at a remote location. Cloud application providers strive to give the same or better service and performance than if the software programs were.

1.2 Characteristics

Cloud computing exhibits the following key characteristics

Empowerment of end-users of computing resources by putting the provisioning of those resources in their own control, as opposed to the control of a centralized IT service (for example)

Agility improves with users' ability to re-provision technological infrastructure resources.

Application programming interface (API) accessibility to software that enables machines to interact with cloud software in the same way the user interface facilitates interaction between humans and computers. Cloud computing systems typically use REST-based APIs.

Cost is claimed to be reduced and in a public cloud delivery model capital expenditure is converted to operational expenditure. This is purported to lower barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation (in-house).

Virtualization technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another.

Multi-tenancy enables sharing of resources and costs across a large pool of users thus allowing for:

> **Centralization** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
>
> **Peak-load capacity** increases (users need not engineer for highest possible load-levels)
>
> **Utilization and efficiency** improvements for systems that are often only 10–20% utilized.

Reliability is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

Scalability and elasticity via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis near real-time, without users has to be engineered for peak loads.

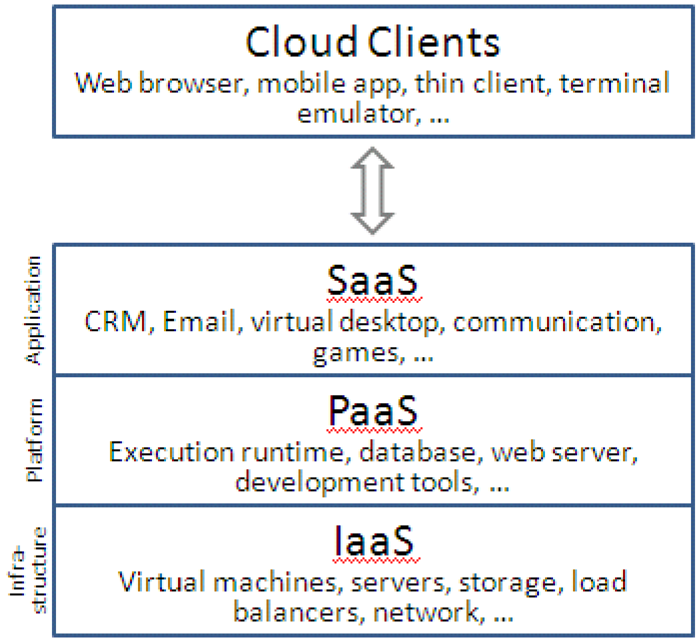
Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.

Security could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users.

Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

1.3 SERVICE MODELS

Infrastructure as a service (IAAS), platform as a service (PAAS), and software as a service (SAAS) .where IAAS is the most basic and each higher model abstracts from the details of the lower models.



Infrastructure as service (IAAS)

In this most basic cloud service model, cloud providers offer computers – as physical or more often as virtual machines , raw (block) storage, firewalls, load balancers, and networks. IaaS providers supply these resources on demand from their large pools installed in data centres. Local area networks including IP addresses are part of the offer. For the wide area connectivity, the Internet can be used or - in carrier - dedicated virtual private networks can be configured.

To deploy their applications, cloud users then install operating system images on the machines as well as their application software. In this model, it is the cloud user who is responsible for patching and maintaining the operating systems and application software. Cloud providers typically bill IaaS services on a utility computing basis, that is, cost will reflect the amount of resources allocated and consumed.

Platform as a Service (PAAS)

In the PAAS model, cloud providers deliver a computing platform and/or solution stack typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PAAS offers, the underlying compute and storage resources scale automatically to match application demand such that the cloud user does not have to allocate resources manually.

Software as a Service (SAAS)

In this model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. The cloud users do not manage the cloud infrastructure and platform on which the application is running. This eliminates the need to install and run the application on the cloud user's own computers simplifying maintenance and support. What makes a cloud application different from other applications is its elasticity. This can be achieved by cloning tasks onto multiple virtual machines at run-time to meet the changing work demand, balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user who sees only a single access point. The pricing model for SAAS applications is typically a monthly or yearly flat fee per user.

1.4 CLOUD CLIENTS

Users access cloud computing using networked client devices, such as desktop computers, laptops, tablets and Smartphone. Some of these devices - cloud clients - rely on cloud computing for all or a majority of their applications so as to be essentially useless without it. Examples are thin clients and the browser-based Chrome book. Many cloud applications do not require specific software on the client and instead use a web browser to interact with the cloud application.

With Ajax and HTML5 these Web user interfaces can achieve a similar or even better look and feel as native applications. Some cloud applications, however, support specific client software dedicated to these applications (e.g., virtual desktop clients and most email clients). Some legacy applications (line of business applications that until now have been prevalent in thin client Windows computing) are delivered via a screen-sharing technology.

1.5 TYPES OF CLOUDS

Public cloud

Applications, storage, and other resources are made available to the general public by a service provider. Public cloud services may be free or offered on a pay-per-usage model. There are limited service providers like Microsoft, Google etc owns all Infrastructure at their Data Centre and the access will be through Internet mode only. No direct connectivity proposed in Public Cloud Architecture.

Community cloud

Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.

Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models.

Private cloud

Private cloud is infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management, essentially lacking the economic model that makes cloud computing such an intriguing concept.

1.6 CLOUD ARCHITECTURE

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud communicating with each other over a loose coupling mechanism such as a messaging queue. Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms such as these and others. The Intercloud is an interconnected global "cloud of clouds" and an extension of the Internet "network of networks" on which it is based. Cloud engineering is the application of engineering disciplines to cloud computing. It brings a systematic approach to the high level concerns of commercialisation, standardisation, and governance in conceiving, developing, operating and maintaining cloud computing systems. It is a multidisciplinary method encompassing contributions from diverse areas such as systems, software, web, performance,information, security, platform, risk, and quality engineering.

