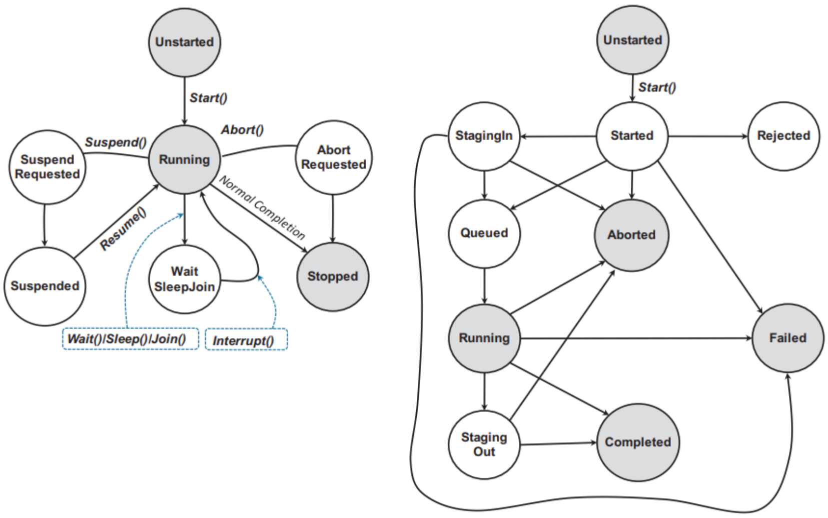
USN

Internal Assessment Test 2 – June 2022

Sub:	Cloud Computing & Its Applications				Sub Code:	18CS643	Branch:	CSE	
Date:	09-06-2022	Duration:	90 mins	Max Marks: 50	Sem / Sec:	VI/ A&B&C		OBE	

Answer any FIVE FULL Questions	MARKS	CO	RBT
1 **With the help of a neat diagram, differentiate the thread life cycle models of System.Thread and Aneka.Thread.**	[10]	2	L2

SOLUTION



a. System.Threading.Threadlife cycle.

b. Aneka.Threading.AnekaThreadlife cycle.

FIGURE 6.6

Thread life-cycle comparison.

Differences:

Thread Synchronization:

The .NET base class libraries provide advanced facilities to support thread synchronization with the help of monitors, and semaphores, read-write locks. Aneka provides minimal support for thread synchronization(thread joining) because of no shared memory. Cannot detect deadlocks.

Thread Priorities:

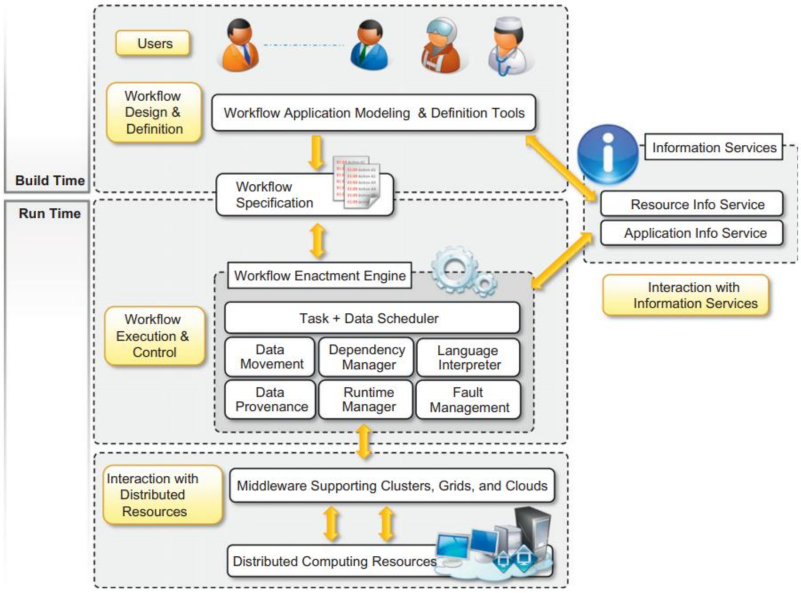
The System.Threading.Thread class supports thread priorities. (Highest, Above normal, Normal, Below Normal or Lowest). Aneka do not support thread priorities. For compatibility purpose, Aneka.Threading.Thread class exhibits Priority property whose values is always set to Normal.

Type Serialization:

Local threads execute within the same address space and share the memory.

It doesn’t need objects to be copied or transferred in to a different address space.

Aneka threads are distributed and execute on remote computing nodes. Hence the instances need to be transferred and reconstructed on the remote execution environment. This particular feature goes under the name of Type Serialization.

