Index

Annexure-I

Micro project proposal

Distributed Operating System

1. Aims/Benefits of the Micro-Project:

The main goal of a distributed system is to make it easy for users to access remote resources, and to share them with other users in a controlled manner. Resources can be virtually anything, typical examples of resources are printers, storage facilities, data, files, web pages, and networks. With resource sharing facility, a user at one site may be able to use the resources available at another. Speedup the exchange of data with one another via electronic mail.
Failure of one site in a distributed system doesn’t affect the others, the remaining sites can potentially continue operating.

2. Course Outcomes:

1) To get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.
2) Git for Distributed Software Development.
3) Distributed programming on the cloud. Carnegie Mellon University’s Cloud Developer course.

3. Proposed Methodology:

A distributed system is broadly divided into two essential concepts — software architecture (further divided into layered architecture, objectbased architecture, data-centered architecture, and event-based architecture) and system architecture (further divided into client-server architecture and peer-to-peer. Social networks, mobile systems, online banking, and online gaming (e.g. multiplayer systems) also use efficient distributed systems. Additional areas of application for distributed computing include e-learning platforms, artificial intelligence, and ecommerce.

A distributed operating system is system software over a collection of independent software, networked, communicating, and physically separate computational nodes. They handle jobs which are serviced by multiple CPUs. Each individual node holds a specific software subset of the global aggregate operating system.An architectural model of a distributed system defines the way in which the components of the system interact with each other and the way in which they are mapped onto an underlying network of computers.There are 3 types of models in the object oriented modeling and design are: Class Model, State Model, and Interaction Model. These are explained as following below. Class Model: The class model shows all the classes present in the system.

4. Action Plan:

Sr. No.	Details of Activity	Planned Start date	Planned Finish date	Name of Responsible Team Members
1	Search the information of Distributed of operating system	24-08-2022	07-09-2022	More Ganesh Vishwanath
2	Collect the information of Regarding operating system	07-09-2022	14-09-2022	
3	Analysis of different information	21-09-2022	28-09-2022	
4	Analysis of information	12-10-2022	02-11-2022	
5	Compression of operating system	09-11-2022	11-11-2022	
6	Features of operating system	16-11-2022	23-11-2022	
7	Advantages and drawback of Distributed operating sytem	30-11-2022	30-11-2022	
8	Final report of project	07-12-2022	14-12-2022	

5. Resources Required:

Sr. No.	Name of resource / material	Specification	Quantity	Remarks
1	Computer	WINDOWS 7,2GB RAM, 160GB HDD	1	
2	Operating System	WINDOWS 7	1	
3	Textbook/manual	Ubuntu Studies 22447	1	
4	Internet	Youtube / Wikipedia	1	

6.Names of Team Members with Roll No.'s:

Sr. No.	Enrollment No.	Name of Team Member	Roll No.
1	2010950104	More Ganesh Vishwanath	27

Mr. Chavan A.Y.

Name and Signature of the Teacher

Annexure – II

Micro-Project Report

Distributed Operating System

1. Rationale:

The primary purpose of a distributed operating system is to hide the fact that resources are shared. Transparency also implies that the user should be unaware that the resources he is accessing are shared. Furthermore, the system should be a separate independent unit for the user. A distributed operating system provides the same functionality and interface as a monolithic operating system. That is, for both systems the goal is to make the computing and storage facilities as provided by the hardware available to the users of the system. How Distributed Systems Work. The most important functions of distributed computing are: Resource sharing – whether it's the hardware, software or data that can be shared. Openness – how open is the software designed to be developed and shared with each other.

Distributed computing allows different users or computers to share information. Distributed computing can allow an application on one machine to leverage processing power, memory, or storage on another machine. Here are a few of them: Solaris – Made for multiprocessor SUN workstations. OSF/1 – Created by the very Open Foundation Software Company and is Unix compatible. Micros – While allocating particular jobs to all nodes present in the system, the MICROS OS ensures a balanced data load.

