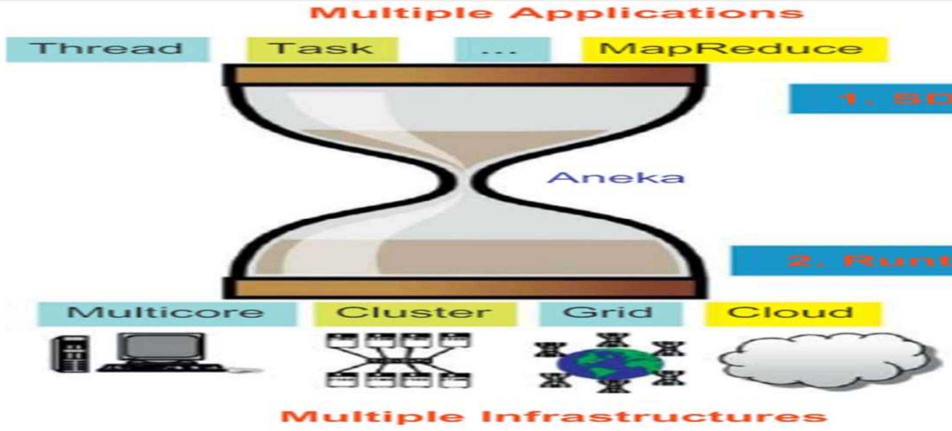
Aneka: Cloud Application Platform

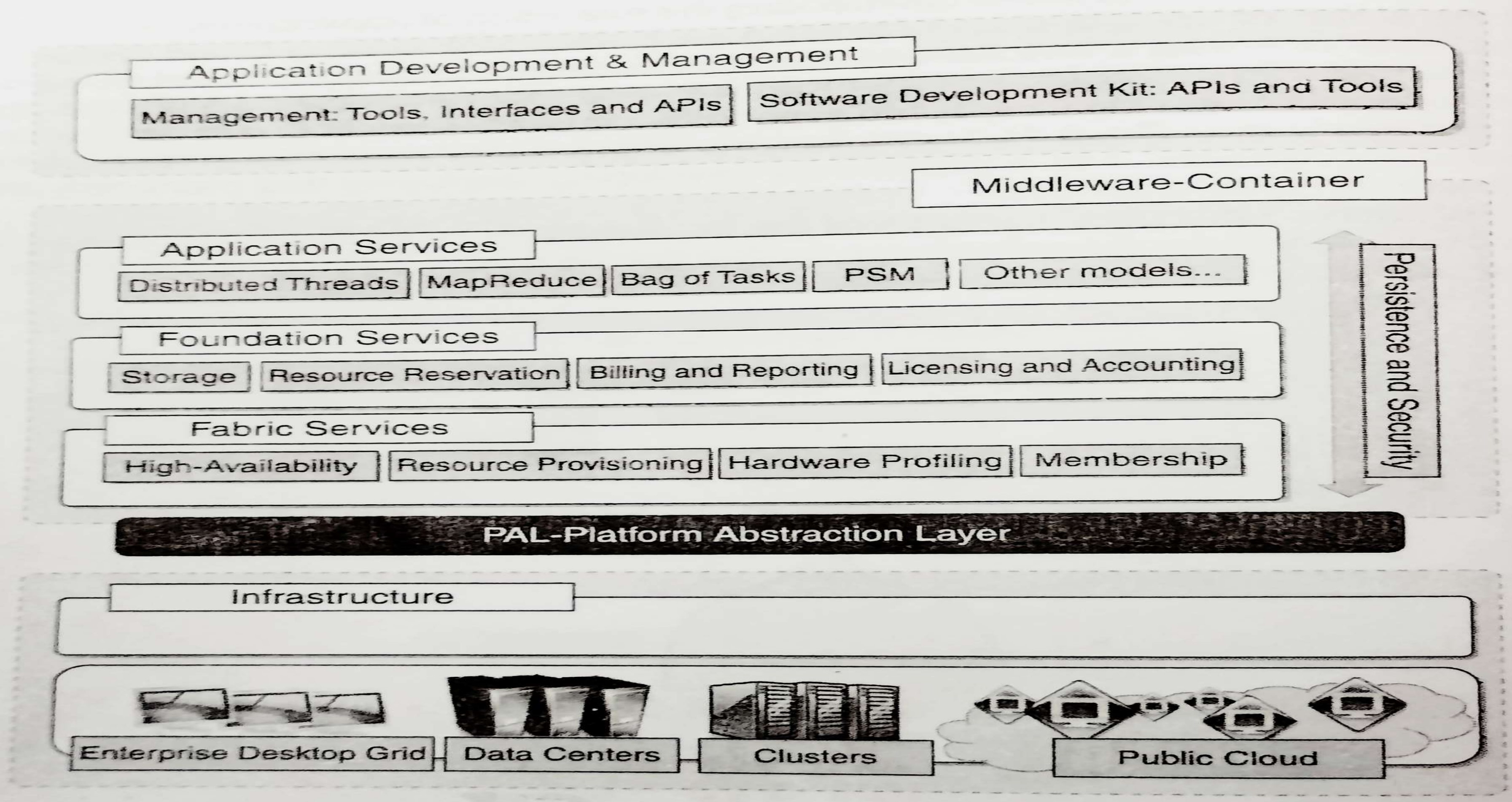
PaaS

Manjrasoft Aneka

- Aneka is **Manjrasoft solution** for **developing, deploying and managing Cloud applications**.
- Aneka is **pure PaaS solution** for cloud computing that can be **deployed on a heterogeneous set of resources** (network of computers, a multi-core server, data centers, virtual cloud infrastructure etc.)
- It **provides middleware for managing and scaling distributed** applications and an **extensible set of APIs** for their development.



Aneka Framework Overview



Aneka Framework Overview.

Framework Overview (Contd)

- The **physical and virtual resource**s representing bare metal of the Cloud are **managed by Aneka Container**.
- Aneka Container is **installed on each node** and constitute the basic building block of the infrastructure.
- Container features three different classes of services:

a) **Fabric Services -** Fabric services **directly interact with the node through the Platform Abstraction Layer (PAL)** and perform **hardware profiling & monitoring & resource management.**

b) **Foundation Services -** Core system of the Aneka middleware, **providing a set of basic features to enable Aneka containers to perform specialized and specific sets of tasks.**

Storage management for applications, Accounting, billing & resource pricing, Resource reservation

c) **Execution Services - E**xecution services directly deal with the scheduling and execution of applications in the Cloud.

Services are made available to developers and administrators by means of ***application management and development layer*** *which* ***includes interfaces & API for developing Cloud Applications, the management tools and interfaces for controlling the Aneka Cloud.***

Framework Overview (Contd)

- Aneka implements **Service Oriented Architecture & services** are the **fundamental** components of Aneka Cloud.
- Services also constitute the **extension and customization point of Aneka Clouds:** the infrastructure allows for integration of new services or replacement of existing ones with different implementation.
- The framework includes **basic services for infrastructure & node management, application execution, accounting** & system monitoring which can be extended & new features can also be added.
- Aneka clouds **support different programming & execution models for applications.**

Framework Contd. (Features)

a) **Elasticity & Scaling**: Aneka supports **up-sizing & down-sizing** of infrastructure available for applications.

b) **Runtime Management**: Runtime machinery is responsible for **keeping the infrastructure up & running, and serves as a hosting environment for services.**

Represented by a container & collection of services **managing service membership & lookup, infrastructure maintenance & profiling**