2	**By taking as a reference the Montage workflow, design a sketch of the control flow of a workflow strategy in Aneka.** **SOLUTION**  Users Workflow Design & Definition Workflow Application Modeling & Definition Tools Build Time Run Time Workflow Specification Information Services Resource Info Service Application Info Service Interaction with Information Services Workflow Enactment Engine Task + Data Scheduler Data Movement Dependency Manager Language Interpreter Data Provenance Runtime Manager Fault Management Workflow Execution & Control Interaction with Distributed Resources Middleware Supporting Clusters, Grids, and Clouds Distributed Computing Resources **FIGURE 7.7** Abstract model of a workflow system. Business-oriented computing workflows are defined as compositions of services, and there are specific languages and standards for the definition of workflows, such as Business Process Execution Language (BPEL) [65]. In the case of scientific computing there is no common ground for defining workflows, but several solutions and workflow languages coexist [66]. Despite such differences, it is possible to identify an abstract reference model for a workflow management system [67], as depicted in Figure 7.7. Design tools allow users to visually compose a workflow application. This specification is normally stored in the form of an XML document based on a specific workflow language and constitutes the input of the workflow engine, which controls the execution of the workflow by leveraging a distributed infrastructure. In most cases, the workflow engine is a clientside component that might interact directly with resources or with one or several middleware components for executing the workflow. Some frameworks can natively support the execution of workflow applications by providing a scheduler capable of directly processing the workflow specification.	[10]	2	L3
3	**Explain Task computing. Describe the parameter sweep application with an example.** **SOLUTION:** The Task Programming Model provides a very intuitive abstraction for quickly developing distributed applications on top of Aneka. It provides a minimum set of APIs that are mostly centered on the Aneka.Tasks.ITask interface. This interface, together with the services supporting the execution of tasks in the middleware, constitutes the core feature of the model. These are a specific class of embarrassingly parallel applications whose tasks are identical in nature but differ only by specific parameters. These are identified by template task and set of parameters. Nimrod/G is designed to support the execution of parameter sweep applications. A template task is a composition of operations	[3+7]	2	L2

concerning the execution of applications with appropriate parameters and set of file system operations for moving data.

Nimrod/G Task Template Definition

```
parameter x float range from 1 to 10 step 1;

parameter y float range from -4 to 5 step 1;

task main

        node: execute /bin/echo x: ${x} Y: $ {Y} > output

        copy node: output output. `expr ${y}\*+ ${x}

endtask
```

Aneka Parameter Sweep file

```
<psm>

        <name> Aneka Blast </name>

        <workspace>C:\Projects </workspace>

        <parameters>

        <single name = “p” type = “string” comment = “Name of the program”
value =  ’blast’>

        <range name = “s” type = “string” comment = “ The sequence file” from =
‘0’ to = ‘2’ interval = ‘1’>

        </parameters>
```

4	**Explain Domain composition Technique for parallel computations. Write Aneka code to create a Matrix Product class.**	[5+5]	2	L2

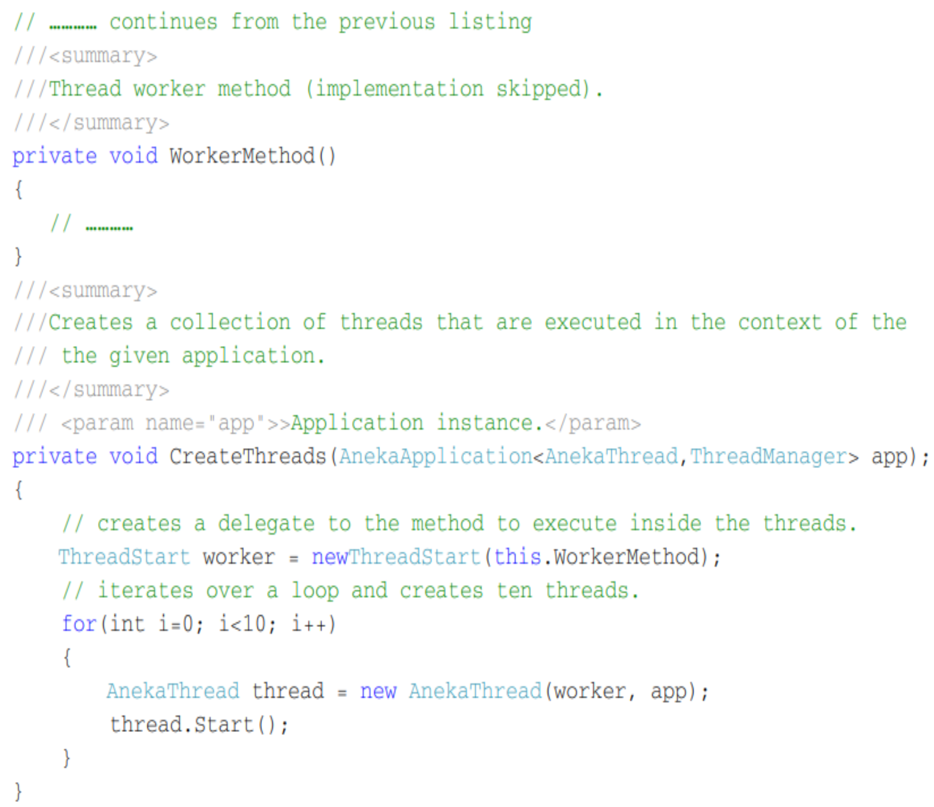
SOLUTION

To port to Aneka threads the multithreaded matrix multiplication, we need to apply the considerations made in the previous section. Hence, we start reviewing the code by first making the proper changes to the ScalarProduct class. Listing 6.6 shows the modified version of ScalarProduct. The class has been tagged with the Serializable attribute and extended with the methods required to implement custom serialization. Supporting custom serialization implies the following:

• Including the System.Runtime.Serialization namespace.