1.7 ISSUES

Privacy

The cloud model has been criticised by privacy advocates for the greater ease in which the companies hosting the cloud services control, thus, can monitor at will, lawfully or unlawfully, the communication and data stored between the user and the host company. Using a cloud service provider (CSP) can complicate privacy of data because of the extent to which virtualization for cloud processing (virtual machines) and cloud storage are used to implement cloud service . The point is that because of CSP operations, customer or tenant data may not remain on the same system, or in the same data centre or even within the same provider's cloud. This can lead to legal concerns over jurisdiction. Cloud computing poses privacy concerns because the service provider at any point in time, may access the data that is on the cloud. They could accidentally or deliberately alter or even delete some info.

Sustainability

Although cloud computing is often assumed to be a form of green computing, there is no published study to substantiate this assumption. Citing the servers affects the environmental effects of cloud computing. In areas where climate favours natural cooling and renewable electricity is readily available, the environmental effects will be more moderate. The same holds true for "traditional" data canters. In the case of distributed clouds over data canters with different source of energies including renewable source of energies, a small compromise on energy consumption reduction could result in high carbon footprint reduction.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

In cloud computing, demand for resources is dynamic. A client can send a request to the server asking for any resource at any point of time. Allocation and scheduling of the resources in an efficient manner is one of the challenges in cloud computing.

2.1 Resource Scheduling

There are many complex computations and transformations in distributed computing. There are many ways to represent these complex computations. Workflow is an emerging technique to represent and manage the complex scientific problems. The workflow of a system provides the information regarding the system for efficient computation. Each workflow is basically a directed graph which depicts the flow of execution of a system. In workflows describe each node is a data computation. The dependencies between each node are denoted by links. This decreases the complexity of the system.

2.2 Types of Workflows

- To design the abstract workflow directly according to a predefined schema

- The second method uses Chimera to build the abstract workflow based on the user-provided partial, logical workflow descriptions specified in Chimera's Virtual Data Language (VDL).

- Thirdly, abstract workflows may also be constructed using assistance from intelligent workflow editors such as the Composition Analysis Tool (CAT)

2.3 Optimizing Workflow Execution using Workflow Reconstructing

Once the target systems for a portion of the workflow are identified, there are still optimizations that can be taken into consideration to improve the overall workflow performance. These involve restructuring the workflow so that the jobs can be run on the systems as efficiently as possible.

One possible restructuring aims at increasing the granularity of computation and thus reduces the scheduling overhead. The second type of restructuring involves scheduling jobs onto multi-processor systems. On these systems it may be more efficient to request more than one processor at a time, since the delay in the scheduling queues may be significant. If there are multiple tasks scheduled for the multiprocessor system, it may be beneficial to cluster them together and run them as one schedulable unit. In general the amount of decisions one would like to make in order to optimize workflow execution is great so in practice, in current workflow mapping systems, only a subset of decisions is taken into account at any one time.

2.4 Algorithm Selection

The scheduling algorithm chosen for the project is Best Resource Selection (BRS) Algorithm. In BRS, work schedule tasks on resources based on the earliest finish time. The resources are selected based on the performance.

Ant Colony Optimization Vs Particle Swarm Optimization techniques

There are many optimization techniques available out of which ACO and PSO are more popular and efficient techniques in distributed computing.

Ant Colony Optimization

Ant colony optimization is modeled based on the real ant system. ACO is more applicable for problems which have predefined sets of input and output. This technique is used where specific results are needed.

Particle Swarm Optimization

It is a clustering algorithm which is based on the social behavior of the animals or insects. It is purely a mathematical concept. PSO is based on gradient based algorithms. It is more applicable in case of dynamic and constraint handling. It is more suitable for multi-objective problems where results need not be specific .
Since we are concerned with a wide variety of scheduling tasks, we have chosen particle swarm optimization technique for computational efficiency.

2.5 Particle Swarm Optimization Technique

Particle Swarm Optimization technique is based on swarm intelligence. The basic PSO algorithm is as follows:

1. Initialize the parameters with random numbers.
2. Calculate pbest of each particle i.e., the minimum fitness value of each particle(local best)
3. Calculate the gbest in the set of particles i.e., the minimum of the pbest.
4. Calculate the velocity of the particle and update the position.
5. Repeat from the step 2 till the criteria are satisfied.
6. In the basic algorithm, if we alter or change the initialization of some of the parameters, we can improvise of the efficiency of the algorithm.

The computational efficiency of the PSO can be improved thus delivering a proper scheduling of resources.

CHAPTER 3

ANALYSIS OF PROBLEM

3. ANALYSIS OF PROBLEM

3.1 System Analysis

3.1.1 Analysis of existing system

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources. To provide cloud computing services economically it is important to optimize resource allocation, assuming that the resources are taken from a pool of available resources. In addition ,to be able to provide computing resources, it is necessary to allocate bandwidth to access them at the same time. Resource allocation is an important problem in cloud because when a user gives a particular task to be performed without the proper resource allocation the service provided by the cloud may not be efficient after the resource scheduling is done there are some optimization techniques applied to make the allocation more efficient.

3.1.2 Drawbacks of existing system

In general cases mostly the resource scheduling is done but not the optimization of those resources. By applying the optimization of scheduled resources it is found that the cost can be reduced significantly. In particular particle swarm optimization (PSO) technique is being applied on the resources scheduled.