Framework Contd. (Features)

- **Resource Management:** Resources are **added & removed dynamically**, **reservation** of nodes etc.
- **Application Management**: Services to manage the applications which include **scheduling, execution, monitoring & storage management**.
- **User Management**: It is possible to **define users, groups and permissions.**
- **QoS/SLA Management & Billing**: Application **execution is metered & billed.** The services take into account usage of resources by each application & bill the owing user accordingly.

Framework (Contd.) SDK & Management Kit

- **Services are available to specific interfaces & APIs**, on top of **which software development kit and management kit are built.**
- **SDK** mainly relates to **application development and modeling**: it provides developers with API to **develop applications with existing programming models and an object model** for creation of new models.
- **Management kit** is focused on **interacting with runtime services for managing the infrastructure, users and applications**.
- **Management kit** gives a **complete view of Aneka Cloud** & allow monitoring its status **whereas SDK is focused on single applicatio**n and provides means to control its execution from a single user.

Anatomy of the Aneka Container

- Aneka **Container** constitutes the **building block of Aneka Clouds & represents the runtime machinery available to services & applications.**

- It is unit of **deployment in Aneka Clouds** and it is a **lightweight** software layer designed to **host services, interact with underlying operating system & hardware, provide communication channels for interaction with other nodes in Aneka Cloud**.

Services installed in Aneka Container

These can be classified in three major categories:

a) Fabric services

b) Foundation services

c) Application services

*The services stack resides on **top of Platform Abstraction Layer (PAL) representing the interface towards underlying operating system & hardware**.*

*The PAL layer provides a **uniform view of software & hardware environment in which container is running**.*

Platform Abstraction Layer

- **Core infrastructure is based on .NET technology.**
- The PAL address the heterogeneity of underlying operating system & provides the container with a **uniform interface for accessing the relevant hardware and OS,** allowing the rest of container to run unmodified on any supported platform. It provides the following features:

a) Uniform & platform independent **implementation interface for accessing the host platform.**

b) Uniform access to **extended & additional properties of hosting platform.**

c) Uniform & platform independent access to **remote nodes.**

d) Uniform & platform independent **management interfaces**.

