# Advancements in Distributed Computing and Internet Technologies:

## Trends and Issues

Al-Sakib Khan Pathan
*International Islamic University Malaysia, Malaysia*

Mukaddim Pathan
*Australian National University, Australia*

Hae Young Lee
*Electronics and Telecommunications Research Institute, South Korea*

Information Science
REFERENCE

Senior Editorial Director: Kristin Klinger
Director of Book Publications: Julia Mosemann
Editorial Director: Lindsay Johnston
Acquisitions Editor: Erika Carter
Development Editor: Mike Killian
Production Editor: Sean Woznicki
Typesetters: Christen Croley
Print Coordinator: Jamie Snavely
Cover Design: Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com



Library of Congress Cataloging-in-Publication Data

Advancements in distributed computing and Internet technologies: trends and issues / Al-Sakib Khan Pathan, Mukaddim Pathan and Hae Young Lee, editors.
p. cm.
Includes bibliographical references and index.
Summary: “This book compiles recent research trends and practical issues in the fields of distributed computing and Internet technologies, providing advancements on emerging technologies that aim to support the effective design and implementation of service-oriented networks, future Internet environments and building management frameworks”-- Provided by publisher.
ISBN 978-1-61350-110-8 (hardcover) -- ISBN 978-1-61350-111-5 (ebook) -- ISBN 978-1-61350-112-2 (print & perpetual access) 1. Electronic data processing--Distributed processing. 2. Service-oriented architecture (Computer science) 3. Internet. I. Pathan, Al-Sakib Khan. II. Pathan, Mukaddim. III. Lee, Hae Young, 1975-
QA76.9.D5A3443 2012
004.67’8--dc23

2011013015

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

# Chapter 18
# Decentralization in Distributed Systems: Challenges, Technologies, and Opportunities

**Mustafizur Rahman**
*The University of Melbourne, Australia*

**Rajiv Ranjan**
*The University of New South Wales, Australia*

**Rajkumar Buyya**
*The University of Melbourne, Australia*

## ABSTRACT

*In recent years, decentralization in distributed computing systems, such as Grids and Clouds has been widely explored in order to improve system performance in terms of scalability and reliability. However, the decentralized nature of the system also raises some serious challenges. This chapter discusses the major challenges of designing and implementing decentralization in Grid and Cloud systems. It also presents a survey of some existing decentralized distributed systems and technologies regarding how these systems have addressed the challenges.*

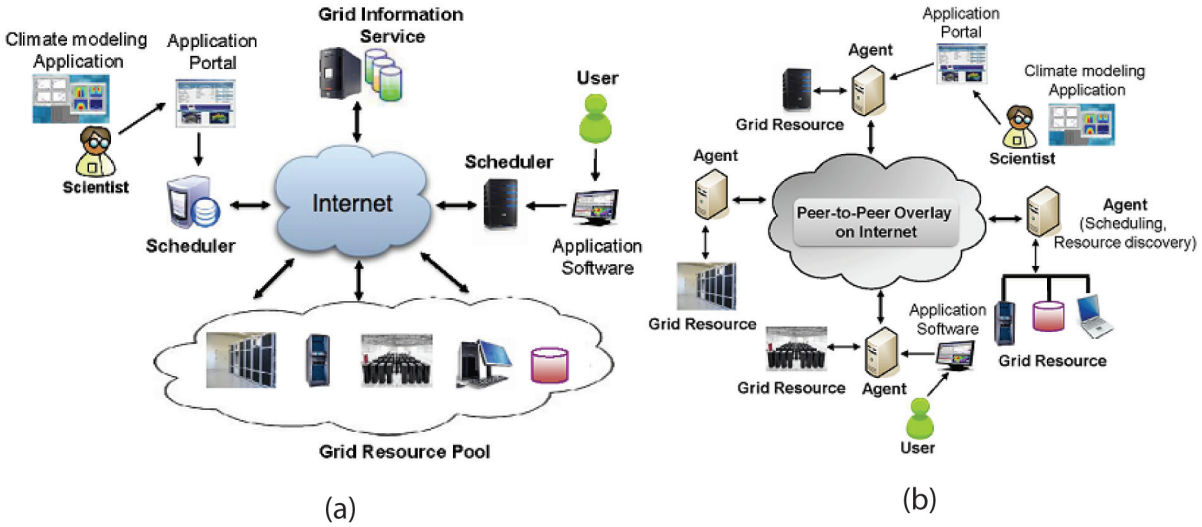
## INTRODUCTION

A distributed computing system enables the sharing, selection, and aggregation of distributed heterogeneous computational and storage resources, which are under the control of different sites or domains. The key applications of the computational distributed systems is to provide solutions to the complex scientific or engineering problems, such as weather forecasting, stock portfolio management, medical diagnoses.

The configuration of a distributed system is considered as decentralized if none of the participants in the system are more important than the others, in case that one of the participants fails, then it is neither more nor less harmful to the

DOI: 10.4018/978-1-61350-110-8.ch018

*Figure 1. Application runtime environment in centralized and decentralized distributed systems: (a) centralized system, (b) decentralized system*




system than caused by the failure of any other participant in the system. Thus, in a Decentralized distributed system, management services, such as application scheduling, resource discovery are distributed over the sites so that if one site is failed, another site can take over its responsibility autonomously. Moreover, decentralized systems are highly scalable as they can seamlessly add or remove the components or resource pool in order to accommodate varying workload. On the other hand, in a centralized distributed system, the central servers play the role of scheduling and resource discovery services. In Figure 1, we present example application runtime scenarios in case of both centralized and decentralized distributed system.

