

Grid Computing Using Instant Messaging

Field of the Invention

5 **[0001]** The present invention relates to a system for sharing computing resources in a grid network using Instant Messaging ("IM").

Background

10 **[0002]** A conventional grid computing allows members to share computing resources in a network of computers in a distributed and coordinated manner. Accessing the grid network is usually done at a user's workstation, either via a command line- or web-based interface. FIG. 1 shows a conventional grid computing system 10 having a plurality of Resource Nodes 20 and a Supervisory Node 30. The resource capability and utilization
15 of each node 20,30 is shown in an Attribute Repository 40. Typically, a user of a Resource Node 20 composes the job specification in accordance with a computing language and syntax. The job specification is then sent to a Resource Broker 32 engine in the Supervisory Node 30 for the job to be submitted by a Job Submitter 34 engine to a suitable Resource Node 20, 30. In addition, the job is monitored by a Job Monitor 36
20 engine. Both the Job Submitter 34 and Job Monitor 36 engines reside in the Supervisory Node 30. The user is, therefore, unaware whether he/she is using a computer or data at one's resource node or that at a remote resource node. In other words, the complexity of sharing computing resources in a grid network is hidden from the users; there is no intuitive means for users to assess or monitor the grid network utilization before
25 submitting a job.

[0003] Further, there is no way to determine whether a Resource Node in the grid has failed. For example, if one of the node connectivity is temporarily affected but a job has been successfully completed, the status of the job is deemed to have failed.

30

[0004] An Instant Messaging ("IM") system is a client-server system. The IM server tracks the presence information of IM clients that an IM client has subscriptions with.

The IM system thus links these IM clients together and allows the IM clients to communicate with one another. FIG. 2 depicts typical IM Clients 51, 52, 53 connected to an IM Server 50. The IM Server 50 is connected to a Directory/Authentication Server 60. The Directory/Authentication Server 60 runs a Roster engine 62. The Roster engine 62 manages transactions and associated roster subscription details of the IM clients connected to the IM Server 50. The roster subscription details of the IM clients are stored in a Repository 64. For example, as shown in FIG. 2, the IM Client 51 connects up with the IM Server 50 and notifies the IM Server 50 that it is connected. The IM Server 50 returns a list of contacts to IM Client 51 together with their presence information from the Repository 64. The contact list contains the contact details of those clients who have subscribed to both the IM Server 50 and the IM Client 51. The IM Client 51 then sends out its presence information to the IM Server 50 stating that it is "available". The IM Server 50 updates its Repository 64 and simultaneously sends the presence information of IM Client 51 to all clients that have subscribed to the IM Client 51. For example, as shown in a Roster 66 in FIG. 2, IM Client 52 has IM Client 51 in its contact list (but not IM Client 53), the IM Server 50 would send the presence information of IM Client 51 to IM Client 52 only.

[0005] Conventional Instant Messenger (IM) systems, such as MSN, Yahoo, AOL and so on, can therefore provide computer users who have subscribed to and are connected on-line to such IM servers to "chat" or communicate synchronously. Some of these messaging may not be routed through the respective IM server but directly with other IM clients 51,52,53 by communicating through the use of Peer-to-Peer protocols after the IM server(s) 50 has/have linked the IM clients together. In addition, these conventional IM systems can also provide asynchronous communication. Hence, an IM system can provide a suitable platform for implementing a grid computing network where the IM clients are Resource nodes or computing machines.

[0006] FIG. 3 shows a conventional IM system being hosted on an internet gateway 70. As depicted in FIG. 3, the IM clients 51,52, etc. are linked to the internet gateway 70 through their respective Internet Service Provider (ISP) servers 80.

[0007] It can thus be seen that there exists a need for a new system for implementing grid computing submission, monitoring and scheduling using Instant Messaging in a manner that can minimise if not overcome the limitations of conventional grid computing systems.

