**concurrent computing in cloud**

Concurrency or concurrent computing refers to the form of computing in which multiple computing tasks occur simultaneously or at overlapping times. These tasks can be handled by individual computers, specific applications or across networks.

**Introducing parallelism for single-machine computation**

**Parallel Computing :**
It is the use of multiple processing elements simultaneously for solving any problem. Problems are broken down into instructions and are solved concurrently as each resource that has been applied to work is working at the same time.

**Advantages** of Parallel Computing over Serial Computing are as follows:

1. It saves time and money as many resources working together will reduce the time and cut potential costs.
2. It can be impractical to solve larger problems on Serial Computing.
3. It can take advantage of non-local resources when the local resources are finite.
4. Serial Computing ‘wastes’ the potential computing power, thus Parallel Computing makes better work of the hardware.

**Types of Parallelism:**

1. **Bit-level parallelism –**
It is the form of parallel computing which is based on the increasing processor’s size. It reduces the number of instructions that the system must execute in order to perform a task on large-sized data.
*Example:* Consider a scenario where an 8-bit processor must compute the sum of two 16-bit integers. It must first sum up the 8 lower-order bits, then add the 8 higher-order bits, thus requiring two instructions to perform the operation. A 16-bit processor can perform the operation with just one instruction.
2. **Instruction-level parallelism –**
A processor can only address less than one instruction for each clock cycle phase. These instructions can be re-ordered and grouped which are later on executed concurrently without affecting the result of the program. This is called instruction-level parallelism.
3. **Task Parallelism –**
Task parallelism employs the decomposition of a task into subtasks and then allocating each of the subtasks for execution. The processors perform the execution of sub-tasks concurrently.

       4. **Data-level parallelism (DLP)** **–**
Instructions from a single stream operate concurrently on several data – Limited by non-regular data manipulation patterns and by memory bandwidth

**Why parallel computing?**

* The whole real-world runs in dynamic nature i.e. many things happen at a certain time but at different places concurrently. This data is extensively huge to manage.
* Real-world data needs more dynamic simulation and modeling, and for achieving the same, parallel computing is the key.
* Parallel computing provides concurrency and saves time and money.
* Complex, large datasets, and their management can be organized only and only using parallel computing’s approach.
* Ensures the effective utilization of the resources. The hardware is guaranteed to be used effectively whereas in serial computation only some part of the hardware was used and the rest rendered idle.
* Also, it is impractical to implement real-time systems using serial computing.

**Applications of Parallel Computing:**

* Databases and Data mining.
* Real-time simulation of systems.
* Science and Engineering.
* Advanced graphics, augmented reality, and virtual reality.

**Limitations of Parallel Computing:**

* It addresses such as communication and synchronization between multiple sub-tasks and processes which is difficult to achieve.
* The algorithms must be managed in such a way that they can be handled in a parallel mechanism.
* The algorithms or programs must have low coupling and high cohesion. But it’s difficult to create such programs.
* More technically skilled and expert programmers can code a parallelism-based program well.

**Future of Parallel Computing:**The computational graph has undergone a great transition from serial computing to parallel computing. Tech giant such as Intel has already taken a step towards parallel computing by employing multicore processors. Parallel computation will revolutionize the way computers work in the future, for the better good. With all the world connecting to each other even more than before, Parallel Computing does a better role in helping us stay that way. With faster networks, distributed systems, and multi-processor computers, it becomes even more necessary.

Programming applications with threads

In a program with multiple threads, each thread runs its code independently of the other threads in the program. The concepts described here pertain to all programming languages. A function is threadsafe if you can start it simultaneously in multiple threads within the same process.



What is a thread?

A thread is a unit of execution within a process that shares the same memory space, code, data, and resources with other threads in the same process. A thread can run concurrently with other threads in the same process or in different processes, depending on the operating system and the hardware capabilities

Threading API

Threading is a process where multiple threads run at the same time to increase the efficiency of the processor. Using the Threading API, you can run a main thread and a worker thread simultaneously in your JavaScript application.