Decentralization of distributed computing systems based on Peer-to-Peer (P2P) network model can certainly overcome the limitations of centralized and hierarchical model in terms of scalability, single point failure, autonomy, and trust-worthiness. However, complete decentralized nature of the system raises other serious challenges in domains of application scheduling, resource allocation, coordination, resource discovery, security, trust, and reputation management between participants.

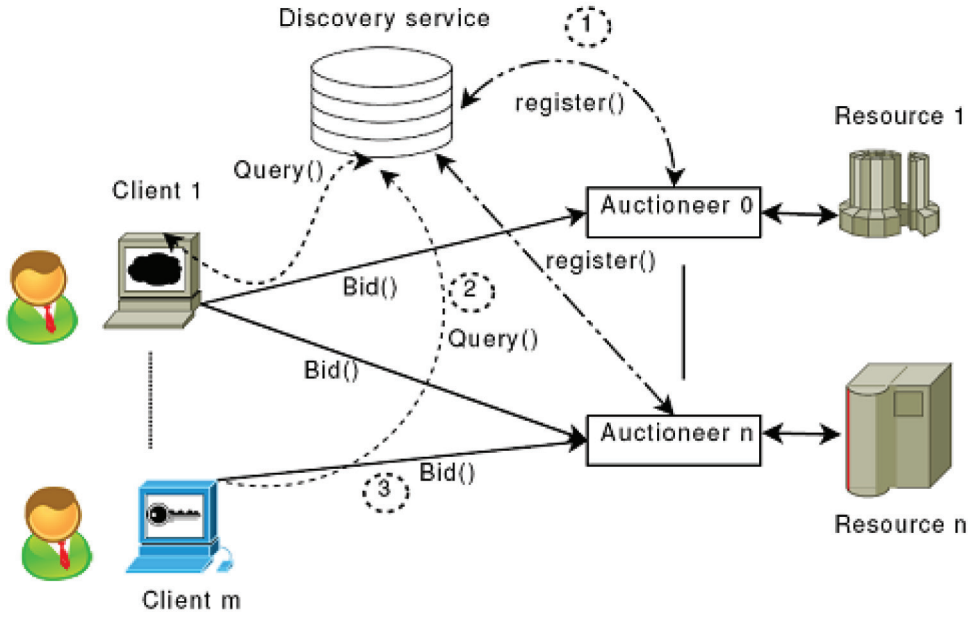
In this chapter, we aim to identify the basic challenges of decentralized distributed systems and survey some existing decentralized distributed systems and technologies along with a case study. Specifically, we describe the basic functionalities and important features of these systems and technologies, as well as compare them in the context of addressing the challenges. Finally, we outline some opportunities or future directions in this research discipline.

## CHALLENGES OF DECENTRALIZED DISTRIBUTED SYSTEMS

### Scheduling

In centralized scheduling approach, all the system-wide decision makings are coordinated by a central controller. Centralized scheduler organization is simple to implement, easy to deploy, and presents few management hassles. However, this scheme

*Figure 2. Decentralized non-coordinated scheduling in Tycoon*




raises serious concerns when subjected to larger system size.

The decentralized scheduler organization negates the limitations of centralized organization with respect to fault-tolerance, scalability, and autonomy (facilitating domain specific resource allocation policies). This approach scales well for both, a small scale resource sharing environment (e.g. resource sharing under same administrative domain) to a large scale environment (e.g. the Internet). However, this approach raises serious challenges in the domain of distributed information management, enforcing system wide coordination, security, resource consumer authenticity, and resource provider's policy heterogeneity. We can classify decentralized scheduling into two categories.

## Non-Coordinated Scheduling

In the non-coordinated scheduling scheme, application schedulers perform scheduling related activities independent of the other schedulers in the system. Condor-G resource brokering system performs non-coordinated or non-cooperative scheduling by directly submitting jobs to the condor pools without taking into account their load and utilization status. This approach exacerbates the load sharing and utilization problems of distributed resources since sub-optimal schedules are likely to occur. Figure 2 shows the decentralized non-coordinated scheduling approach in Tycoon resource sharing system. Auctioneers advertise the resource availability and configuration to the discovery service. Client agents query the discovery service to gather information about available auctioneers in the system. As a result, both Client agents end up bidding to the auctioneer n because of lack of coordination among them.

## Coordinated Scheduling

Coordinated scheduling scheme negotiates resource conditions with the local site managers in the system, if not, with the other application level schedulers. Legion-Federation system co-

ordinates scheduling decision with other sites in the distributed environment through job query mechanism. A job query request (containing job type and composition) is sent to k remote sites for bidding. The scheduler of each remote site then contacts its Local Resource Management System LRMS) to obtain job completion time on their local resources and sends this information back to the initiator's site. Finally, the site who bids with the least projected job completion time is selected for job scheduling.