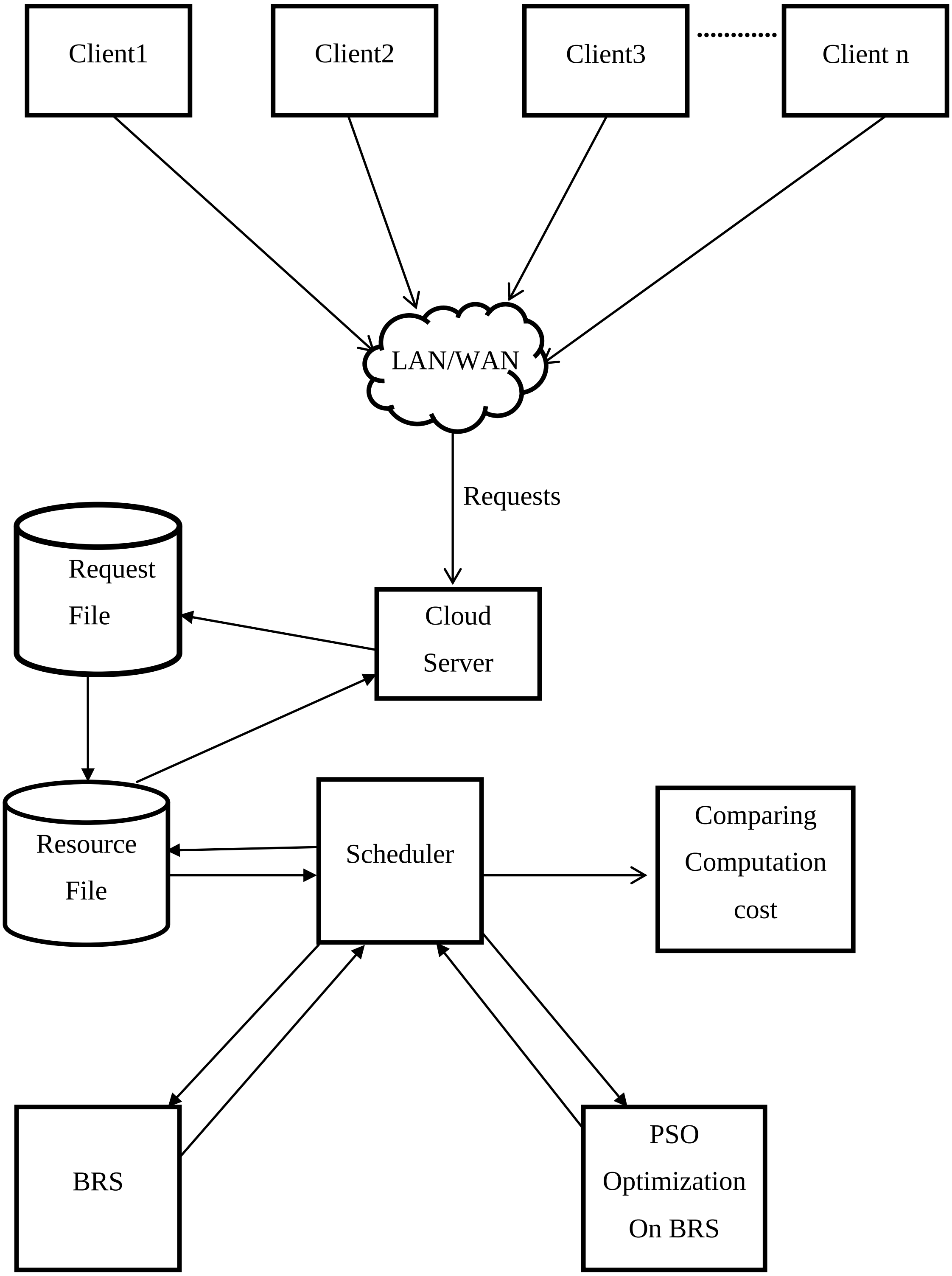
3.2 Proposed method

In the proposed method the resources are scheduled are optimally by using the PSO technique. The user and provider are benefited mutually as the resources are not being wasted and the cost for the user is reduced significantly.