The Threading API uses kony namespace and the following API elements.

| **Function** | **Description** |
| --- | --- |
| [kony.runOnMainThread](https://docs.kony.com/konylibrary/visualizer/viz_api_dev_guide/content/kony_functions_threadingapis.htm#kony.runOnMainThread) | Helps you run the JavaScript code on the main thread. It is an asynchronous API. It posts a message to the main thread to invoke a function f with parameters arguments. |
| [kony.runOnWorkerThread](https://docs.kony.com/konylibrary/visualizer/viz_api_dev_guide/content/kony_functions_threadingapis.htm#kony.runOnWorkerThread) | Provides apps with multithreading capabilities. |

Create a Main thread or a UI thread using the[kony.runOnMainThread](https://docs.kony.com/konylibrary/visualizer/viz_api_dev_guide/content/kony_functions_threadingapis.htm#kony.runOnMainThread) function to run all the operations that involve interaction with the UI. The Worker thread can be created using the [kony.runOnWorkerThread](https://docs.kony.com/konylibrary/visualizer/viz_api_dev_guide/content/kony_functions_threadingapis.htm%22%20%5Cl%20%22kony.runOnWorkerThread) function to run all the background tasks in parallel with the main thread. For example, Garbage collection thread is a worker thread that runs in the background to clear all the unused data in an application.

Guidelines for using Threading API

To use Threading APIs in Quantum Visualizer, follow the guidelines:

* Threading APIs natively support multi-threading environment.
* An application can be composed of multiple concurrent threads.
* The UI or main thread is responsible for dispatching events to the user interface widgets and drawing the elements of the UI.
* Do not block the UI thread. Performing long operations, like network access or database queries on the UI thread, will block the user interface.
* Do not access the UI components from outside the UI thread.
* JavaScript Thread:
	+ Application logic written in JavaScript that does not require UI update executes in a different thread than the Main or UI thread.
	+ Operations that update the UI are posted on to the Main or UI thread.
* Use the API kony.runOnMainThread(function, args) to execute JavaScript bindings and JavaScript logic on UI Thread.

Techniques for parallel computation with threads

# What is Parallel Computing?



Parallel computing refers to the process of executing several processors an application or computation simultaneously. Generally, it is a kind of computing architecture where the large problems break into independent, smaller, usually similar parts that can be processed in one go. It is done by multiple CPUs communicating via shared memory, which combines results upon completion. It helps in performing large computations as it divides the large problem between more than one processor.

Parallel computing also helps in faster application processing and task resolution by increasing the available computation power of systems. The parallel computing principles are used by most supercomputers employ to operate. The operational scenarios that need massive processing power or computation, generally, parallel processing is commonly used there.

Typically, this infrastructure is housed where various processors are installed in a server rack; the application server distributes the computational requests into small chunks then the requests are processed simultaneously on each server. The earliest computer software is written for serial computation as they are able to execute a single instruction at one time, but parallel computing is different where it executes several processors an application or computation in one time.

There are many reasons to use parallel computing, such as save time and money, provide concurrency, solve larger problems, etc. Furthermore, parallel computing reduces complexity. In the real-life example of parallel computing, there are two queues to get a ticket of anything; if two cashiers are giving tickets to 2 persons simultaneously, it helps to save time as well as reduce complexity.