Platform Abstraction Layer

- The collectible data that is exposed by PAL includes:

  a) No of cores, frequency & CPU usage.

  b) Memory size and usage.

  c) Aggregate available disk space.

  d) Network addresses and devices attached to the node etc.

Fabric Services

- These define the **lowest level of software stack representing the Aneka Container.**
- They provide access to **resource provisioning subsystem and to the monitoring facilities implemented in Aneka.**
- **Resource provisioning servic**e are in charge of **dynamically providing new nodes on demand by relying on virtualization technologies**
- **Monitoring** services allows for **hardware profiling & implement a basic monitoring infrastructure** that can be used by all services installed in the container.

Fabric Services (Contd.): Profiling & Monitoring

- Profiling & monitoring services are mostly exposed through: ***Heartbeat, Monitoring & Reporting Services.***
- The Heartbeat service makes available the information that is collected through PAL
- Monitoring & Reporting service implement a generic infrastructure for monitoring the activity of any service in the Aneka Cloud.

Fabric Services (Contd.): Profiling & Monitoring

- Heartbeat **service periodically collects the dynamic performance information** about the node & **publish** the information to **membership** service in Aneka Cloud.
- The data is collected by **index node of the Cloud**, which makes them available for services such as **reservation & scheduling in** order to optimize the usage of a heterogeneous infrastructure.
- The basic information **about memory, disk space, CPU and OS** is collected along with **additional information.**
- Information published by Heartbeat service is concerned with the **properties of the node (text – valued properties)**
- A specific component called **Node Resolver is in charge of collecting the data** & making them **available to the Heartbeat service.**

Fabric Services (Contd.): Profiling & Monitoring

- The generic monitoring infrastructure allowing **any custom service to report the activity** is composed by *Reporting & Monitoring services.*

- The **Reporting Service** manages the **store for monitored data & make them accessible to other services or external applications for analysis.**

- On each node, an instance of **Monitoring service acts a gateway to Reporting Service and forwards all monitored data that has been collected on the node to it.**

Fabric Services (Contd.): Resource Management

It comprises of several tasks & services:

a) Resource membership: Index Service (or Membership Catalogue)

b) Resource provisioning: Resource Provisioning Service

Resource Management: Membership Catalogue

- It keeps track of **the basic node information for all nodes that are connected or disconnected.**
- It implements the **basic services of a directory service** allowing **the search for services by using attributes such as names, & nodes.**
- During the **container startup, each instance publish the information to Membership Catalogue** & update it constantly during its lifetime.
- Services **and external applications can query the membership catalogue** in order to **discover the available services** and interact with them.
- To **speed up & enhance the performance** of queries, membership catalogue is organized as a **distributed database**.
- It is **also** the collector of **dynamic performance data of each node, which is sent to the local monitoring service** to be persisted for long term.

Resource Management: Dynamic Resource Provisioning

- Allows **integration and management of virtual resources leased from IaaS** providers into Aneka Cloud.
- The service **changes the structure of the Aneka Cloud** by allowing it to scale up and down, handling node failures, ensuring the QoS for applications, maintaining the constant performance & throughput of the Cloud.
- Aneka defines a **flexible infrastructure** for resource provisioning where it is **possible to change the logic that triggers provisioning, support several back-ends & change the runtime strategy with which a specific backend is selected for provisioning.**
- Mainly concentrated in Resource Provisioning Service, including all information needed for provisioning virtual instances.
- Implementation is based on concept of **resource pools** which abstract interaction with a specific IaaS provider **providing a common interface.**
- **Support QoS requirements**

Foundation Services

- Define the **basic infrastructure management features** of the system.

- Related to logical management of the distributed system built on top of the infrastructure & **provide supporting services for executing of distributed applications.**

- **Provide a set of basic features to enable Aneka containers to perform specialized and specific sets of tasks.**

- The services cover:

  a) Storage management for applications

  b) Accounting, billing & resource pricing

  c) Resource reservation

Storage Management

- Applications operate on data, which are mostly persisted & moved in the format of files.
- Aneka provides 2 different facilities for storage management:

a) **Centralized File Storage:** Used for compute-intensive applications.

Require powerful processors and do not have high demands in terms of storage, and in many cases are used to store small files easily transferred from one node to another. In the scenario, centralized storage node can constitute an appropriate solution

b) **Distributed File System:** Suitable for data-intensive applications.