5

Summary

[0008] In one embodiment of the present invention, a method of networking a plurality of computing resource nodes and a plurality of IM clients (users) in a grid using Instant
 10 Messaging is provided. The method comprises: connecting each of the plurality of resource nodes and each of the plurality of IM clients (users) to an Instant Messaging (IM) server; and installing a listing engine in each IM client (user) terminal such that the listing engine provides a contact list of on-line resource nodes; wherein a job request is operable to be submitted by selecting a resource node or a group of resource nodes from
 15 the contact list. In one embodiment, the contact listing comprises: node status; node resource utilization; and job status of each of said plurality of resource nodes.

[0009] In another embodiment of the present invention, a grid computing system using Instant Messaging is provided. The system comprises: a plurality of Resource nodes and
 20 a plurality of IM clients (users); wherein an IM client (user) terminal is operable to run a listing engine for displaying node status, node utilization, and job status of each of the plurality of Resource nodes in a contact list; and an instant Message (IM) is operable to be sent by the IM client (user) by selecting a resource node or a group of resource nodes from the contact list.

25

[0010] In one embodiment, the listing engine further comprises a brokering engine. When a job request is submitted to a group selected from contact listing, a predetermined resource node from the group acts as a supervisory node for splitting the job requests into a job units according to the number of resource nodes selected or the number
 30 specified in a job specification. The supervisory node then consolidates the completed job result from each of the job units; for sending an IM message to the requestor IM

client (user) informing it of job completion; and for transferring the consolidated job result to the requestor IM client (user).

5 **[0011]** In another embodiment of the system, it further comprises a client authorisation server. The client authorisation server allows registration of IM clients before having access to the grid computing system; this server further allows registration of IM clients with different access control levels and rights.

10 **[0012]** In yet another embodiment of the present invention, a computer program stored on a computer readable medium for launching an Instant Messaging (IM) program for a grid computing network is provided. The computer program comprises: a registration engine operable with a client authorisation engine; and a contact listing engine, which comprises a brokering engine.

15

Brief Description of the Drawings

20 **[0013]** This invention will be described by way of non-limiting embodiments of the present invention, with reference to the accompanying drawings, in which:

20

[0014] FIG.1 illustrates a prior art grid computing system;

[0015] FIG.2 illustrates a prior art Instant Messaging architecture;

25 **[0016]** FIG. 3 illustrates a prior art Instant Messaging system hosted an internet gateway;

[0017] FIG. 4A illustrates an Instant Messaging system according to one embodiment of the present invention; FIG. 4B illustrates another Instant Messaging system according to another embodiment of the present invention;

30

[0018] FIG. 5 illustrates a dialogue box when a new Resource node is added according to one embodiment of the Instant Messaging system of the present invention;

[0019] FIG. 6 illustrates a contact list according to another embodiment of the Instant Messaging system of the present invention;

- 5 [0020] FIG. 7 illustrates a process of submitting a job request according to another embodiment to the Instant Messaging system of the present invention;

[0021] FIG. 8 illustrates a screen shot of job monitoring according to another embodiment of the present invention;

10

[0022] FIG. 9A illustrates another screen shot of job monitoring according to another embodiment of the present invention; and FIG. 9B illustrates a screen shot of a job completed according to another embodiment of the present invention; and

- 15 [0023] FIG. 10 illustrates a processing of a job with priority given to a resource node's own job request according to another embodiment of the present invention.

Detailed Description

20

[0024] Systems and methods for implementing Instant Messaging (IM) in a virtual environment are described. In the following description, details are provided to describe specific and alternative embodiments of an IM system. It shall be apparent to one skilled in the art, however, that the invention may be practised without some of these details.

- 25 Some of the details may not be described at length so as not to obscure the present invention.