Backward Skip 10sPlay VideoForward Skip 10s

## Types of parallel computing

From the open-source and proprietary parallel computing vendors, there are generally three types of parallel computing available, which are discussed below:

1. **Bit-level parallelism:** The form of parallel computing in which every task is dependent on processor word size. In terms of performing a task on large-sized data, it reduces the number of instructions the processor must execute. There is a need to split the operation into series of instructions. For example, there is an 8-bit processor, and you want to do an operation on 16-bit numbers. First, it must operate the 8 lower-order bits and then the 8 higher-order bits. Therefore, two instructions are needed to execute the operation. The operation can be performed with one instruction by a 16-bit processor.
2. **Instruction-level parallelism:** In a single CPU clock cycle, the processor decides in instruction-level parallelism how many instructions are implemented at the same time. For each clock cycle phase, a processor in instruction-level parallelism can have the ability to address that is less than one instruction. The software approach in instruction-level parallelism functions on static parallelism, where the computer decides which instructions to execute simultaneously.
3. **Task Parallelism:** Task parallelism is the form of parallelism in which the tasks are decomposed into subtasks. Then, each subtask is allocated for execution. And, the execution of subtasks is performed concurrently by processors.

## Applications of Parallel Computing

There are various applications of Parallel Computing, which are as follows:

ADVERTISEMENT

ADVERTISEMENT

* One of the primary applications of parallel computing is Databases and Data mining.
* The real-time simulation of systems is another use of parallel computing.
* The technologies, such as Networked videos and Multimedia.
* Science and Engineering.
* Collaborative work environments.
* The concept of parallel computing is used by augmented reality, advanced graphics, and virtual reality.

## Advantages of Parallel computing

Parallel computing advantages are discussed below:

* In parallel computing, more resources are used to complete the task that led to decrease the time and cut possible costs. Also, cheap components are used to construct parallel clusters.
* Comparing with Serial Computing, parallel computing can solve larger problems in a short time.
* For simulating, modeling, and understanding complex, real-world phenomena, parallel computing is much appropriate while comparing with serial computing.
* When the local resources are finite, it can offer benefit you over non-local resources.
* There are multiple problems that are very large and may impractical or impossible to solve them on a single computer; the concept of parallel computing helps to remove these kinds of issues.
* One of the best advantages of parallel computing is that it allows you to do several things in a time by using multiple computing resources.
* Furthermore, parallel computing is suited for hardware as serial computing wastes the potential computing power.

## Disadvantages of Parallel Computing

There are many limitations of parallel computing, which are as follows:

* It addresses Parallel architecture that can be difficult to achieve.
* In the case of clusters, better cooling technologies are needed in parallel computing.
* It requires the managed algorithms, which could be handled in the parallel mechanism.
* The multi-core architectures consume high power consumption.
* The parallel computing system needs low coupling and high cohesion, which is difficult to create.
* The code for a parallelism-based program can be done by the most technically skilled and expert programmers.
* Although parallel computing helps you out to resolve computationally and the data-exhaustive issue with the help of using multiple processors, sometimes it affects the conjunction of the system and some of our control algorithms and does not provide good outcomes due to the parallel option.
* Due to synchronization, thread creation, data transfers, and more, the extra cost sometimes can be quite large; even it may be exceeding the gains because of parallelization.
* Moreover, for improving performance, the parallel computing system needs different code tweaking for different target architectures

## Fundamentals of Parallel Computer Architecture

Parallel computer architecture is classified on the basis of the level at which the hardware supports parallelism. There are different classes of parallel computer architectures, which are as follows:

### **Multi-core computing**

A computer processor integrated circuit containing two or more distinct processing cores is known as a multi-core processor, which has the capability of executing program instructions simultaneously. Cores may implement architectures like VLIW, superscalar, multithreading, or vector and are integrated on a single integrated circuit die or onto multiple dies in a single chip package. Multi-core architectures are classified as heterogeneous that consists of cores that are not identical, or they are categorized as homogeneous that consists of only identical cores.

### **Symmetric multiprocessing**

In Symmetric multiprocessing, a single operating system handles multiprocessor computer architecture having two or more homogeneous, independent processors that treat all processors equally. Each processor can work on any task without worrying about the data for that task is available in memory and may be connected with the help of using on-chip mesh networks. Also, all processor contains a private cache memory.