2. Aim/benefits of the microproject:

The main goal of a distributed system is to make it easy for users to access remote resources, and to share them with other users in a controlled manner. Resources can be virtually anything, typical examples of resources are printers, storage facilities, data, files, web pages, and networks. With resource sharing facility, a user at one site may be able to use the resources available at another. Speedup the exchange of data with one another via electronic mail. Failure of one site in a distributed system doesn't affect the others, the remaining sites can potentially continue operating.

3.course outcome:

1)To get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.

2)Git for Distributed Software Development.

3)Distributed programming on the cloud. Carnegie Mellon University's Cloud Developer course.

4.Review:

A distributed operating system is an operating system that runs on a number of technologies whose function is to make available a useful set of services, generally to make the set of machines act more like a only machine.

The constant paging and the garbage collector eating up CPU cycles make the process slower. Eventually, when there is no more physical memory, and there is no more space in the swap file, the process won't be able to allocate more memory, and most operations will fail. You have a lot of different machines. They're running different processes. They only have message parsing via unreliable networks with variable delays, and the system may suffer from a host of partial failures, unreliable clocks, and process pauses. Distributed computing is really hard to reason about. Increment in the potential for hardware and software incompatibility across the organization is a disadvantage of distributed data processing.

The incompatibility in the potential of software and hardware creates a hurdle in distributing the data for further process.

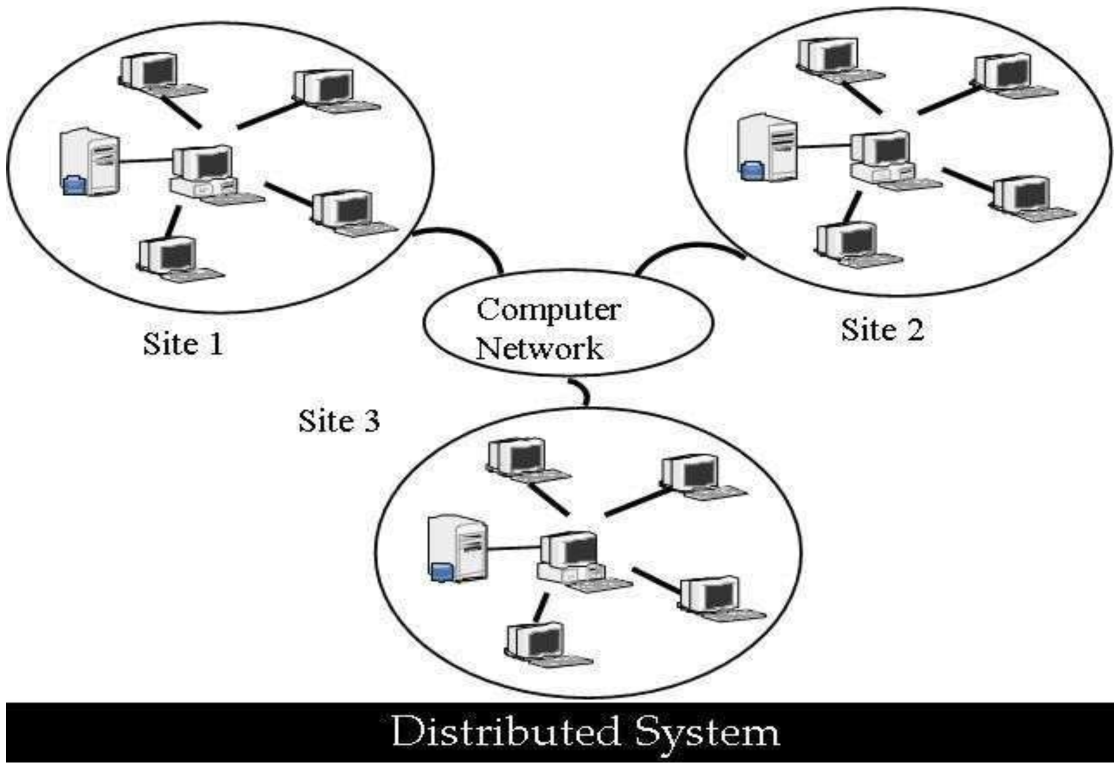
Distributed computing can be an important enabling technology for “big data”. Hadoop is a great, and strange as it might sound, simple, distributed computing example. Having an understanding of underlying technology can be helpful in making better use of your tools.