## Objective Function

Resources in a distributed system are dynamic in nature and their states can change within small interval of time. Therefore, we need scheduling and resource allocation policies that can adapt to these changing resource conditions. As a result, the participants including resource providers and resource consumers associate various objective functions with respect to resource allocation and scheduling processes. These objective functions are formulated based on the policies and strategies enforced by resource providers and consumers. For example, a resource provider in a decentralized distributed system can enforce pricing policy, admission control policy, and domain specific resource allocation strategy. Similarly, the resource users or consumers can associate QoS-based utility constraints to their applications and expect that the constraints are satisfied within the acceptable limits. We can distinguish the objective functions into two categories.

## System Centric

Based on the system centric mechanism, a decentralized distributed system defines relatively simple objective functions. A system centric scheduler focuses on maximizing resource throughput on the provider side, while minimizing overall consumer's application completion time.

## User Centric

User centric scheduling mechanisms are market driven and define objective functions based on QoS parameters. From the resource providers' perspective, these QoS parameters include profit, reputation, security or combination of all, whereas QoS parameters for users are cost, budget spent, response time or combination of all.

Exact combination of QoS parameters is determined by the applied economic model. Some of the commonly used economic models in resource allocation include commodity market model, tendering/contract-net model, auction model, bid-based proportional resource sharing model, bartering model, and monopoly model. In cooperative market model, such as bartered economy, there is singleton objective function shared by both consumer and provider, which is maximizing its bartering reputation. On the other hand, in competitive market models, such as commodity market, bid-based proportional sharing, auction, resource consumer and provider usually have different objective functions. Resource providers define objective function with focus on maximizing profit, whereas consumers mainly focus on minimizing cost and response time.

## Coordination

The effectiveness of a decentralized distributed system depends on the level of coordination and cooperation among the participants. The participants in a decentralized environment are pools of diverse peers or brokers, which have agreed to co-operate for sharing and controlling resources in order to enhance overall utility of the system. Realizing such a co-operation among these dynamic and selfish participants requires robust mechanism for coordination and negotiation policies. In general, the process of coordinated application scheduling and resource management involves dynamic information exchange between various schedulers and LRMSs in the system.

Negotiation among all the participants can be done based on well-known agent coordination mechanism called contract net protocol (Smith, 1988). Contract net partitions the coordination space into two distinct domains including a manager and a contractor. A resource broker in a decentralized distributed system can adhere to the role of a contractor that negotiates SLAs with resource providers. Effectively, resource provider works as a manager that exports its local resources to the outside contractors and is responsible for decision regarding admission control based on negotiated Service Level Agreements (SLA).

However, distributed negotiation has substantial message overhead and it can worsen as system scales to a large number of participants. For the communication among the participants, we distinguish between three different approaches.

- one-to-all broadcast.
- selective broadcast.
- one-to-one negotiation.

Communication protocols based on one-to-all broadcast is very expensive in terms of number of messages and network bandwidth usage. Similar negotiation protocol has been proposed in the work Legion-Federation for decentralized scheduling. Therefore, Condor-Flock P2P system proposed selective broadcast to the flocks currently indexed by the Pastry routing table (Rowstron et al., 2001). The SLA-based scheduling approach proposed by Ranjan et al. (Ranjan et al., 2006) advocates one-to-one negotiation among contractors and managers.

Some approaches including Bellagio advocate coordinating resource activity among decentralized participants based on centralized coordinators. Figure 3 shows centralized coordination methodology applied by Bellagio system. Resource agents register the resource configuration with the Sword (Oppenheimer et al., 2005) resource discovery service. Client agents query the Sword to locate available resources in the system. Once the resource lists are obtained, Client agents bid for resources with the centralized auction coordinator. The bid parameters include the sets of resources desired, a time for which application would be deployed on resources, and the amount of virtual money clients are ready to spend.

## Security and Trust

The decentralized organization of distributed systems raises serious challenges in the domains of security and trust management. Implementing a secure decentralized distributed system requires solutions that can efficiently address the following security issues:

- preserve the privacy of participants.
- ensure authenticity of the participants.
- provide robust authorization.
- route messages securely between distributed services.