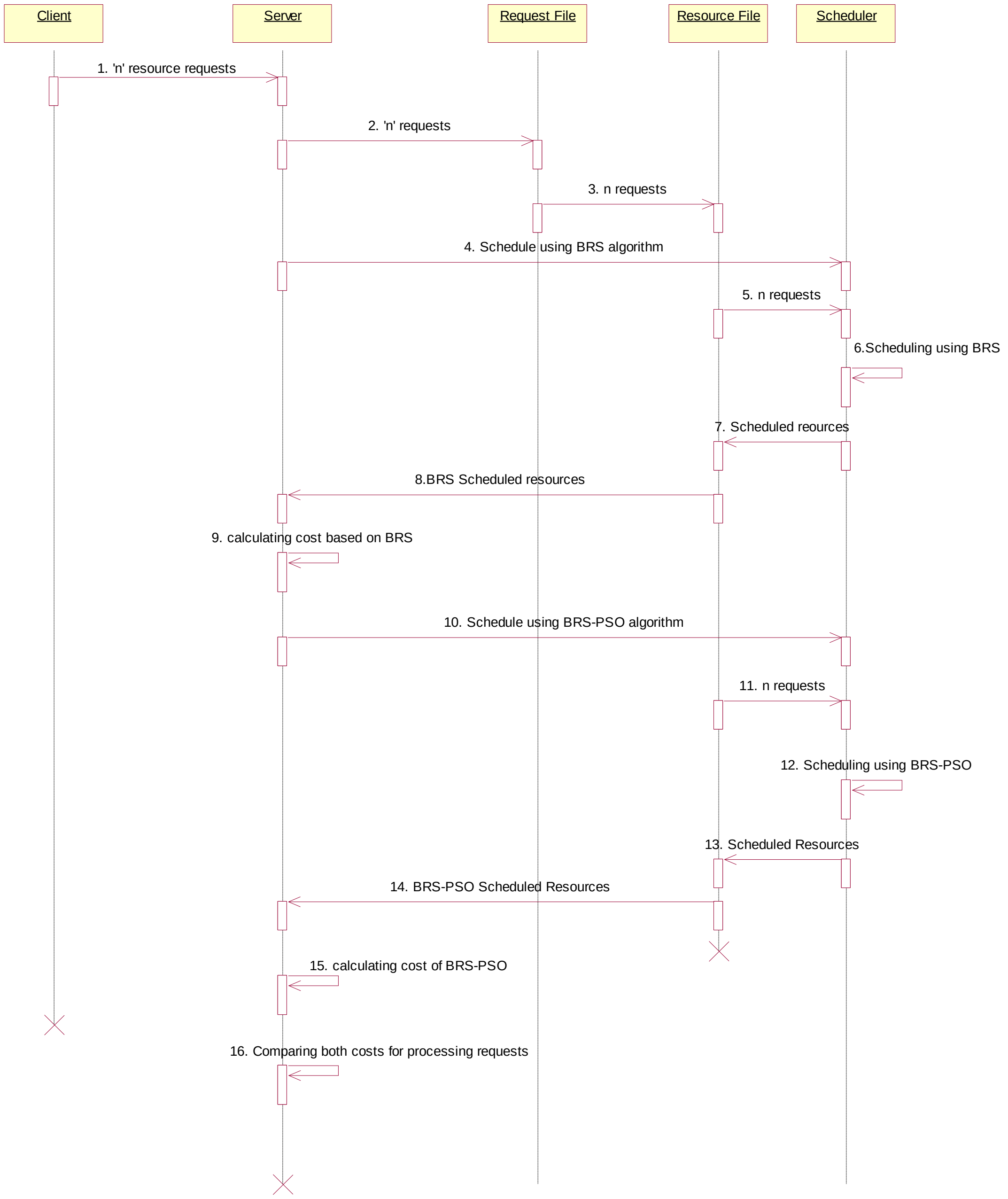
CHAPTER 4

ARCHITECTURE DIAGRAM

4. ARCHITECTURE DIAGRAM



4.1 SEQUENCE DIAGRAM FOR SERVICING THE REQUESTS



CHAPTER 5

SYSTEM IMPLEMENTATION

5. SYSTEM IMPLEMENTATION

Implementation is a phase where theoretical design is turned out into working system. Implementation includes the overall practical design of the server configuration, client configuration, and network configuration, scheduling the requests and allocation of the resources.

5.1 Environment

The system is implemented in Java. The environment used is Eclipse.

5.2 Input and Output

The inputs from the client are:

- Ram space
- Number of processing elements
- Resource utilization time
- Disk Space

The outputs from the server side are

- Scheduled requests based on the resources
- Comparison graph for BRS and PSO on BRS

5.3 Network configuration

Server and client are being connected through wireless LAN connection. Any number of clients can be connected to the server.

5.4 Server Configuration

LAN connection is created on the server side with unique IP address. Network connection is made public for the visibility of the network for all the clients.

5.5 Client Configuration

Connected to the servers network. Eclipse is installed. Servers IP address is used in the java program. A unique ID is being set for each client. Once the client is connected to the server, its ID is sent to server along with the request asked by that particular client.

5.6 Scheduling the requests

The scheduling is static. Once requested by the clients that are connected to the server, the server invokes the scheduler program. The scheduler program is implemented in java applets. The scheduling algorithm used is Best Resource Selection algorithm and the technique used for optimization is Particle Swarm Optimization.

5.6.1 Scheduling Algorithm

1: Initialize the RAM size, processing elements and disk space as defined.

2: repeat

3: Set the node T as the request from the ready queue

4: allocate the resources of T by deducting from the maximum limit

5: for all requests in the ready queue do

6: Compare the resource utilization time of the node T and other requests

7: Set T as the request which has shorter time

8: Set the other requests in the waiting queue

9: end for

10: Once the time is over for the request T, de allocate the resources.

11: Change the next request from the waiting queue to the ready queue

12: Update the ready queue list

13: until there are unscheduled requests

5.6.2 Particle Swarm Optimization

The PSO technique is implemented on the scheduling algorithm.

1: Set particle dimension as equal to the size of ready queue.

2: Initialize particles position randomly.

3: Initialize gbest randomly.

4: Compare the gbest with the pbest of each particle

5: Assign the gbest with the shortest pbest.

6: Update the position of the particle whose pbest is assigned as gbest.

7: If the stopping criteria or maximum iteration is not satisfied, repeat from step 4

5.7 Comparison Graph

Once the requests are received by the server, the requests are categorized based on the memory size. The categorizations are

- m1.small (memory size less than 128 MB)
- c1.medium
- m1.large
- m1.xlarge
- c1.large
- c1.xlarge

Once the scheduling is done, the computation cost is calculated for each request using the cost of that requests category and finishing time of that request. The same calculation is used for PSO on BRS scheduling. A graph is plotted using memory size along X axis and Environmental utilization along Y axis. The comparison of cost is observed.

CHAPTER 6

SYSTEM SIMULATION

6. SYSTEM SIMULATION

In the previous chapter we have seen about system implementation of the specified algorithms in detail. Cloud is a very vast domain i.e. we need large amounts of resources(RAM, Processing elements, Disk size) but amount of resources are limited so we go for a system implementation. Implementing cloud in n number systems and allocate resources efficiently using the resource scheduling algorithm and then optimizing it.

6.1 Server Configuration

In server side administrator needs to configure the virtual machine types i.e. giving amount RAM, Disk size, Processing Elements. Then administrator saves the virtual machine types to server database. After the configuration the server when run shows the available resources. The tasks are accepted and when they are run each task when run is allocated resources and when the task is completed then the resources are reallocated.

In the server side the resources that are given for each category are categorized into different types of VM types, the ram space that is given is split into different types (i.e. m1.small, m1.large, m1.xlarge, c1.medium, c1.large).Each VM type that is categorized has a particular cost pricing for it and when the user requests the type of resource the pricing is done by the environmental utilization and the time that each task that is need to be processed by the server.

6.2 Client Configuration

The client's needs to give the tasks that requires to be processed by the server and the tasks are taken and after they are processed, the results of the processing of each task is calculated and is shown in the console of the server , the clients are informed about the cost of their tasks .

In the allocation method users are asked to enter the number of resource types they require from each category, these are stored in database for cost calculation. On server side VMtypes like m1.Large are given the values in terms of resources they donate. These are used by the consumers while they place a request for the resource.

6.3 Resource Scheduling and Price Calculation

The task scheduling is done by the server using the Best Resource Selection Algorithm. After the completion of this scheduling particle Swarm Optimization technique is applied on resources of scheduled tasks. The price calculation system depends on the environmental utilization and the time required by the task for its execution in the server. The price calculation system uses the defined amount of cost for each VM types defined by the server. Different types of defined VM types have different costs these costs are defined by the size of memory allocated to the defined VM types. Once the task completes its execution the cost of the task is calculated depending upon the VM type requested by the task and the amount of time it spent in the server for execution.