5. Actual Methodology followed:

A distributed operating system (DOS), is a recent advancement in the technological world of computers, furthermore, they are being accepted/utilised all over the world, as one of the main advantageous characteristics of these systems are that they provide great pace.

A DOS is a system which contains multiple components located on different machines, which coordinate and communicate actions in order to appear as a single coherent working system to the user.A distributed operating system (DOS), are systems which model where distributed applications are running on multiple computers, linked by communications.Processors in a DOS communicate with each other through various communication lines like high-speed buses.

A DOS involves a collection of autonomous computer systems, which are able to communicate with each other through LAN/WAN. This system will provide a virtual machine abstraction to its users and wide sharing of resources like computational capacity and input/output etc.This system as mentioned above, incorporates various autonomous interconnected computers that communicate with each other using a shared communication network, furthermore they are independent systems that possess their own memory unit and CPU.Another term which is used along side-distributed operating systems is a loosely coupled system. The processors in these systems may differ in size and function.The fundamental implementations of primitive distributed operating systems data back to the 1950's. Some of these concepts were not focused directly on DOS, and at the time many not have had realised their important impact. These pioneering efforts had laid down the fundamentals and inspired more research. However it was only when the acceleration of multi-processor/multi-core processor system ignited, that it led to the resurgence of DOS.One of the big advantages of working with DOS is that it is always possible that one user can access the files or software, which they require, and utilise them, however in reality these files are present on another system network (so think of it similar to remote working).Distributed systems can be considered to be more reliable than a central system because if the system has only one instance of a critical peripheral/component, like the CPU, network interface, disk, and so if that one instance fails, the system will go down completely.However when there are multiple instances in the system, like in a distributed operating system, then if a component fails, the system may be able to continue to function despite the failure. Distributed systems also allow software failures to be dealt with, rather than stopping the whole system.



Distributed computing models

Three basic distributions

To better illustrate this point, examine three system architectures; centralized, decentralized, and distributed. In this examination, consider three structural aspects: organization, connection, and control. Organization describes a system's physical arrangement characteristics. Connection covers the communication pathways among nodes. Control manages the operation of the earlier two considerations.

Organization:-

A centralized system has one level of structure, where all constituent elements directly depend upon a single control element. A decentralized system is hierarchical. The bottom level unites subsets of a system's entities. These entity subsets in turn combine at higher levels, ultimately culminating at a central master element. A distributed system is a collection of autonomous elements with no concept of levels.

Connection

Centralized systems connect constituents directly to a central master entity in a hub and spoke fashion. A decentralized system (aka network system) incorporates direct and indirect paths between constituent elements and the central entity. Typically this is configured as a hierarchy with only one shortest path between any two elements. Finally, the distributed operating system requires no pattern; direct and indirect connections are possible between any two elements. Consider the 1970s phenomena of “string art” or a spirograph drawing as a fully connected system, and the spider's web or the Interstate Highway System between U.S. cities as examples of a partially connected system.

Control

Centralized and decentralized systems have directed flows of connection to and from the central entity, while distributed systems communicate along arbitrary paths. This is the pivotal notion of the third consideration. Control involves allocating tasks and data to system elements balancing efficiency, responsiveness, and complexity.

Centralized and decentralized systems offer more control, potentially easing administration by limiting options. Distributed systems are more difficult to explicitly control, but scale better horizontally and offer fewer points of system-wide failure. The associations conform to the needs imposed by its design but not by organizational chaos

Overview

The kernel

At each locale (typically a node), the kernel provides a minimally complete set of node-level utilities necessary for operating a node's underlying hardware and resources. These mechanisms include allocation, management, and disposition of a node's resources, processes, communication, and input/output management support functions. Within the kernel, the communications sub-system is of foremost importance for a distributed OS.

In a distributed OS, the kernel often supports a minimal set of functions, including low-level address space management, thread management, and inter-process communication (IPC). A kernel of this design is referred to as a microkernel. Its modular nature enhances reliability and security, essential features for a distributed OS.