### **Distributed computing**

On different networked computers, the components of a distributed system are located. These networked computers coordinate their actions with the help of communicating through HTTP, RPC-like message queues, and connectors. The concurrency of components and independent failure of components are the characteristics of distributed systems. Typically, distributed programming is classified in the form of peer-to-peer, client-server, n-tier, or three-tier architectures. Sometimes, the terms parallel computing and distributed computing are used interchangeably as there is much overlap between both.

### **Massively parallel computing**

In this, several computers are used simultaneously to execute a set of instructions in parallel. Grid computing is another approach where numerous distributed computer system execute simultaneously and communicate with the help of the Internet to solve a specific problem.

Multithreading with Aneka

# Aneka in Cloud Computing

Aneka includes an extensible set of APIs associated with programming models like MapReduce.

These APIs support different cloud models like a private, public, hybrid Cloud.

Manjrasoft focuses on creating innovative software technologies to simplify the development and deployment of private or public cloud applications. Our product plays the role of an application platform as a service for multiple cloud computing.

* Multiple Structures:
* Aneka is a software platform for developing cloud computing applications.
* In Aneka, cloud applications are executed.
* Aneka is a pure PaaS solution for cloud computing.
* Aneka is a cloud middleware product.
* Manya can be deployed over a network of computers, a multicore server, a data center, a virtual cloud infrastructure, or a combination thereof.

### **Multiple containers can be classified into three major categories:**

* Textile services
* Foundation Services
* Application Services

**1. Textile Services:**

Fabric Services defines the lowest level of the software stack that represents multiple containers. They provide access to resource-provisioning subsystems and monitoring features implemented in many.

**2. Foundation Services:**

Fabric Services are the core services of Manya Cloud and define the infrastructure management features of the system. Foundation services are concerned with the logical management of a distributed system built on top of the infrastructure and provide ancillary services for delivering applications.

**3. Application Services:**

Application services manage the execution of applications and constitute a layer that varies according to the specific programming model used to develop distributed applications on top of Aneka.

### **There are mainly two major components in multiple technologies:**

**The SDK (Software Development Kit)** includes the Application Programming Interface (API) and tools needed for the rapid development of applications. The Anka API supports three popular cloud programming models: **Tasks, Threads** and **MapReduce**;

A runtime engine and platform for managing the deployment and execution of applications on a private or public cloud.

One of the notable features of Aneka Pass is to support the provision of private cloud resources from desktop, cluster to a virtual data center using **VMware, Citrix Zen Server**, and public cloud resources such as **Windows Azure, Amazon EC2**, and **GoGrid cloud service**.

Aneka's potential as a Platform as a Service has been successfully harnessed by its users and customers in three different areas, including **engineering, life sciences, education,** and **business intelligence.**

## Architecture of Aneka



Aneka is a platform and framework for developing distributed applications on the Cloud. It uses desktop PCs on-demand and CPU cycles in addition to a heterogeneous network of servers or datacenters. Aneka provides a rich set of APIs for developers to transparently exploit such resources and express the business logic of applications using preferred programming abstractions.

System administrators can leverage a collection of tools to monitor and control the deployed infrastructure. It can be a public cloud available to anyone via the Internet or a private cloud formed by nodes with restricted access.

A multiplex-based computing cloud is a collection of physical and virtualized resources connected via a network, either the Internet or a private intranet. Each resource hosts an instance of multiple containers that represent the runtime environment where distributed applications are executed. The container provides the basic management features of a single node and takes advantage of all the other functions of its hosting services.

Services are divided into clothing, foundation, and execution services. Foundation services identify the core system of Anka middleware, which provides a set of infrastructure features to enable Anka containers to perform specific and specific tasks. Fabric services interact directly with nodes through the Platform Abstraction Layer (PAL) and perform hardware profiling and dynamic resource provisioning. Execution services deal directly with scheduling and executing applications in the Cloud.