## Privacy

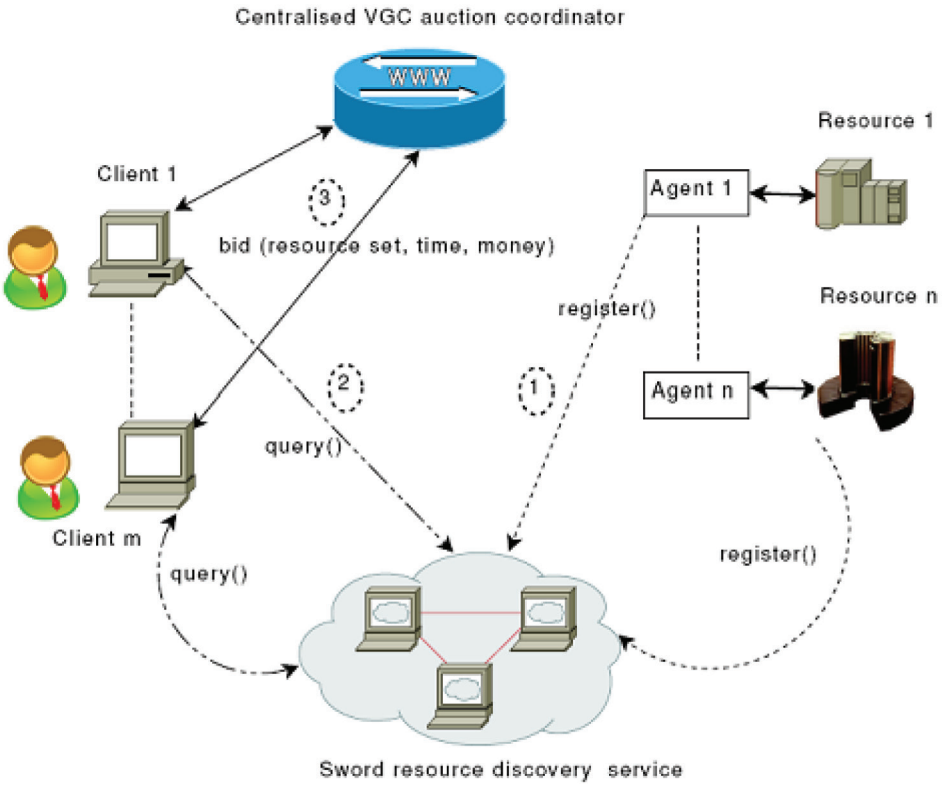
The privacy of the participants can be ensured through secret key-based symmetric cryptographic algorithms, such as 3DES, RC4, etc. These secret keys must be securely generated and distributed in the system. Existing key management systems, such as public key algorithms (including DH, RSA, elliptic) and Kerberos (trusted third party) can be utilized for this purpose.

## Authentication

Authentication of the participants can be achieved through trust enforcement mechanisms including (i) Public Key Infrastructure (X.509 certificates), (ii) Kerberos (third party authentication), (iii) distributed trust, and (iv) SSH.

Authentication based on X.509 certificates requires a trusted Certifying Authority (CA) in the system. A system can have a single CA, which is trusted by all the participants. However, single CA

*Figure 3. Centralized coordination in Bellagio*




approach has limited scalability. An alternative to this is to have multiple CAs combining together to form a trust chain. In this case, a certificate signed by any CA in the system has global validity.

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos based implementation has significant shortcomings as it requires synchronous communication with the ticket granting server in order to setup communication between a client and server. If the ticket granting server goes offline or has a security breach then there is no way the system can operate.

JXTA (Gong, 2001) provides a completely decentralized X.509 based PKI. Each JXTA peer is its own CA and issues a certificate for each service it offers. Each of the CA certificate is verified via the *Poblano: "web of trust"*, a distributed reputation management system. A similar distributed trust mechanism is PeerReview (Durschel, 2006). These distributed trust management systems determine malicious participants through behavioral auditing. An auditor node A checks if it agrees with the past actions of an auditee node B. In case of disagreement, A broadcasts an accusation of B. Interested third party nodes verify evidence, and take punitive action against the auditor or the auditee.

The SSH based authentication scheme is comparatively easier to implement as it does not require trusted third party certification. However, it does not allow the creation of a dynamic trust chain, and in case a participant's private key is compromised, it requires every public key holder to be informed about this event. Unlike X.509 and Kerberos implementation, SSH does not support

certificate translation mechanism (i.e. from X.509 to Kerberos or vice versa).

### Authorization

Authorization deals with the verification of an action that a participant is allowed to undertake after a successful authentication. Particularly in Grids, site owners have the privilege to control how their resources are shared among the participants. The resource sharing policy takes into account the participant's identity and membership to groups or virtual organizations. For instance, Globus based Grid installation defines the access control list using a Gridmap file.

### Secure Message Routing

Implementing secure and trusted message routing in decentralized environment requires solution to the following problems:

- secure generation and assignment of nodeIds.
- securely maintaining the integrity of routing tables.
- secure message transmission between peers.

Secure nodeId assignment ensures that an attacker or a malicious peer cannot choose the value of nodeIds that can give it membership of the overlay. If the node assignment process is not secure, then an attacker could sniff into the overlay with a chosen nodeId and get control over the local objects, or influence all traffic to and from the victim node. The nodeId assignment process is secured by delegating this capability to a central, trusted authority. Secure message forwarding in the Internet can be achieved through secure transport layer connections, such as TLS and SSL.

## RELATED DECENTRALIZED DISTRIBUTED SYSTEMS AND TECHNOLOGIES

Let us now look at some existing decentralized distributed systems and technologies commonly in practice. For each system or technology, we describe the basic functionalities and important features. Table 1 compares these systems and technologies in the context of how do they address the challenges of decentralized distributed system discussed above.

### Bellagio

Bellagio (Auyoung et al., 2004) is a market-based resource allocation system for federated distributed computing infrastructure. In Bellagio, users specify resources of interest in the form of combinatorial auction bids. Thereafter, a centralized auctioneer allocates resources and decides payments for users. The Bellagio architecture consists of *resource discovery and resource market*. For resource discovery of heterogeneous resources, Bellagio uses SWORD (Oppenheimer et al., 2005). For resource market, Bellagio uses a centralized auction system, in which users express resource preferences using a bidding language, and a periodic auction allocates resources to users. A bid for resource includes sets of resources desired, processing duration, and the amount of virtual currency which a user is willing to spend. The centralized auctioneer clears the bid every hour.