6.4 Connection of a single client to server

Every client connected to the server has some unique IP address. Each client has an id as a label to their requests and are later displayed in the statistics. The requests received from a particular client are shown in the statistics view.

6.5 Inputs from client to server

The inputs from the client side to server include various resources like RAM space, number of processors, resource utilization time and disk space. Each request is processed and the type of resource that it is categorized into is shown. After every request the client is asked as to whether the requests should be submitted for scheduling. Once the client asks for scheduling the requests, all the requests read from the client and are displayed in the input view.

6.6 Input View

The resource requests from multi-clients to single server are shown in an applet which displays the request number, arrival time and resource utilization time. The applet has a dropdown menu with a choice to select the option displayed. The options BRS algorithm and optimized BRS with PSO are available in the Gantt chart selection bar. When the algorithm from dropdown menu is selected, the animation view of the algorithm scheduling the resource requests from multi-clients is shown on Gantt chart. This applet also gives us flexibility to clear the values in the applet and run, pause, and resume options of Gantt chart animation view.

6.7 Scheduling

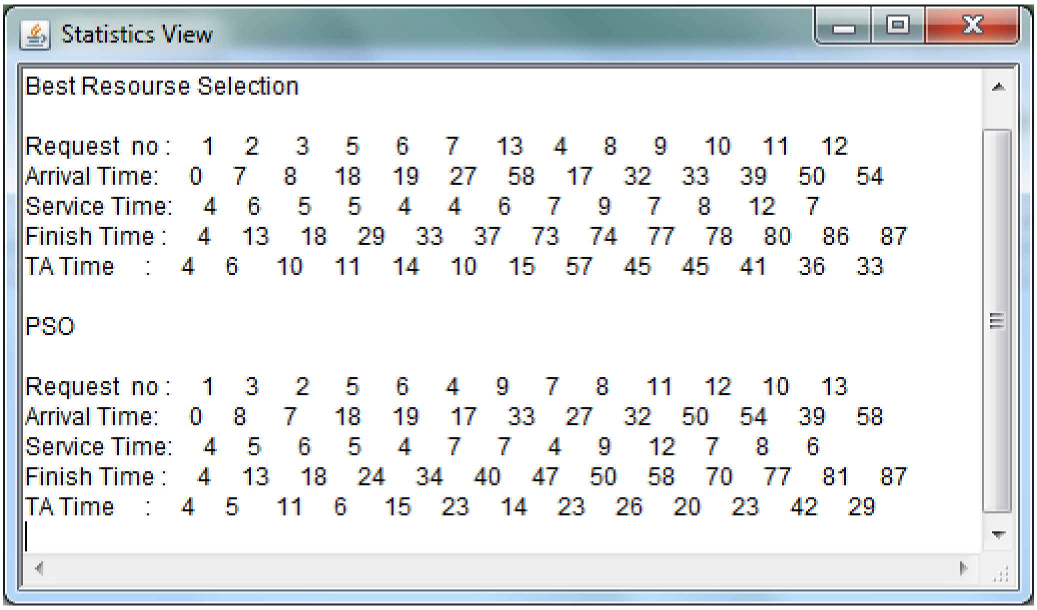
Once the request for scheduling is sent to the server, the scheduling is performed and the progress shown in console and as well as animation view. The scheduling is based on the input given in the input view to schedule the resource requests. The allocation and de-allocation of resources are being shown in console. The resources de-allocated are allocated back. The allocation is done according to the scheduled requests.

6.8 Gantt Chart

The scheduling of requests is shown in the Gantt chart. The process is shown by taking the timeline, starting from 0, along X axis and request number along Y axis. The time taken for each request to utilize the resources is shown while scheduling.

6.9 Statistics

Statistics includes request numbers, arrival times, service times requested, finishing times and turn-around times.



Statistics View

Best Resourse Selection

Request no :	1	2	3	5	6	7	13	4	8	9	10	11	12
Arrival Time:	0	7	8	18	19	27	58	17	32	33	39	50	54
Service Time:	4	6	5	5	4	4	6	7	9	7	8	12	7
Finish Time :	4	13	18	29	33	37	73	74	77	78	80	86	87
TA Time :	4	6	10	11	14	10	15	57	45	45	41	36	33

PSO

Request no :	1	3	2	5	6	4	9	7	8	11	12	10	13
Arrival Time:	0	8	7	18	19	17	33	27	32	50	54	39	58
Service Time:	4	5	6	5	4	7	7	4	9	12	7	8	6
Finish Time :	4	13	18	24	34	40	47	50	58	70	77	81	87
TA Time :	4	5	11	6	15	23	14	23	26	20	23	42	29

6.10 Graph Plot

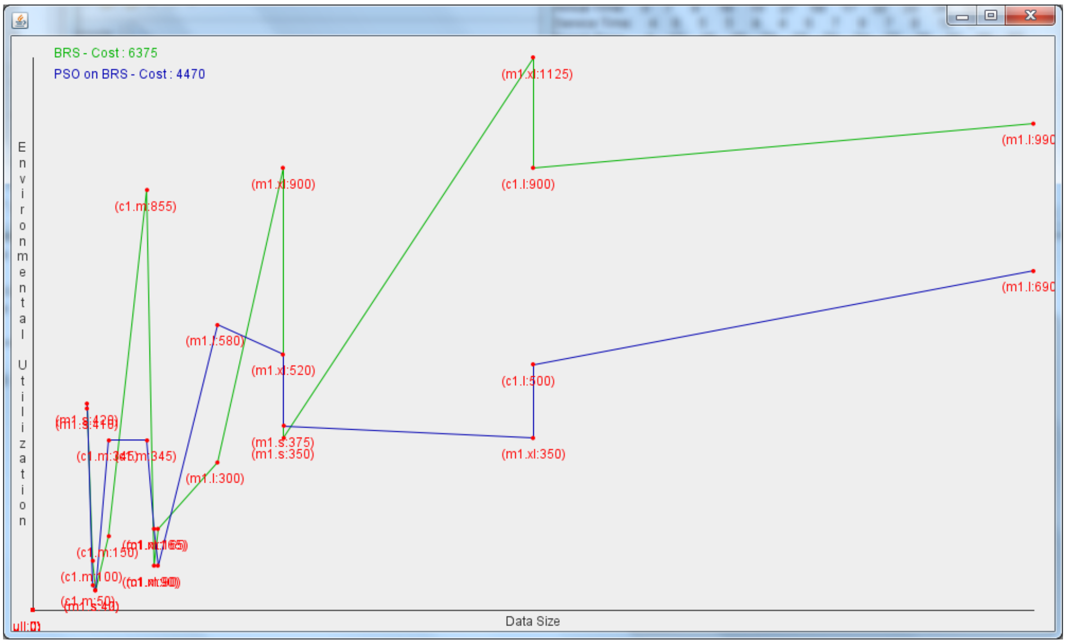
The plotted graph is based on costs calculated for requests that are scheduled based on BRS algorithm and the PSO optimized algorithm.