[0025] FIG. 4A shows an architecture of an Instant Messenger (IM) system 100 forming a grid computing network according to an embodiment of the present invention. The IM system 100 has an Instant Messaging (IM) server 110; an Authentication server 120 connected to the IM server 110; a Repository 130 connected to the Authentication server 120; and Resource nodes (IM clients) 151, 152, 153, etc. and IM clients (user) 161, 162,

30

etc. connected to the IM server 110. As shown in FIG. 4A, the Repository includes a Roster 140, which contains all the Resource nodes' (IM clients') and IM clients' (users') subscriptions to the IM server 110. In one embodiment of the IM system 100, all the Resource nodes (IM clients) 151, 152, etc. and IM clients (users) 161, 162, etc. have
 5 subscription with each other. Each Resource node (IM client) 151, 152, 153, etc. is connected to the IM system 100 via a background service process and does not require human intervention or a user terminal; the background service process connects a Resource node (IM client) to the IM system 100 each time the computing machine is started.

10

[0026] FIG. 4B shows another architecture of an Instant Messenger (IM) system 100a according to another embodiment of the present invention. The IM system 100a is different from the previous embodiment 100 in that the Resource nodes (IM clients) 151, 152, 153, etc. are additionally connected to a Client Authorisation server 190 and the
 15 Resource nodes form a grid computing system as a subnet of a large IM network. In the IM system 100, 100a, the Resource nodes (IM clients) 151, 152, 153, etc. are computing machines but the IM clients (users) 161, 162, etc. are individual IM users and grid computing users who can submit jobs to a Resource node.

20 **[0027]** In another embodiment of the IM system 100, 100a, the IM server 110 is a cluster of servers instead of a single server unit. In one embodiment, each of the servers in the cluster has its own repository. In another embodiment of the IM system, the IM system 100, 100a is hosted on an internet gateway.

25 **[0028]** In IM system 100, once an IM client (user) 161, 162, etc. has successfully signed-on to the IM system 100, the user is allowed to submit jobs to one or more Resource nodes 151, 152, 153, etc.. In IM system 100a, each IM client (user) 161, 162, etc. is required to register with the IM system 100a; the registration data are stored in the Client Authorisation server 190, for example, in a Grid Repository 192 connected therewith. In
 30 another embodiment, the Client Authorisation server 190 additionally allows an IM client (user) 161, 162, etc. to register with a choice of a number of access control levels or rights when using the IM system 100a; access control levels and rights include the

number of nodes a job request may be sent to or the size of a job request. In yet another embodiment, expiry and renewal data are also stored in the Repository 130 or Client Authorisation server 190.

5 **[0029]** In one embodiment of registering with the IM system 100a, registration is carried out via on-line registration. In another embodiment, registration is carried out via email. If a registration process is unsuccessful, an IM client (user) would not have access to the IM system 100a and, therefore, would not be able to submit a job request to the grid computing system associated with the IM system 100a.

10

[0030] Each IM client (user) 161,162, etc. installs listing engine 170 into one's computer. The listing engine 170 provides a contact list 172 showing the status of all the available or active Resource nodes 151, 152, etc. that are on-line by using the log-in data kept in the Repository 130 of IM system 100 or the Client Authorisation server 190 of
15 IM system 100a. An IM client (user) 161,162, etc. is then able to submit a job request to the grid computing system via the IM system 100,100a by selecting a Resource node or group of Resource nodes from the contact list 172. In one embodiment, the listing engine 170 is installed when sign-on or registration to the IM system 100, 100a is successfully completed.

20

[0031] In another embodiment, the listing engine 170 prompts an IM client (User) 161, 162, etc. that a new Resource node 151, 152, etc. is added to the IM system 100,100a after a new sign-in or registration is made. An example of a dialogue prompt 174 from the listing engine 170 is shown in FIG. 5. As shown in FIG. 5, an IM client (user)
25 161,162, etc. who is on-line can either choose to allow the new Resource node to be added to one's contact list or not. If the user's choice is positive, data in the Repository 130 or Client Authorisation server 190 is updated, and the user's contact list 172 would indicate the newly added Resource node as "available" or "on-line"; if the user's choice is negative, data in the Repository 130 or Client Authorization server 190 is not updated;
30 the user's contact list 172 would not list the newly added Resource node or would indicate it as "not available". FIG. 6 shows a typical contact list 172a of an IM client (user) 161,162, etc.