### CondorFlock P2P

Butt et al. (Butt et al., 2003) present a scheme for connecting existing Condor work pools using P2P routing substrate Pastry (Rowstron et al., 2001). Inherently, P2P substrate (overlay network) aids in automating the resource discovery in the Condor Flock Grid. Resource discovery in the flock is facilitated through resource information broadcast to the pools, whose ids appear in the

*Table 1. Comparison of different decentralized distributed systems and technologies*

| System Name | Type | Organization | Scheduling Model | Objective Function | Coordination Model | Security Model |
|---|---|---|---|---|---|---|
| Aneka Federation (Ranjan et al., 2009) | P2P Grid | University of Melbourne | Decentralized coordinated | System centric | Selective broadcast | Distributed trust |
| Bellagio (Auyoung et al., 2004) | Grid | University of California, San Diego | Centralized | User centric, Bid-based proportional sharing | centralized | SSH |
| CondorFlock P2P (Butt et al., 2003) | P2P Grid | Purdue University | Decentralized coordinated | System centric | Selective broadcast | PKI / Globus |
| InterGrid (Assuncao et al., 2008) | Grid | University of Melbourne | Decentralized coordinated | User centric | Selective broadcast | PKI |
| Legion-Federation (Weissman et al., 1996) | Grid | University of Virginia | Decentralized coordinated | System centric | One-to-All broadcast | Public-key cryptography based on RSAREF 2.0 |
| MOSIX-Fed (Barak et al., 2005) | Grid | Hebrew University of Jerusalem | Centralized | System centric | Centralized | SSH |
| Sharp (Fu et al., 2003) | P2P | Duke University | Decentralized coordinated | User centric, bartering | One-to-one negotiation | PKI |
| Trader-Federation (Frerot et al., 2000) | Grid | UFR Science et Techniques | Decentralized coordinated | User centric, Commodity market | One-to-All broadcast | N.A. |
| Tycoon (Lai et al., 2004) | Grid | HP Labs | Decentralized non-coordinated | User centric, Auction | One-to-All broadcast | PKI |
| Amazon EC2 (Amazon, 2010) | Cloud | Amazon.com | Centralized | User centric | Centralized | PKI |
| Azure (Nagy, 2010) | Cloud | Microsoft Corporation | Centralized | User centric | Centralized | TLS/SSL |
| Eucalyptus (Eucalyptus, 2009) | Cloud | Eucalyptus Systems, Inc. | Centralized | User centric | Centralized | SSH |

Pastry node's routing table. The proposed P2P-based overlay network facilitates only resource discovery, while other decisions such as resource sharing policy is controlled by the pool managers. Core Condor LRMS has also been extended to work with Globus (Foster et al., 1997), the new version is called Condor-G resource broker, which enables creation of global Grids and is designed to run jobs across different administrative domains.

## InterGrid

InterGrid (Assuncao et al., 2008) provides a software system that allows the creation of collaborative execution environments for various scientific applications on top of the physical infrastructure provided by the participating Grids in the federation. The allocation of resources from multiple Grids to fulfill the requirements of the execution environments is enabled by peering arrangements established between InterGrid Gateways (IGGs). An IGG is aware of the terms of the peering among the Grids connected to it. Thus, it can select the suitable Grids that are able to provide the required resources for a particular application. Moreover, it can also send request to other IGGs for resource provisioning and replies to requests from other IGGs. Request redirection policies determine which peering Grid is selected to process a request and a price at which the processing is performed.

## Legion-Federation

Weissman et al. (Weissman et al., 1996) devise a federated model for distributed cooperative resource management. The model proposes federated resource sharing using Legion LRMS. It considers two levels of application schedulers in the system namely, Local Site (LS) Scheduler and Wide-Area (WA) scheduler. Every member site has to instantiate these scheduling services. LSs are responsible for managing and controlling the set of resources assigned to them. WA scheduler has two functional components including a Scheduling Manager (SM), which is an interface to LS, and a Grid Scheduler (GS), which connects to other SMs in the federated system. The connection topology between GSs is a fully connected graph structure.

## MOSIX-Fed

MOSIX is a cluster management system that applies process migration to enable a loosely coupled Linux cluster to work like a shared memory parallel computer. Recently, it has been extended to support a Grid of clusters to form a single cooperative system (Barak et al., 2005). Basic feature of this cooperative environment includes automatic load balancing among participant clusters (owned by different owners) while preserving the complete autonomy. Proposed resource coupling scheme can be applied to form a campus or an enterprise Grid. MOSIX federation aims at hierarchical coupling of cluster resources under same administrative domain. Resource discovery in such an arrangement is facilitated by hierarchical information dissemination scheme that enables each node to be aware of the latest system wide state.