Characterized by large data files (GB/TB) & processing power doesn't constitute performance bottleneck. In this case, distributed file system harnessing the storage space of all the nodes belonging to the Cloud might be a more scalable solution

Centralized Storage

- It is implemented through ***Storage Service.***
- Services constitute the data staging facilities of Aneka.
- It provides **distributed applications with basic file transfer facility.**
- The option **that is currently installed is normal FTP.**
- In order to support different protocols, **system introduces the concept of file channel that identifies a pair of components: a file channel controller and file channel handler.**
- **File channel controller constitute the server component of the channel, where files are stored and made available**.
- **File channel handler represents the client component which is used by user applications** or other system components to **upload, download or browse files.**

Distributed Storage

- Aneka provides a simple **Distributed File System (DFS) which rely on file system services of Windows Operating System.**

Accounting Management, Billing & Resource Pricing

Accounting Management

- The information collected for accounting is **related to usage of infrastructure and execution of applications.**
- **A complete history of application execution, storage as well other resource utilization parameter are captured by the accounting services.** The information constitutes the **foundation on top which users are charged in Aneka.**

Accounting Management, Billing & Resource Pricing

Billing

Aneka billing service provide **detailed information about resource usage of each user with the associated costs.**

Each resource can be priced differently according to the different set of services that are available on the corresponding Aneka container or the installed software in the node.

Accounting Management, Billing & Resource Pricing

The accounting capabilities are concentrated with **the Accounting Service and Reporting Service.**

Accounting service keeps track of the information that is related to application execution, such as **distribution of jobs among the available resources, timings of each job and the associated cost.**

The **reporting** service makes available the **information collected from the monitoring services for accounting purpose**: storage utilization & CPU performance.

Resource Reservation

- It allows for **reserving the resources for exclusive use by specific applications.**
- It is built on two different kinds of services: **Resource Reservation and Allocation Service.**
- Resource Reservation keeps track of **all reserved time slots in Aneka Cloud and provides a unified view of the system.**
- **Allocation service** is installed on each node featuring execution services and **manages the database of information regarding allocated slots on local node.**

Resource Reservation

- Applications that need to complete within a given **deadline can make the reservation request for a specific no of nodes in a given time** frame.
- If possible to serve the request, **Reservation Service will return a reservation identifier** as a part of the resource booking.
- On each reserved node, **execution services will check with the Allocation Service that each job has the valid permissions to occupy** the **execution timeline** by verifying the reservation identifier.

Resource Reservation

- Framework support the following implementation:

a) **Basic reservation:** Features **basic capability** of reserving execution slots on nodes & **implements alternate offers protocols**

b) **Libra reservation:** Variation of basic reservation featuring ability to **price nodes differently according to their hardware capability.**

c) **Relay reservation:** Allow **resource broker to reserve node in Aneka Cloud** & control the logic with which these nodes are reserved.

Useful in integration scenario **where Aneka operates in an inter-cloud environment.**

Application Services

- They manage the **execution of applications** and constitute a layer that **differentiates according to specific programming model** used for developing distributed applications on top of Aneka.

- Two major types of activities that are common across all the supported model: **scheduling and execution.**

Application Services: Scheduling

- Common tasks that are performed by the scheduling component are following:

a) **Job-to-node** mapping

b) **Rescheduling** of **failed** jobs

c) **Job status monitoring**

d) **Application status monitoring**

Application Services: Execution

- Execution services **control the execution of single jobs** that compose the **applications**.
- They are in charge of **setting the runtime environment hosting the execution of jobs.**
- Some common operations that apply across all range of supported models:

a**) Unpacking the jobs** received from **scheduler**.

b) **Retrieval of input files** required for job execution.

c) **Sandboxed execution** of **jobs**

d) **Submission of output files** at the end of execution.

e) **Execution failure management**

f) **Performance monitoring**

g) **Packing jobs and sending them back to the scheduler**.

Supported programming models

- **Task model:** Application is modelled as a **collection of tasks that are independent from each other** and whose **execution can be sequenced in any order.**
- **Thread model:** It provides an **extension to classical multi-threaded programming** to **distributed infrastructure** and use abstraction of **Thread to wrap a method that is executed remotely.**
- **MapReduce model: Implementation of MapReduce** as proposed **by Google on top of Aneka.**
- **Parameter Sweep Model: a) Specialization of task model**

  b) Applications can be described by a **template**

  task **whose instances are created by generating different combinations of parameters.**

References

- **MASTERING CLOUD COMPUTING, Rajkumar Buyya , Christian Vecchiola, S. Thamarai Selvi, McGraw Hill Education**
- **http://www.manjrasoft.com/aneka_architecture.html**