THE STATE UNIVERSITY OF ZANZIBAR

(SUZA)



**THE SCHOOL OF NATURAL AND SOCIAL SCIENCES**

**DEPARTMENT OF COMPUTER AND INFORMATION TECHNOLOGY**

**GROUP ASSIGNMENT NO ONE**

**NAME OF PARTICIPANTS REG NUMBER**

**1. FAHAD HAMAD RAMADHAN *BITAM/8/20/051/TZ***

**2. HASSAN ALI HASSAN *BITAM/8/20/001/TZ***

**3. THABIT MOHAMMED ALI *BITAM/8/20/030/TZ***

**LACTURE:** ***DR MARYAM MASOUD***

**ACADEMIC YEAR:** ***2021/2022***

**1.Grid Workflow languages and systems (suggestions: Triana, Taverna, Workflow Management System and Monitor for Grid systems (GWFE))**

Grid workflow languages and systems it is known as the formation of technologies and tools which are used to manage and make execution of complex scientific workflow on distributed infrastructures of computing such as grids and clouds.

(Suggestions)

**Triana**

By looking the context of grid workflow languages and systems, Triana refers to the graphical workflow environment allows users to design and execute scientific workflows. It provide a user-friendly interface for building complex workflow using a drag-and-drop approach and integrate different tools and services and to monitor the execution of workflows in real-time.

It is also make the execution supports of workflows on grid computing resources, permit users to have the computing power and data storage capabilities of grid infrastructures.

**Taverna**

This is among of the free and open-source scientific workflow management system which is designed to facilitate the construction, execution on grid computing resources and sharing of scientific workflows.

It allowing users permission of computer power and data storage capabilities and grid infrastructure and execution of workflows on cloud computing resources, allowing users to easily planning the workflows to manage large datasets or computationally intensive tasks.

**Workflow Management System.**

It is also known as (**WMS**) which is the component of the gLite middleware used in the European Grid Infrastructure (**EGI**), it provide an intuitive interface and efficient management of computing resources and a platform for the execution of workflows on distributed computing infrastructures and supports the integration of different workflow engines and resource managers.

**Monitor for Grid systems (GWFE)** refer to the graphical user interface for monitoring and managing workflows on Grid computing infrastructures. It provides real-time monitoring of workflow execution jobs running, job completion rates, resource utilization and enables users to view workflow progress, error rates, and manage workflow resources. Also sometime (**GWFE**) system provide information on the status and performance of the system.

Overall, a GWFE monitor can be an essential tool for system administrators and users to understand the performance and status of a grid system and to quickly identify and resolve any issues that may arise.

2. **Desktop Grids and Volunteer Computing (suggestions: BOINC and OurGrid)**

**Desktop Grids** typically involve the installation of a software program on the personal computers of volunteers that allows the program to use the idle processing power of the computer when it is not in use. This idle processing power is then combined with that of other computers to create a large computing grid, which can be used for a variety of tasks such as scientific simulations, data analysis, and cryptography.

**Volunteer Computing** on another way is more general term that refers to the use of distributed computing by volunteers to perform tasks that would require a huge amounts of computing power or resources. This may include tasks such as protein folding simulations, climate modelling and gravitational wave analysis.

Therefore these approaches can be an effective way to harness the power of distributed computing without requiring expensive hardware or dedicated infrastructure.

Overall these two term which are Desktop Grids and Volunteer Computing are two related concepts in distributed computing that involves harness the computing power of a large number of volunteer contribution over the internet to perform computationally intensive tasks.

**BOINIC**

The acronym BOINIC stands for **Berkeley Open Infrastructure for Network Computing** which is known as a software platform for volunteer computing and desktop grids. It make users to denote their idle computing resources like a CPU and GPU cycles, to scientific research projects that require massive amounts of computing power. It is an open source and has been used for many variety of scientific research projects, institutions and universities around the world.