## Sharp

Sharp (Fu et al., 2003) is a framework for secure distributed resource management. In Sharp, participant sites can trade their resources with peering partners or contribute them to a peer federation according to the local site sharing policies. Sharp framework relies on bartering economy as the basis to exchange resources among various resource domains. A cryptographically signed object called Resource Tickets (RTs) is issued by each participating site. These RTs are exchanged between the participating sites for facilitating coordinated resource management. The fundamental resource management software entities in Sharp include site authority, service manager, and agents. These entities connect to each other based on a peer-to-peer network model.

## Trader-Federation

Frerot et al. (Frerot et al., 2000) present a scheme called federation of distributed resource traders, which couples various autonomous resources or resource providers. A resource trader entity acts as an intermediary between consumers and providers. Every trader has local users, clients, and resources who are members of the local resource domain. Federation of traders enables the participants to trade resources at both local and the Internet levels. Various traders cooperate within the federation to maximize a trading function. The trader presents two interfaces, local interface for its local users and resource providers, while remote interface to other traders in the federation. The federation works as a market place where various traders can negotiate for QoS parameter (response time, accuracy) requested by the local users.

## Tycoon

Tycoon (Lai et al., 2004) is a distributed market-based resource allocation system. Application scheduling and resource allocation in Tycoon is based on decentralized isolated auction mechanism. Every resource owner in the system runs its own auction for his local resources. In addition, auctions are held independently, thus clearly lacking any coordination. Tycoon system relies on centralized Service Location Services (SLS) for

indexing resource auctioneers' information. Application level super-schedulers contact the SLS to gather information about various auctioneers in the system. Once this information is available, the super-schedulers (on behalf of users) issue bids for different resources. In this setting, the super-schedulers might end up bidding for small subset of resources while leaving the rest under-utilized.

### Amazon EC2

Amazon Elastic Compute Cloud (EC2) (Amazon, 2010) is a web service that provides resizable compute capacity in the cloud environment. It's simple web service interface that provides the complete control of the leased computing resources to run on Amazon's computing environment. Resource provisioning is achieved in Amazon EC2 by utilizing three web services: Elastic Load Balancer, CloudWatch and Auto Scaling. Elastic Load Balancer is in charge of delivering incoming connections across multiple Amazon EC2 instances automatically. It continuously monitors the health conditions of instances, and re-route traffic from faulty instances to faultless instances within a single availability zone or across multiple zones. Whereas, CloudWatch, is responsible for monitoring cloud resources (i.e. Amazon EC2, Elastic Load Balancer) in real-time and provides information about the performance metrics related to the Amazon EC2 instances, such as resource utilization and network traffic.

### Azure

Microsoft Windows Azure Platform (Nagy, 2010) is a cloud platform providing a wide range of Internet services that can be consumed from both on-premises environments and the Internet. It uses a specialized operating system, called Windows Azure, to run its Fabric Layer, which provisions and manages computing and storage resources for the applications running on top of Windows Azure. Azure Fabric Controller is a redundancy tolerance service designed for monitoring and maintaining machines/resources to host the applications that are created and stored in Windows Azure. Besides, it is also in charge of resource provisioning by supporting a declarative service model. Declarative service specifications is appointed in every application and the Fabric Controller looks through Azure Fabric to match resources that meet required demands of CPU, bandwidth, operating system and redundancy tolerance.

### Eucalyptus

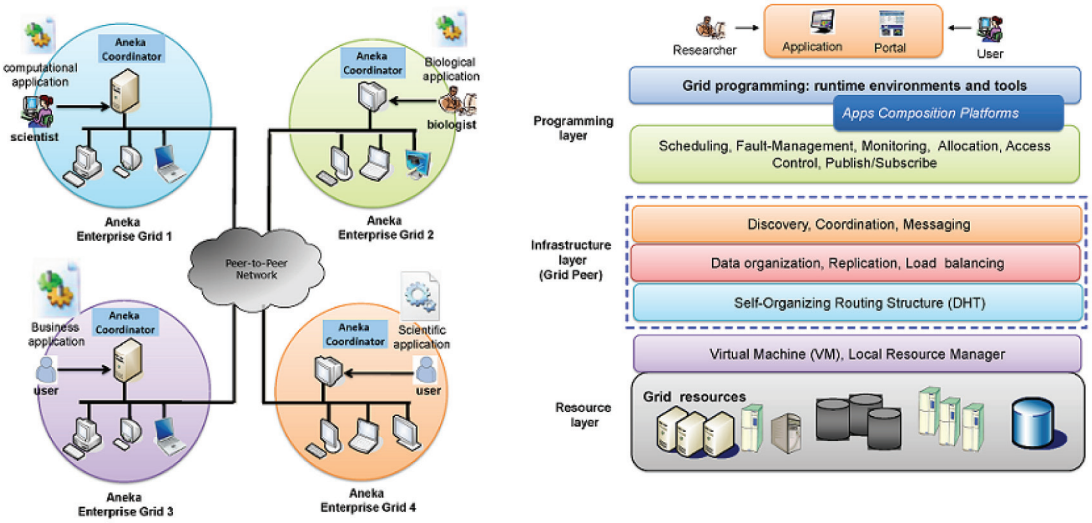
Eucalyptus Systems (Eucalyptus, 2009) is an open source software infrastructure for implementing public or private clouds on existing Enterprise IT and service provider infrastructure. Enterprise Eucalyptus provides capabilities, such as self-service provisioning, customized SLAs, cloud monitoring, metering, and support for auto-scaling, as a highly available cloud platform. It is composed of four controllers (Cloud Controller, Cluster Controller, Node Controller, and Storage Controller) to control the virtualization environment in a manner of centralized and hierarchical structure. These controllers are used for managing the underlying virtualized resources (servers, network, and storage), monitoring and scheduling Virtual Machine (VM) execution on specific nodes, hosting VMs, and interfacing with various storage systems (i.e. NFS, iSCSI).