• Implementing the ISerializable interface. This interface has only one method that is void GetObjectData(SerializationInfo, StreamingContext), and it is called when the runtime needs to serialize the instance.

• Providing a constructor with the following signature: ScalarProduct(SerializationInfo, StreamingContext). This constructor is invoked when the instance is deserialized



```
// ……… continues from the previous listing
///<summary>
///Thread worker method (implementation skipped).
///</summary>
private void WorkerMethod()
{
   // ………
}
///<summary>
///Creates a collection of threads that are executed in the context of the
/// the given application.
///</summary>
/// <param name="app">>Application instance.</param>
private void CreateThreads(AnekaApplication<AnekaThread,ThreadManager> app);
{
    // creates a delegate to the method to execute inside the threads.
    ThreadStart worker = newThreadStart(this.WorkerMethod);
    // iterates over a loop and creates ten threads.
    for(int i=0; i<10; i++)
    {
        AnekaThread thread = new AnekaThread(worker, app);
        thread.Start();
    }
}
```

LISTING 6.5

Thread Creation and Execution.

5	**Among the Heartbeat, Fabric, and Foundation services provided by Aneka Infrastructure, which services are the most important. Justify the answer.**	[5+5]	2	L3

SOLUTION:

Fabric services are the fundamental services of Aneka Cloud, and defines the basic infrastructure management features of the system. Fabric services provide access to the resource provisioning, monitoring facilities. Profiling and Monitoring: These services are mostly exposed to the following:

1. Heartbeat
2. Monitoring
3. Reporting

The first makes the information available, which is collected from PAL. Other two implement a generic infrastructure for monitoring the activity of any service in Aneka.

Heartbeat Services periodically collects the dynamic performance about the node and publishes the information. It also collects the information about installed software or any other useful information. Node resolver is responsible for collecting all the information.The reporting services collects the report information and makes it available to other services.On each node, monitoring services acts as a gateway to the reporting service.

Resource Management:

Fundamental feature of Aneka cloud.It comprises several tasks: resource membership, resource reservation, resource provisioning.Aneka provides a collection of services that are incharge of managing the resources. They are:

○ Index services (Membership catalogue).

- ○ Reservation service.
- ○ Resource provisioning service

Storage Management: Any infrastructure supporting the execution of distributed applications need to provide facilities for file/data transfer management and persistent storage. Aneka offers two different facilities of storage:

Centralized file storage: Mostly used for the execution of Compute-Intensive applications Requires powerful processors and no high demands for storage

Distributed file system: Suitable for Data-Intensive applications. Include large data files, need to be scalable.

6 **Write the pseudocode for the following:** [4+4+2] 2 L3

a. **Create an ITask for finding the Matrix Multiplication Task.**
b. **Create an AnekaTask for the Matrix Multiplication Task.**
c. **If the tasks are finished, then application must be stopped.**

SOLUTION

a. ITask and Aneka Task

```
Namespace Aneka.Tasks {

        Public interface ITask {

                Public void execute();

        }

}
```

b. conf.SingleSubmission = true;

```
AnekaApplication <AnekaTask, TaskManager> app =

new AnekaApplication<Task, TaskManager>(conf);

for(int i =0; i<400; i++){

        MatrixTask Matrix = new MatrixTask();

        Matrix.X=i;

        AnekaTask task = new AnekaTask(Matrix);

        app.AddWorkUnit(Matrix);

}
```

	app.SubmitExecution();
	c. If (this.taskCount == 0) {
	this.app.StopExecution();
	}			

CO PO Mapping

Course Outcomes		Modules covered	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	
CO1	Explain cloud computing, virtualization and classification of services of cloud computing	1,2	2	2	2	2	0	3	3	0	0	0	0	0	3	0	2	2	
CO2	Illustrate architecture and programming examples in cloud	**2,3,4**	2	2	0	2	2	3	3	0	0	0	0	0	3	2	0	2	
CO3	Describe the platforms for development of cloud applications with examples	**4,5**	2	3	3	3	2	3	3	0	0	0	0	0	3	2	0	2	

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Develop applications using different stacks of web and programming technologies				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems				

PSO3	Apply software engineering methods to design, develop, test and manage software systems.
PSO4	Develop intelligent applications for business and industry