**OurGrid**

Refer to the open-source platform for desktop grid and volunteer computing that is designed to make users to denote their idle computing resources to scientific research projects that require a huge amount of computer power. It is used client-architecture like BOINIC to manage the distribution of computational tasks to individual computers.

The OurGrid system also includes tools for managing the distribution of tasks, monitoring the status of tasks, and collecting results from volunteer computers.

**3. Grid simulators (SimGrid and GridSim)**

Grid simulators are computer programs or tools that simulate the behavior and performance of electrical power systems or grids under various conditions.

These simulators are used to model the behavior of electrical systems in order to test, analyse, and optimize their performance

**SimGrid**

Is a type of Grid simulators that used to model and simulate distributed systems including grid computing environments. It is an open-source toolkit developed by SimGrid team for Research in Computer Science and Automation.

Also SimGrid provides a set of tools and libraries for simulating distributed systems, including networks, computing resources, and storage. One of the main components of SimGrid is the SimGrid simulation kernel, which provides a simulation environment for running distributed applications.

**GridSim**

Is a discrete-event simulation toolkit that is used to model and simulate grid computing environments. It is an open-source toolkit developed by the Grid Computing and Distributed Systems Laboratory at the University of Melbourne.

GridSim provides a set of tools and libraries for simulating various aspects of grid computing environments, including resource management, job scheduling, and network communication. It allows researchers and engineers to study the behavior and performance of complex grid systems under different conditions, including varying workloads and resource availability.

4.Middleware for Grids is a software layer that provides services and interfaces to allow users to access and utilize distributed computing resources, such as clusters, supercomputers, and storage systems, as a single integrated system. Unicore and NorduGrid are two examples of middleware for grids.

UniCore is a middleware system for Grid computing, which provides a set of services and tools for managing distributed computing resources. UniCore is designed to support the development of Grid applications that require high performance computing, data management, and collaboration among distributed teams.

Some of the key competencies of UniCore middleware for Grid computing include:

* Resource Management: UniCore provides a flexible and scalable framework for managing computing resources, including processors, storage, and network bandwidth
* Security: UniCore provides strong security mechanisms to ensure the confidentiality, integrity, and availability of Grid resources and data. It supports various security mechanisms, including authentication, authorization, and encryption.
* Data Management: UniCore supports data management tasks, including data replication, data caching, and data transfer among distributed nodes. It provides efficient and reliable mechanisms for storing and accessing data across the Grid.
* Workflow Management: UniCore provides a workflow engine for automating complex Grid applications that involve multiple steps and tasks. It allows users to define, execute, and monitor workflows in a distributed environment.

Critical thinking available

* 1.Scalability: The UniCore must be able to scale to support a large number of resources and users, without compromising performance.
* Reliability: The UniCore must be reliable and fault-tolerant to ensure that users can access and utilize the resources even in the event of failures or downtime.
* Interoperability: The UniCore must be able to work with different hardware and software platforms, enabling users to access and utilize resources from diverse sources.

NorduGrid is a middleware system for Grid computing that is specifically designed for scientific applications. NorduGrid provides a set of services and tools for managing distributed computing resources and enables scientific communities to collaborate and share resources.

Some of the key competencies of NorduGrid middleware for Grid computing include:

* 1.High Performance Computing: NorduGrid provides a high-performance computing environment for scientific applications, enabling users to run large-scale simulations, compute-intensive algorithms, and data analysis tasks.
* Resource Management: NorduGrid provides a flexible and scalable framework for managing computing resources, including processors, storage, and network bandwidth.
* Data Management: NorduGrid supports data management tasks, including data replication, data caching, and data transfer among distributed nodes. It provides efficient and reliable mechanisms for storing and accessing data across the Grid.
* Collaboration: NorduGrid supports collaboration among distributed teams by providing a platform for sharing resources, data, and applications. It enables users to work together on scientific projects, share data, and collaborate on data analysis tasks.