## CASE STUDY

### Aneka Federation

Aneka Federation system logically connects topologically and administratively distributed Aneka Enterprise Grids as part of a single cooperative system. It uses a Distributed Hash Table (DHT), such as Pastry, Chord based Peer-to-Peer (P2P) network model for discovering and coordinating the provisioning of distributed resources in Aneka

*Figure 4. Aneka Federation: (a) architecture, (b) layered design*




(a) (b)

Grids. It also employs a novel resource provisioning technique that assigns the best possible resource sets for the execution of applications, based on their current utilization and availability in the system.

Aneka Federation utilizes the Grid-Federation model in regards to distributed resource organization, sharing and Grid networking. Grid-Federation is defined as a large scale resource sharing system that consists of a coordinated federation of distributed Aneka Enterprise Grids. Figure 4 shows the architecture and layered design of Aneka Federation resource sharing environment, consisting of Internet-wide distributed parallel resources in different Aneka Enterprise Grids. Every contributing site or Grid maintains its own Aneka Coordinator service and all these sites are connected through a DHT based P2P network (see Figure 4(a)).

## Scheduling

The application scheduling and resource discovery in Aneka-Federation is facilitated by a specialized Grid Resource Management System known as Aneka Coordinator (AC). AC is composed of three software entities: Grid Resource Manager (GRM), LRMS and Grid Peer. The GRM component of AC exports a Grid site to the federation and is responsible for coordinating federation wide application scheduling and resource allocation. GRM is also responsible for scheduling locally submitted jobs in the federation using LRMS. Grid peer implements a DHT based P2P overlay (see Figure 4(b)) for enabling decentralized and distributed resource discovery supporting resources status lookups and updates across the federation. It also enables decentralized inter-AC collaboration for optimizing load-balancing and distributed resource provisioning.

Grid Peer accepts two types of objects from GAM regarding decentralized and coordinated scheduling: Claim and Ticket. A Claim object is

sent by GAM to DHT overlay for locating the resources that match with user's application requirements and a Ticket is an update object sent by a Grid site, mentioning about the underlying resource conditions. These objects are also called coordination objects as they encapsulate the coordination logic in Aneka Federation.

## Coordination

Aneka Federation uses a DHT (such as Chord, Pastry) based P2P overlay for handling resource discovery and scheduling coordination. The employment of DHT gives the system the ability to perform deterministic discovery of resources and produce controllable number of messages (by using selective broadcast approach) in comparison to using other One-to-All broadcast techniques such as JXTA.

Generally, resources hosted by a Grid site are identified by more than one attribute; thereby a Claim or a Ticket object is always multi-dimensional in nature. In order to support multi-dimensional data indexing (processor type, OS type, CPU speed) over DHT overlay, Aneka Federation leverages a spatial indexing technique, which is a variant of MX-CIF Quad tree. The indexing technique builds a multi-dimensional attribute space based on the Grid resource attributes, where each attribute represents a single dimension.

## Objective Function

The main objective function employed in Aneka federation is to increase system's efficiency by balancing the load across the Grid resources in the federation, while minimizing overall user's application completion time by avoiding resource contention.

The load balancing decision is based on the principle that it should not lead to over-provisioning of resources at any Grid site. This mechanism leads to coordinated load-balancing across Aneka Federation and aids in achieving system-wide objective function, while at the same time preserving the autonomy of the participating Aneka Enterprise Grids. The process of coordinated load balancing is facilitated by implementing the P2P coordination space that takes the scheduling decisions.

## Security

Aneka Federation uses distributed trust mechanism to ensure secured resource management across the federation. It utilizes a reputation based scheduling technique implemented by the coordination space in order to prune out the malicious and unwanted users from the system. Furthermore, the Aneka Container component of AC provides the base infrastructure that consists of services for persistence and security (authorization, authentication, and auditing).

# CONCLUSION

In recent years, executing various scientific and business workflow applications in distributed systems (Grids and Clouds) has become a common practice. The inherent complexity in workflows requires an execution environment that addresses issues, such as scalability, reliability, user support, and system openness. However, the traditional centralized system for managing these workflows cannot satisfy these requirements. Thus, we can leverage the decentralized systems and technologies to achieve a better solution, given the nature of application environment.

Realizing an efficient, scalable, and robust Relational Database Management System (RDBMS) based on decentralized Grid and Cloud systems is an interesting future research problem. Fundamental to decentralized RDBMS is the development of distributed algorithms for: (i) query processing; (ii) data consistency, integrity; and (iii) transaction atomicity, durability, and isolation. First step in designing a decentralized RDBMS is to partition the relational tuple space across a

set of distributed storage resources in the system. The data partition strategy should be such that the query workload is uniformly distributed while efficiently utilizing the resources computational and network bandwidth capability.