[0032] The contact list 172 in FIGs. 4A or 4B also shows the status of utilization and job status of each Resource node 151, 152, etc. in the IM system 100, 100a. The status of utilization of each available Resource node 151, 152, etc. is indicated by the CPU usage, RAM size and available data storage or disc capacity, whereas the status of a job at each Resource node is indicated by a job identification number if it is processing a job. In addition, the status of each Resource node 151, 152, etc. is colour coded. For example, when a Resource node is "available" or on-line, it is indicated by a green smiley icon; if a Resource node is not available or off-line, it is indicated by a grey button. In addition, a Resource node is also identified if it is a Resource broker. In another embodiment, a Resource node is identified by its node numbering and broker label.

[0033] In another embodiment of the listing engine, the listing engine 170 includes a brokering engine 171. The brokering engine 171 allows a Resource node 151, 152, etc., for example a host node, to be additionally given brokering function in a similar manner as the resource broker in the prior art grid network. In another embodiment, each IM client (user) 161, 162, etc. is allowed to be additionally given the brokering function. When a group of Resource nodes is selected to execute a job, a predetermined Resource node in the group of selected nodes acts as a supervisory broker. An advantage of the IM system 100,100a according to the present invention is that a Resource node having brokering function is able to broker a job it is handling or assigned to to other Resource nodes, for example, when demand on its resources changes, resulting in it not meeting the job specification.

[0034] In one embodiment of the present invention, a job specification includes an executable application and defines the computing resources, the number of Resource nodes required to execute a job, the input data file(s) and a job expiry time; the computing resource definition includes the operating system, processor type, random access memory (RAM), data storage capacity, and so on.

[0035] FIG. 7 shows a process 200 in submitting a job request to a group of Resource nodes according to an embodiment of the present invention. As shown in FIG. 7, the

process 200 starts with step 210. From step 210, the process 200 progresses to step 220. In step 220, an IM client (user) 161, 162, etc. selects a group of Resource nodes 151, 152, etc. from one's IM contact list 172, 172a. In the next step 230, the user submits one's job request via one's IM window, for example, by attaching the job specification, and submitting it to the group of selected Resource nodes for execution. In step 240, the group of selected Resource nodes received the job request and a selected Resource node, designated as a supervisory broker, splits the job request into smaller job units according to the number of nodes specified in the job specification and/or number of selected Resource nodes. For example, when 10 Resource nodes are selected to perform a job involving image rendering of 1000 frames, each selected Resource node would then render 100 frames of the image. The job units are then brokered out among the selected Resource nodes for processing in step 250. FIG. 8 shows an IM window 252 of a selected Resource node participating in a job execution. When the job is completed, the supervisory broker consolidates all the completed job units together and sends the completed job result, in step 260, back to the IM client (user) 161, 162, etc. that issued the job. FIG. 9A shows an IM window 262 of the supervisory broker (i.e. gridnode3) transferring the result of a completed job. FIG. 9B shows a window 264 of the requestor IM client (user) 161, 162, etc. acknowledging receipt of the completed job result from the supervisory broker (i.e. gridnode3). Once the completed job result is received by the requester IM client (user), the process 200 ends in step 290.

[0036] In another embodiment of the job submission process 200, a predetermined Resource node from the group of selected Resource nodes acts as the supervisory broker, instead of the first selected Resource node.

25

[0037] FIG. 10 shows a re-brokering process 300 of a job that is being processed according to an embodiment of the present invention. As shown in FIG. 10, a job request according to the present invention starts with step 305. In the next step 310, an IM client (user) 161, 162, etc. selects a Resource node 151, 152, etc. from one's IM contact list 172, 172a to process the job. In step 315, the IM client (user) submits the job request via one's IM window. In response, the selected Resource node acknowledges receipt of the job request, in step 320, and proceeds to execute the job.

30

[0038] In one embodiment of the present IM system 100, 100a, an owner of a Resource node 151, 152, etc. is given priority over other IM clients (users) 161, 162, etc. according to owners' policy. As a result, a job running at a Resource node can be interrupted by its owner's job request; this is shown in step 325. The brokering engine 171 of the selected Resource node then has to check the computing resources of all on-line Resource nodes that meet the job specification and to broker the job out to another Resource node in step 330.