The difference is shown in two colors.

Green - BRS algorithm

Blue - PSO – BRS algorithm

The data size is plotted on X-axis and environment utilization is plotted on Y-axis.



*************************SUMMARY **************************

Request	Client ID	Type of Res	BRS-Cost(Rs)	PSO on BRS-Cost(Rs)
1	3	m1.small	410	420
2	2	c1.medium	100	50
3	1	m1.small	40	40
4	1	c1.medium	150	345
5	3	c1.medium	855	345
6	2	m1.xlarge	90	165
7	1	c1.medium	165	90
8	3	m1.large	300	580
9	2	m1.xlarge	900	520
10	1	m1.small	350	375
11	3	m1.xlarge	1125	350
12	2	c1.large	900	500
13	1	m1.large	990	690
			---------	---------
		TOTAL	6375	4470

6.11 System Specification

6.11.1 Hardware specification

- Microprocessor : Intel Core i3 Processor or higher
- Base Memory : 4 GB RAM
- Disk memory : 500GB

6.11.2 Software specification

- Operating system : Windows XP or higher or Linux
- Environment : Eclipse Helios service release 2
- Programming Language : Java

CHAPTER 7

TESTING

7. TESTING

7.1 Unit Testing

Unit testing is done to check whether a module works as required with different set of data's. It is mainly used to differentiate valid data types, range of values from all available possibilities.

Module

Server Capacity

Test Cases Checked

The Server is checked for its performance by giving the maximum load by connecting maximum number of clients and by giving parallel inputs from each of the clients(in this experiment maximum of 20 clients can be connected to the server).

Module

Client Inputs.

Test Cases Checked

Clients side checking is done by giving excess amount of resource requests than the resources already present in the server. The server returns a message to the client asking the client to give appropriate number of resources.

Module

Effectiveness of the PSO optimized algorithm.

Test Cases Checked

After all the inputs are given to the server, it goes to the scheduling part. Here in scheduling the inputs are arranged and are executed depending on the amount of resources requested by the clients, the cost varies according to the environment utilization. When the number of requests by the clients to the server is increased, the server throughput increases and hence the performance of the PSO optimization is effective.

CHAPTER 8

CONCLUSION

8. CONCLUSION

Resource allocation problem has been an important topic in the fields of networking, parallel and distributed computing. Cloud computing enables enterprises to reserve group of resources, use them to establish common platform and run computation tasks. In this project we have worked on the issue of optimizing the scheduling of resources deciding how many and what kind of resources need to be reserved for these computations after the computations have been linearized into the form of a sequence of tasks. This algorithm is implemented on server system to reduce time as well as cost of the processes and to allow a higher degree of resource sharing. We compare the cost savings when using PSO and existing 'Best Resource Selection' (BRS) algorithm. Results show that PSO can achieve very good cost savings as compared to BRS, and a good distribution of workload onto resources.

CHAPTER 9

REFERENCES

9. REFERENCES

[1] Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya. A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments.

[2] Y. Gil, E. Deelman, M. Ellisman, T.Fahringer, G. Fox, D. Gannon, C. Goble, M.Livny, L. Moreau, and J. Myers. Examining the challenges of scientific workflows. Computer,40(12), 2007.

[3] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu. Cloud Computing and Grid Computing 360-Degree Compared.

[4] Gerhard Venter (gventer@vrand.com) ∗Vanderplaats Research and Development, Inc. Particle Swarm Optimization. AIAA 2002–1235.

[5] Yuhui Shi and Russell C. Eberhart. Parameter Selection in Particle Swarm Optimization.

[6] Christian Vecchiola, Michael Kirley, and Rajkumar Buyya. Multi-Objective Problem Solving With Offspring on Enterprise Clouds.

[7] V.Selvi, Dr.R.Umarani. Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques. International Journal of Computer Applications (0975 – 8887) Volume 5– No.4, August 2010.

[8] I.Foster, et al., "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," Proceedings of Scientific and Statistical Database Management, 2002.

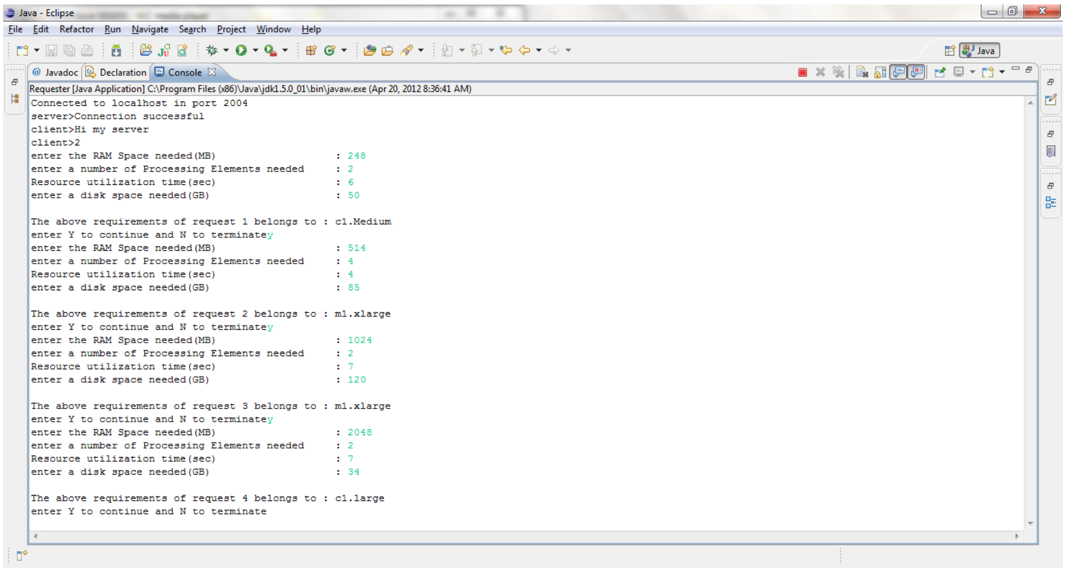
[9] T. Sousa, A. Silva, and A. Neves. Particle swarm based data mining algorithms for classification tasks. Parallel Computing,30(5-6):767–783, 2004.