Moreover, research in these challenging domains of decentralized workflow management and RDMS is still in early stage. We believe that applying decentralized technologies for efficient and reliable management of workflows and storage will be an area of great interest in the coming years.

## REFERENCES


Amazon. (2008). *Elastic compute cloud*. Retrieved November 15, 2010, from http:// www.amazon.com/ ec2

Assuncao, M. D., Venugopal, S., & Buyya, R. (2008, June). Intergrid: A case for internetworking islands of grids. *Concurrency and Computation*, *20*(8), 997–1024. doi:10.1002/cpe.1249

Auyoung, A., Chun, B., Snoeren, A., & Vahdat, A. (2004, October). Resource allocation in federated distributed computing infrastructures. In *Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT Infrastructure*, Boston, USA.

Barak, A., Shiloh, A., & Amar, L. (2005, May). An organizational grid of federated mosix clusters. In *Proceedings of the 5th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID),* Cardiff, UK.

Butt, A. R., Zhang, R., & Hu, Y. C. (2003). A self-organizing flock of condors. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, CA, USA.

Durschel, P. (2006, June). *The renaissance of decentralized systems*. Keynote talk at the 15th IEEE International Symposium on High Performance Distributed Computing, Paris, France.

Eucalyptus. (2009, August). *Open-source cloud computing infrastructure - An overview*. Retrieved from http:// www.eucalyptus.com/ whitepapers

Foster, I., & Kesselman, C. (1997). Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications*, *11*(2), 15–128.

Frerot, C. D., Lacroix, M., & Guyennet, H. (2000, August). Federation of resource traders in objects-oriented distributed systems. In *Proceedings of the International Conference on Parallel Computing in Electrical Engineering,* Quebec, Canada.

Fu, Y., Chase, J., Chun, B., Schwab, S., & Vahdat, A. (2003). SHARP: An architecture for secure resource peering. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles,* NY, USA.

Gong, L. (2001). JXTA: A network programming environment. *IEEE Internet Computing*, *5*(3), 88–95. doi:10.1109/4236.935182

Lai, K., Huberman, B. A., & Fine, L. (2004). *Tycoon: A distributed market-based resource allocation system. Technical Report*. USA: HP Labs.

Nagy, S. (2010). *The azure fabric controller*. Retrieved November 15, 2010, from http://azure.snagy.name/blog/?p=89

Oppenheimer, D., Albrecht, J., Vahdat, A., & Patterson, D. (2008). Design and implementation trade-offs for wide-area resource discovery. *ACM Transactions on Internet Technology*, *8*(4), 18.

Ranjan, R., & Buyya, R. (2009, July). Decentralized overlay for federation of enterprise clouds . In Li, K. C. (Eds.), *Handbook of research on scalable computing technologies*. Hershey, PA: IGI Global.

Ranjan, R., Harwood, A., & Buyya, R. (2006). SLA-based coordinated superscheduling scheme for computational grids. In *Proceedings of the 8th IEEE International Conference on Cluster Computing,* Barcelona, Spain.

Rowstron, A., & Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms,* Heidelberg, Germany.

Smith, R. G. (1988). The contract net protocol: High-level communication and control in a distributed problem solver . In Lenat, D. B. (Ed.), *Distributed artificial intelligence* (pp. 357–366). San Francisco, CA: Morgan Kaufman Publishers.

Weissman, J. B., & Grimshaw, A. (1996). Federated model for scheduling in wide-area systems. In *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing,* NY, USA.

## KEY TERMS AND DEFINITIONS

**Aneka Federation:** The Aneka Federation integrates numerous small scale Aneka Enterprise Grid services and nodes that are distributed over multiple control and enterprise domains as parts of a single coordinated resource leasing abstraction.

**Cloud Computing:** It is a market-oriented distributed computing paradigm consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers.

**Coordination:** The effectiveness of a distributed computing system often depends on the level of coordination among the distributed components of the system. Lack of coordination among the components may result in communication overhead that eventually degrades the performance of the system.

**Decentralization:** A distributed system configuration is considered to be decentralized if none of the components in the system are more important than the others, in case that one of the component fails, then it is neither more nor less harmful to the system than caused by the failure of any other component in the system.

**Distributed Systems:** A distributed system consists of a collection of autonomous computers that are connected and communicated through computer network and distribution middleware, which enable computers to coordinate their activities and share resources in the system, so that users perceive the system as an integrated single computing facility.

**Grid Computing:** Grid computing enables the sharing, selection, and aggregation of geographically distributed heterogeneous computational and storage resources, which are under the control of different sites or domains.

**P2P Computing:** Peer-to-Peer (P2P) computing is a distributed application architecture that distributes the tasks or workloads among the available peers/nodes in the network, where each peer is equally privileged and collaborate with others. The term, P2P implies that either peer can initiate a session and has equal responsibility.

**Scheduling:** In a distributed computing system, scheduling is a process of finding the efficient mapping of tasks in an application to the suitable resources in the system so that the execution can be completed with the satisfaction of objective functions, such as execution time minimization as specified by the users.