Critical thinking available

* 1.Scalability: The UniCore must be able to scale to support a large number of resources and users, without compromising performance.
* Reliability: The UniCore must be reliable and fault-tolerant to ensure that users can access and utilize the resources even in the event of failures or downtime.
* Interoperability: The UniCore must be able to work with different hardware and software platforms, enabling users to access and utilize resources from diverse sources.
* 4.Must support high integration of different scientific tools and applications, which can promote interdisciplinary collaboration and foster creative problem-solving.

WSRF-Lite (Web Services Resource Framework Lite) is a lightweight middleware framework for building web services-based grid applications. It provides a standardized set of interfaces and protocols for managing resources in a grid environment, such as resource discovery, resource creation, and resource lifetime management.

5.Grid-aware operating systems such as XtreemOS, Vega GOS, and MOSIX are designed to support the efficient management of resources across a distributed computing environment. Competences and critical thinking are important aspects of these systems as they help to ensure that the resources are utilized optimally and that the system is operating effectively.

Competences in this context refer to the knowledge, skills, and abilities required to operate, manage, and optimize a grid-aware operating system. This includes knowledge of the underlying architecture of the system, as well as expertise in the use of the various tools and interfaces that are available for managing and monitoring the system. Critical thinking is also important in this context, as it enables operators to analyze system performance data, identify issues or bottlenecks, and develop strategies for optimizing system performance.

Some of the competences and critical thinking skills required for effective management of grid-aware operating systems include:

* Knowledge of distributed computing concepts and architectures: A deep understanding of the principles and concepts of distributed computing is critical for effective management of grid-aware operating systems. This includes knowledge of data storage and retrieval, message-passing protocols, fault tolerance, and load balancing.
* Proficiency in system administration: Effective system administration skills are essential for the management of grid-aware operating systems. This includes knowledge of system security, user management, network administration, and software installation and configuration.
* Proficiency in programming and scripting languages: Knowledge of programming and scripting languages such as Python, Perl, and Bash is essential for developing custom scripts and tools for managing and monitoring grid-aware operating systems.
* Analytical skills: Analytical skills are critical for analyzing system performance data and identifying issues or bottlenecks. This requires the ability to collect and interpret data, as well as the ability to develop strategies for optimizing system performance.
* Troubleshooting skills: Troubleshooting skills are essential for identifying and resolving issues with grid-aware operating systems. This requires a deep understanding of the system architecture, as well as the ability to diagnose and resolve problems in a timely and efficient manner.

In summary, competences and critical thinking are essential for effective management of grid-aware operating systems such as XtreemOS, Vega GOS, and MOSIX. Operators must possess a deep understanding of distributed computing concepts and architectures, proficiency in system administration and programming/scripting languages, analytical skills, and troubleshooting skills to effectively manage and optimize these systems.

6.Monitoring and accounting tools such as Ganglia, Nagios, NWS, DGAS, and Real Time Monitor are designed to provide insight into the performance of distributed computing systems, as well as track resource utilization and billing information. Competences and critical thinking are important aspects of these tools, as they enable operators to interpret the data provided by these tools, identify issues or anomalies, and take appropriate actions.

Competences in this context refer to the knowledge, skills, and abilities required to operate, manage, and optimize monitoring and accounting tools. This includes knowledge of the underlying architecture of the system, as well as expertise in the use of the various tools and interfaces that are available for managing and monitoring the system. Critical thinking is also important in this context, as it enables operators to analyze monitoring and accounting data, identify trends, and develop strategies for optimizing system performance.

Some of the competences and critical thinking skills required for effective use of monitoring and accounting tools include.