System management

System management components are software processes that define the node's policies. These components are the part of the OS outside the kernel. These components provide higher-level communication, process and resource management, reliability, performance and security. The components match the functions of a single-entity system, adding the transparency required in a distributed environment.

The distributed nature of the OS requires additional services to support a node's responsibilities to the global system. In addition, the system management components accept the "defensive" responsibilities of reliability, availability, and persistence. These responsibilities can conflict with each other. A consistent approach, balanced perspective, and a deep understanding of the overall system can assist in identifying diminishing returns. Separation of policy and mechanism mitigates such conflicts.

Working together as an operating system

The architecture and design of a distributed operating system must realize both individual node and global system goals. Architecture and design must be approached in a manner consistent with separating policy and mechanism. In doing so, a distributed operating system attempts to provide an efficient and reliable distributed computing framework allowing for an absolute minimal user awareness of the underlying command and control efforts.

The multi-level collaboration between a kernel and the system management components, and in turn between the distinct nodes in a distributed operating system is the functional challenge of the distributed operating system. This is the point in the system that must maintain a perfect harmony of purpose, and simultaneously maintain a complete disconnect of intent from implementation. This challenge is the distributed operating system's opportunity to produce the foundation and framework for a reliable, efficient, available, robust, extensible, and scalable system. However, this opportunity comes at a very high cost in complexity.

The price of complexity

In a distributed operating system, the exceptional degree of inherent complexity could easily render the entire system an anathema to any user. As such, the logical price of realizing a distributed operation system must be calculated in terms of overcoming vast amounts of complexity in many areas, and on many levels. This calculation includes the depth, breadth, and range of design investment and architectural planning required in achieving even the most modest implementation.

These design and development considerations are critical and unforgiving. For instance, a deep understanding of a distributed operating system's overall architectural and design detail is required at an exceptionally early point. An exhausting array of design considerations are inherent in the development of a distributed operating system. Each of these design considerations can potentially affect many of the others to a significant degree. This leads to a massive effort in balanced approach, in terms of the individual design considerations, and many of their permutations. As an aid in this effort, most rely on documented experience and research in distributed computing power.

6. Actual Resources Used:

Sr. No.	Name of resource / material	Specification	Quantity	Remarks
1	Computer	WINDOWS 7,2GB RAM, 160GB HDD	1	
2	Operating System	WINDOWS 7	1	
3	Textbook/manual	Ubuntu Studies 22447	1	
4	Internet	Youtube / Wikipedia	1	

7. Output of the Microproject:

Here are a few of them: Solaris – Made for multiprocessor SUN workstations. OSF/1 – Created by the very Open Foundation Software Company and is Unix compatible.
Micros – While allocating particular jobs to all nodes present in the system, the MICROS OS ensures a balanced data load. It enables the distribution of full systems on a couple of center processors, and it supports many realtime products and different users. Distributed operating systems can share their computing resources and I/O files while providing users with virtual machine abstraction.

One example of a distributed application is an e-commerce platform that distributes different functions of the application to different computers in its network. The servers or computers host different functions, such as the following: Accept payment from customers at checkout.

8. Skill developed / Learning out of this Micro-Project:

Resource Sharing: The significant component of this framework is that it permits clients to share resources in a secure and controlled way. …

Openness: …
Concurrency: …

Scalability: …

Fault Tolerance

Sharding. …

Distributed Databases. …

Paxos/Raft algorithms. …

Tools for Distributed Computation. …

Distributed File Systems. …

Remote Procedure Call (RPC) …

Distributed Messaging Systems. …

Distributed Ledgers.

9. Applications of this Micro-Project:

A distributed operating system (DOS) is an essential type of operating system. Distributed systems use many central processors to serve multiple real-time applications and users. As a result, data processing jobs are distributed between the processors. It connects multiple computers via a single communication channel.

Advantages:

1.Low cost

2.Improved performance and we can increasing processing power by adding node to connect more computers and decrease the power by removing computers.

3.Flexible.

4.Time saving and reliable.

Disadvantages:

1.It is difficult to provide adequate security in distributed systems because the nodes as well as the connections need to be secured.

2.Some messages and data can be lost in the network while moving from one node to another.