[0039] If a new Resource node is found to meet the job specification, the new Resource node is delegated, in step 335, to take over the job. In step 340, the selected Resource node then transfers the job specification to the delegated Resource node. In step 345, the delegated resource node acknowledges receipt of the job specification and proceeds to execute the job. Once the job is completed by the delegated Resource node, the result of job completed is transferred back to the selected Resource node in step 350. In turn, the selected Resource node acknowledges receipt of the completed job result from the delegated node, in step 355, and relays the completed job result to the requestor IM client (user) 161, 162, etc.. Once, the requestor IM client (user) 161, 162, etc. acknowledges receipt of the completed job, the job request process 300 then ends in step 390.

[0040] If another Resource node 151, 152, etc. cannot be found, the selected Resource node sends an IM message to the requestor IM client (user) 161, 162, etc., in step 360, and enquire whether the user is willing to wait for the resources of the selected node to be available again. If the user is not willing to wait, the user cancels the job request, in step 365, and proceeds to end the job process 300 in step 390. If the user is willing to wait, the job request is kept pending, in step 370, for the owner's job to be completed at the selected Resource node. Upon complete execution of the job request by the selected Resource node, in step 375, the job completed result is sent back to the requestor IM client (user) 161, 162, etc. before the process 300 ends in step 390.

[0041] In the process of submitting a job specification from the requestor IM client (user) or transferring a completed job result to the requester IM client (user), input or output data are being transferred by a conventional IM file transfer engine. With the file transfer engine, an IM client (user) 161, 162, etc. who is off-line, for example, at the end
5 of a work day, the completed job result is sent by the file transfer engine to a storage unit associated with the IM server 110; the completed job result is then sent to the requestor IM client (user) 161, 162, etc. when the user next logs in the IM system 100, 100a. In addition, the file transfer engine allows IM clients (users) 161, 162, etc. to send IM messages and job requests to Resource nodes (151, 152, etc.) which are off-line and yet
10 allow the off-line Resource nodes to receive messages/job requests when the computing machines are re-started. If an IM client (user) 161, 162, etc. specifies a job expiry time in a job specification and the job request is not processed within the job expiry time, the predetermined supervisory Resource node 151, 152, etc. sends an IM message to notify the requestor IM client (user) 161, 162, etc. that the job request has not yet been
15 executed; if the requestor IM client (user) 161, 162, etc. is off-line, the IM message or job result output file is sent to the file transfer engine. In a similar manner, if an off-line Resource node 151, 152, etc. is selected to execute a job, a job request is stored in a queue by file transfer engine.

[0042] Another advantage of the present invention is that the IM system 100,100a of the present invention allows IM clients (users) 161, 162, etc. to view the grid computing resources anywhere and anytime so long as connectivity to the IM system is available. In addition, job requests can be sent and completed job results received anywhere and anytime at the convenience of an IM client (user). Even when the listing engine 170 is
25 not installed at a workstation, an IM client (user) 161, 162, etc. can still manually add a Resource node 151, 152, etc, into one's contact list 172, 172a and send a job request to each manually selected Resource node 151, 152, etc.

[0043] While specific embodiments have been described and illustrated, it is understood
30 that many changes, modifications, variations and combinations thereof could be made to the present invention without departing from the scope of the invention.

CLAIMS:

1. A method of networking a plurality of computing resource nodes and a plurality of IM clients (users) in a grid using Instant Messaging, the method comprising:
 - 5 connecting each of the plurality of resource nodes and each of the plurality of IM clients (users) to an Instant Messaging (IM) server; and
 - installing a listing engine in each IM client (user) terminal such that the listing engine provides a contact list of on-line resource nodes;
 - wherein a job request is operable to be submitted by selecting a resource node or
 - 10 a group of resource nodes from the contact list.
2. A method according to claim 1, wherein the contact list comprises:
 - node status;
 - node resource utilization; and
 - 15 job status of each of said plurality of resource nodes.
3. A method according to claim 2, wherein the node status comprises an icon, which is also colour coded.
- 20 4. A method according to any one of the preceding claims, wherein the listing engine further comprises a brokering engine.
5. A method according to claim 4, wherein each of the plurality of resource node is operable to broker and delegate a job to another resource node that meets a job
 - 25 specification.
6. A method according to claim 5, wherein the job specification comprises an executable application and defines computing resources, a number of resource nodes to execute the job, input data file(s) and a job expiry time.
 - 30
7. A method according to claim 6, wherein the computing resources comprises

the operating system, processor type, random access memory (RAM) and data storage capacity.

8. A method according to any one of the preceding claims, wherein a job request is submitted to a select group of resource nodes, the job request is split into a number of job units according to the number of selected nodes or the number of nodes specified in the job specification.
9. A method according to claim 8, wherein a predetermined resource node in the select group of nodes acts as a supervisory broker for splitting the job request into job units.
10. A method according to claim 8 or 9, wherein a predetermined resource node in the select group of nodes act as a supervisory broker for consolidating the completed job result from each of the job units; sending an IM message to the requestor IM client (user) informing it of job completion; and transferring the consolidated job result to the requestor IM client (user).
11. A method according to any one of claims 6-10, wherein the input file(s) of the job specification or job result is transferred via a file transfer engine.
12. A method according to claim 11, wherein an IM message, job specification or job result is sent by the file transfer engine to a queue list if the relevant resource node or IM client (user) is off-line, and the file transfer engine re-sends the IM message, job specification or job result from the queue list to the respective resource node or IM client (user) when the computing machine at the resource node is re-started or the IM client (user) next logs-in to the IM system.
13. A method according to any one of the preceding claims, further comprising registering each of the plurality of resource nodes and each of the plurality of IM clients (users) with a client authorisation server.

14. A method according to claim 13, wherein the client authorization server is operable to register each IM client (user) with different access control levels and rights.
15. A grid computing system using Instant Messaging (IM), said system comprising:
5 a plurality of Resource nodes and a plurality of IM clients (users);
wherein an IM client (user) terminal is operable to run a listing engine for displaying node status, node utilization, and job status of each of the plurality of Resource nodes in a contact list; and an instant Message (IM) is operable to be sent by the IM client (user) by selecting a resource node or a group of resource nodes from the
10 contact list.
16. A system according to claim 15, wherein the listing engine comprises a brokering engine.
- 15 17. A system according to claim 14, wherein each of the plurality of resource nodes is operable to broker and delegate a job to another resource node that meets a job specification.
18. A system according to claim 17, wherein the job specification comprises an
20 executable application and defines computing resources, a number of resource nodes to execute the job, input data file(s) and a job expiry time.
19. A system according to any one of claims 15-18, wherein a job request is submitted to a select group of resource nodes, the job request is split into a number of
25 job units according to the number of selected nodes or the number of nodes specified in the job specification.
20. A system according to claim 19, wherein a predetermined resource node in the select group of nodes also acts as a supervisory broker which is operable for splitting the
30 job request into job units.

21. A system according to claim 19 or 20, wherein a predetermined resource node in the select group of nodes act as a supervisory broker operable for consolidating the completed job result from each of the job units; for sending an IM message to the requestor IM client (user) informing it of job completion; and for transferring the consolidated job result to the requestor IM client (user).
22. A system according to any one of claims 15-21, further comprising a client authorisation server to which the plurality of Resource nodes and the plurality of IM clients (users) are connected to.
23. A system according to claim 22, wherein each of the plurality of IM clients (users) registers with the client authorisation server before having access to the grid computing system.
24. A system according to claim 23, wherein the client authorization server is operable to allow each IM client (user) to register with different access control levels and rights.
25. A computer program stored on a computer readable medium for launching an Instant Messaging (IM) program for a grid computing network, said computer program comprising:
a registration engine operable with a client authorisation engine; and
a contact listing engine, which comprises a brokering engine.