[10] K. Veeramachaneni and L. A. Osadciw. Optimal scheduling in sensor networks using swarm intelligence. 2004.

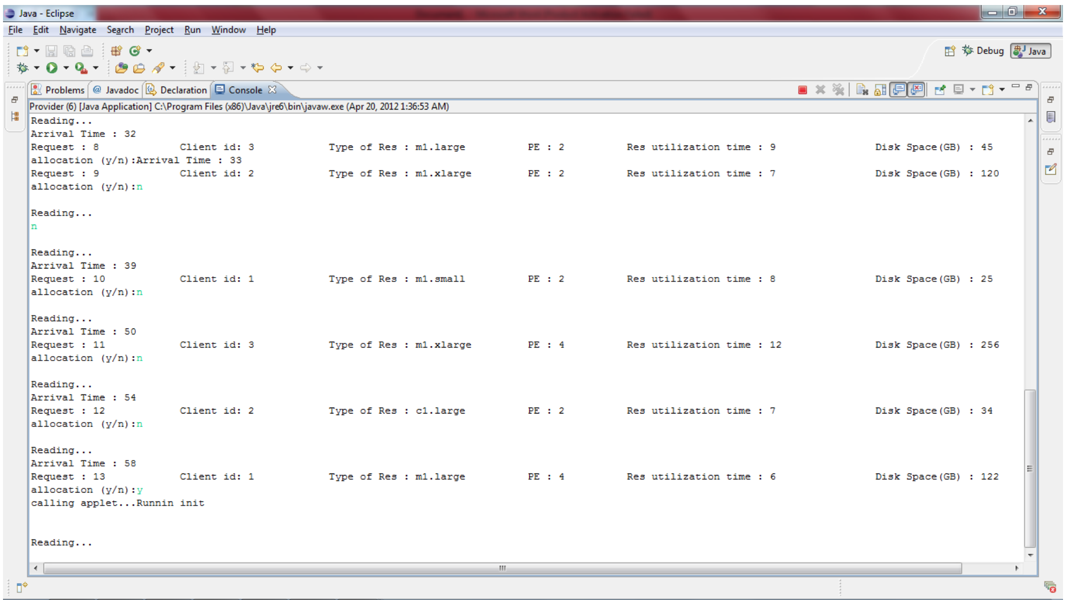
APPENDIX

APPENDIX

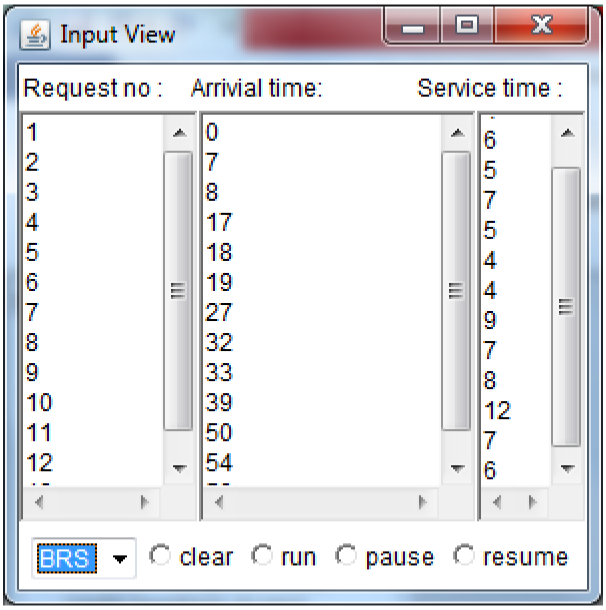
10.1 Console input at client system

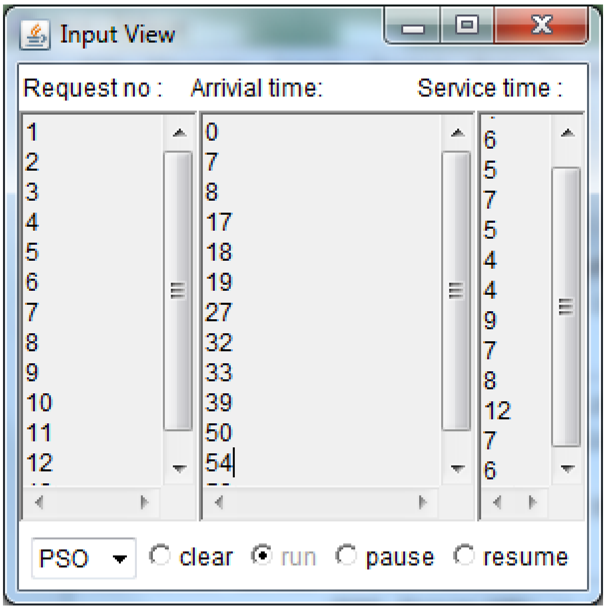


10.2 Console read of inputs from multi-clients in server

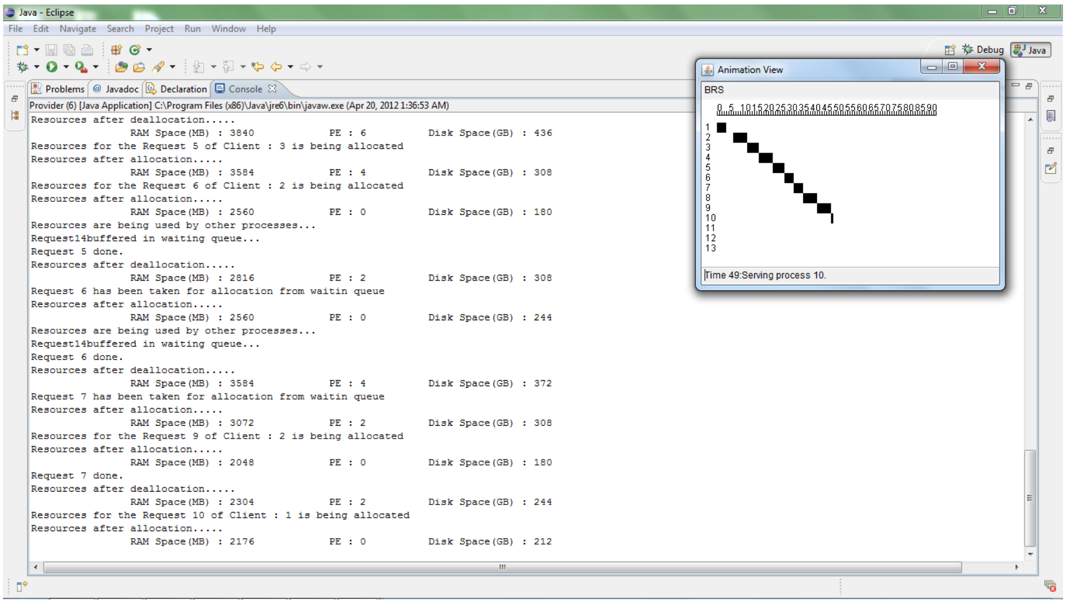


10.3 Input view of resource requests and dropdown menu for choosing algorithm to schedule

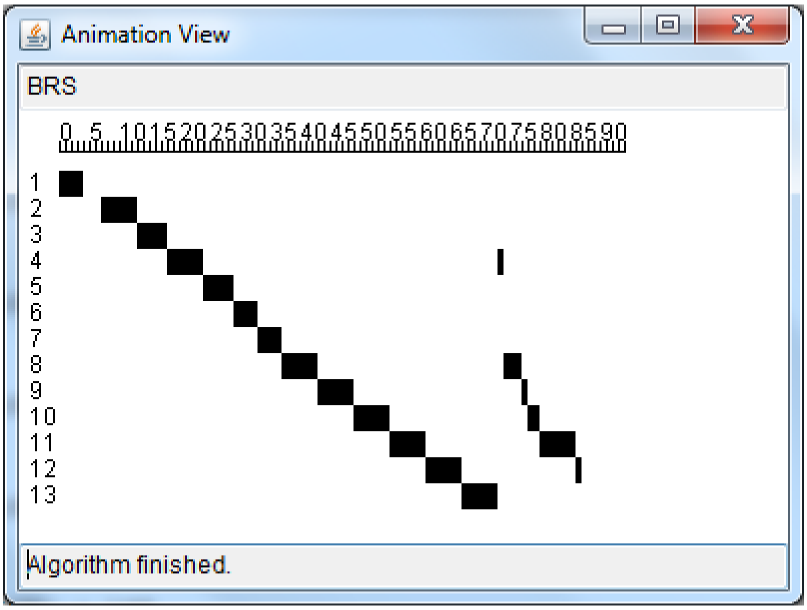


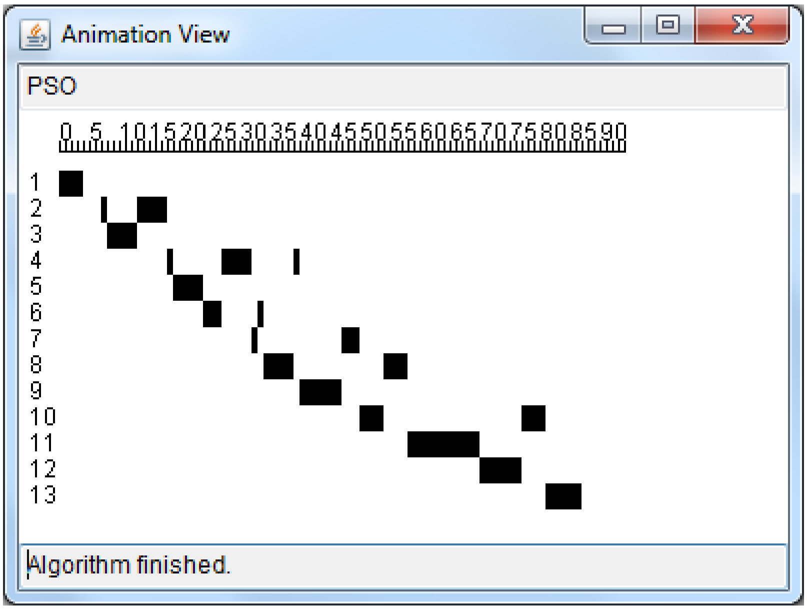


10.4 Scheduling of resource requests based on choosed algorithm in console



10.5 Animation view of algorithm implementation on requests





10.6 Statistics after the implementation of algorithm

Statistics View

Best Resourse Selection

Request no :	1	2	3	5	6	7	13	4	8	9	10	11	12
Arrival Time:	0	7	8	18	19	27	58	17	32	33	39	50	54
Service Time:	4	6	5	5	4	4	6	7	9	7	8	12	7
Finish Time :	4	13	18	29	33	37	73	74	77	78	80	86	87
TA Time :	4	6	10	11	14	10	15	57	45	45	41	36	33

PSO

Request no :	1	3	2	5	6	4	9	7	8	11	12	10	13
Arrival Time:	0	8	7	18	19	17	33	27	32	50	54	39	58
Service Time:	4	5	6	5	4	7	7	4	9	12	7	8	6
Finish Time :	4	13	18	24	34	40	47	50	58	70	77	81	87
TA Time :	4	5	11	6	15	23	14	23	26	20	23	42	29