One of the key features of Aneka is its ability to provide a variety of ways to express distributed applications by offering different programming models; Execution services are mostly concerned with providing middleware with the implementation of these models. Additional services such as persistence and security are inverse to the whole stack of services hosted by the container.

At the application level, a set of different components and tools are provided to

* Simplify the development of applications (SDKs),
* Port existing applications to the Cloud, and
* Monitor and manage multiple clouds.

An Aneka-based cloud is formed by interconnected resources that are dynamically modified according to user needs using resource virtualization or additional CPU cycles for desktop machines. A common deployment of Aneka is presented on the side. If the deployment identifies a private cloud, all resources are in-house, for example, within the enterprise.

This deployment is enhanced by connecting publicly available on-demand resources or by interacting with several other public clouds that provide computing resources connected over the Internet.

Aneka thread vs. common threads

"Aneka thread" and "common threads" are not standard terms in the context of multithreading programming. However, I can provide an interpretation based on common understanding:

1. Common Threads:

 - Common threads refer to the typical threads used in multithreading programming. These threads are part of the standard multithreading model provided by programming languages and operating systems.

 - They are created using language-specific constructs or APIs provided by the operating system or libraries.

 - Common threads are used to execute concurrent tasks within a program, allowing for parallelism and better utilization of system resources.

2. Aneka Thread:

 - "Aneka" might refer to the Aneka platform, a cloud application platform for developing, deploying, and managing distributed applications. Aneka provides programming abstractions and tools for developing distributed applications that can leverage cloud resources.

 - In the context of Aneka, "Aneka thread" could potentially refer to threads or tasks managed by the Aneka platform within distributed applications.

 - These threads might have specific characteristics or behaviors tailored for distributed computing environments, such as communication across networked nodes and efficient utilization of cloud resources.

neka includes an extensible set of APIs associated with programming models like MapReduce.

These APIs support different cloud models like a private, public, hybrid Cloud.

Manjrasoft focuses on creating innovative software technologies to simplify the development and deployment of private or public cloud applications. Our product plays the role of an application platform as a service for multiple cloud computing.

* Multiple Structures:
* Aneka is a software platform for developing cloud computing applications.
* In Aneka, cloud applications are executed.
* Aneka is a pure PaaS solution for cloud computing.
* Aneka is a cloud middleware product.
* Manya can be deployed over a network of computers, a multicore server, a data center, a virtual cloud infrastructure, or a combination thereof.

### **Multiple containers can be classified into three major categories:**

* Textile services
* Foundation Services
* Application Services

**1. Textile Services:** Skip 10sPlay VideoForward Skip 10s

Fabric Services defines the lowest level of the software stack that represents multiple containers. They provide access to resource-provisioning subsystems and monitoring features implemented in many.

**2. Foundation Services:**

Fabric Services are the core services of Manya Cloud and define the infrastructure management features of the system. Foundation services are concerned with the logical management of a distributed system built on top of the infrastructure and provide ancillary services for delivering applications.

**3. Application Services:**

Application services manage the execution of applications and constitute a layer that varies according to the specific programming model used to develop distributed applications on top of Aneka.

### **There are mainly two major components in multiple technologies:**

**The SDK (Software Development Kit)** includes the Application Programming Interface (API) and tools needed for the rapid development of applications. The Anka API supports three popular cloud programming models: **Tasks, Threads** and **MapReduce**;

**And**

A runtime engine and platform for managing the deployment and execution of applications on a private or public cloud.

One of the notable features of Aneka Pass is to support the provision of private cloud resources from desktop, cluster to a virtual data center using **VMware, Citrix Zen Server**, and public cloud resources such as **Windows Azure, Amazon EC2**, and **GoGrid cloud service**.

Aneka's potential as a Platform as a Service has been successfully harnessed by its users and customers in three different areas, including **engineering, life sciences, education,** and **business intelligence.**