* Knowledge of distributed computing concepts and architectures: A deep understanding of the principles and concepts of distributed computing is critical for effective use of monitoring and accounting tools. This includes knowledge of data storage and retrieval, message-passing protocols, fault tolerance, and load balancing.
* Proficiency in system administration: Effective system administration skills are essential for the management of monitoring and accounting tools. This includes knowledge of system security, user management, network administration, and software installation and configuration.
* Proficiency in scripting languages: Knowledge of scripting languages such as Python, Perl, and Bash is essential for developing custom scripts and tools for managing and monitoring distributed computing systems.
* Analytical skills: Analytical skills are critical for interpreting monitoring and accounting data, identifying trends and anomalies, and developing strategies for optimizing system performance.
* Communication and collaboration skills: Effective communication and collaboration skills are important for working with other members of the distributed computing team to address issues and optimize system performance.

7. Data management and meta-scheduling are two important aspects of grid computing.   
  
Data management refers to the storage, organization, and retrieval of data in a distributed computing environment. In a grid computing environment, data may be stored in multiple locations, and may need to be accessed and manipulated by multiple users and applications. Therefore, effective data management is crucial for ensuring efficient and secure data access and sharing.  
  
Meta-scheduling, on the other hand, refers to the orchestration of computational tasks across multiple distributed resources, such as computing clusters, grids, and clouds. Meta-scheduling systems schedule and manage the execution of jobs that require access to distributed resources, while also optimizing the utilization of those resources.  
  
GridWay is an open-source meta-scheduling system that provides a framework for managing jobs on a grid computing environment. It supports multiple job submission interfaces, including command-line, web-based, and programmatic interfaces, and allows users to manage jobs across multiple computing platforms and resource managers. GridWay also provides advanced scheduling policies, such as backfilling, gang scheduling, and resource overbooking, to optimize resource utilization and improve job throughput.  
  
In addition to its meta-scheduling capabilities, GridWay also provides support for data management, including data staging and data replication. This allows users to manage their data efficiently and effectively, ensuring that it is available when and where it is needed for computational tasks.

8. Portals and friendly interfaces for job submission and control are important components of grid computing systems, as they provide users with an easy-to-use interface for accessing and controlling resources on a grid. Here are some examples of portals and friendly interfaces for job submission and control:  
  
1. Genius: Genius is a web-based portal that provides a user-friendly interface for submitting, monitoring, and managing jobs on a grid. It supports a wide range of job submission interfaces, including command-line, web-based, and programmatic interfaces, and allows users to manage jobs across multiple computing platforms and resource managers.  
  
2. UCLA Grid Portal: The UCLA Grid Portal is a web-based portal that provides a user-friendly interface for submitting and monitoring jobs on a grid. It supports job submission for a variety of scientific applications, including molecular dynamics simulations, bioinformatics, and climate modeling.  
  
3. GridSphere: GridSphere is an open-source portal framework that provides a customizable web-based interface for accessing and controlling grid resources. It supports job submission and monitoring, as well as data management and collaboration.  
  
4. PGrade: PGrade is a graphical user interface (GUI) for job submission and control on a grid. It provides a drag-and-drop interface for creating and submitting complex computational workflows, and supports a range of resource managers and job submission interfaces.  
  
These portals and friendly interfaces provide users with an easy-to-use interface for accessing and controlling grid resources, allowing them to focus on their scientific research instead of the complex details of grid computing.

9. Security is a critical concern in grid computing systems, as grids typically involve sharing resources across multiple organizations and domains. Here are two examples of security mechanisms used in grids:  
  
1. SLCS: SLCS (Simple Lightweight Certificate Service) is a lightweight certificate authority designed for use in grid computing environments. It provides a simple and secure way to issue and manage digital certificates, which are used to authenticate users and services in a grid. SLCS is designed to be easy to deploy and manage, and it supports integration with existing grid security infrastructure.  
  
2. Shibboleth: Shibboleth is an open-source software package that provides federated identity management for web applications. It allows users to authenticate themselves once with their home organization, and then access multiple web applications without having to authenticate themselves again. Shibboleth is often used in grid computing environments to provide secure access to grid resources across multiple organizations and domains.  
  
Both SLCS and Shibboleth provide mechanisms for secure authentication and authorization in grid computing environments, allowing users to access grid resources with confidence in the security of their data and applications.