

User Defined Runtime Environments using UNICORE

UNICORE Summit 2011, Toruń, PL

Agenda

Introduction

Design

- Fitting the Model

- VMM Abstraction

- VM Images

- XaaS

Related Work

Summary and Future Work

Current situation



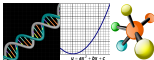
User

accesses



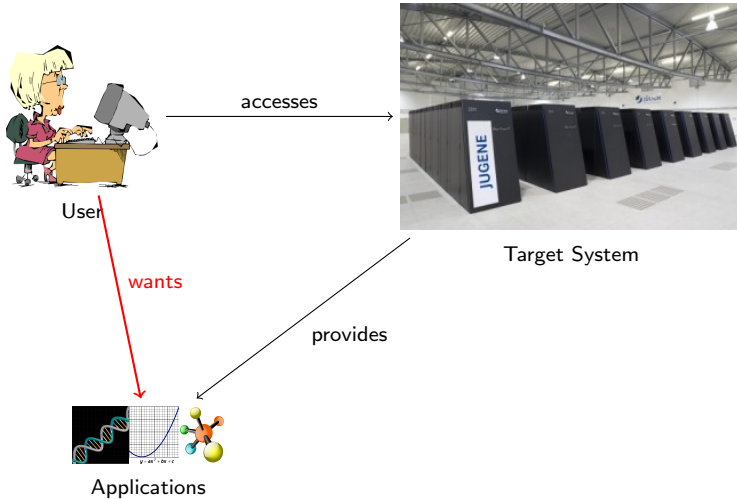
Target System

provides



Applications

Current situation

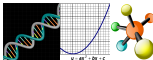


Current situation



User

wants



Applications

Motivation

User

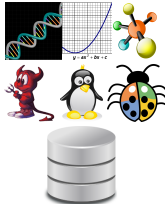


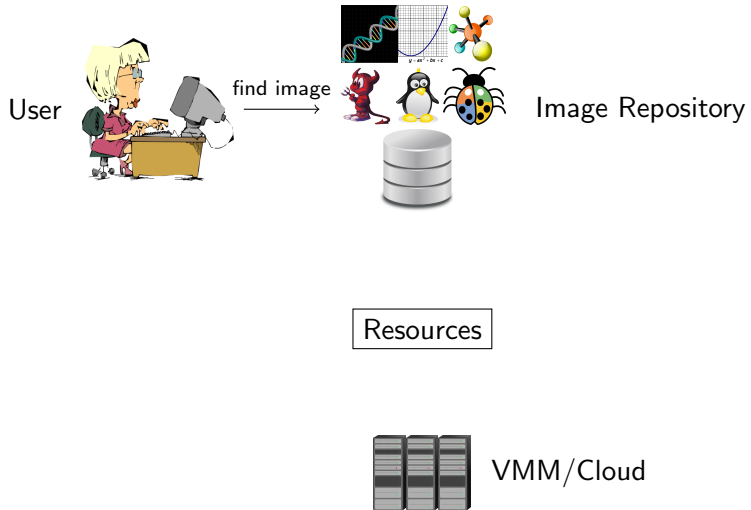
Image Repository

Resources

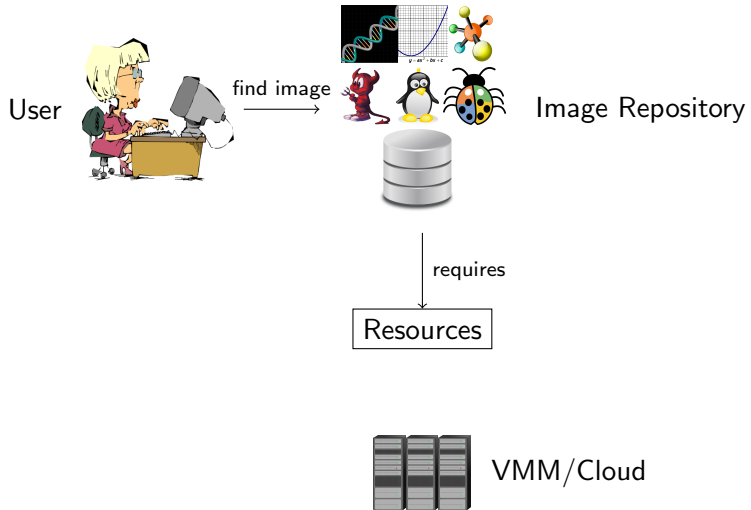


VMM/Cloud

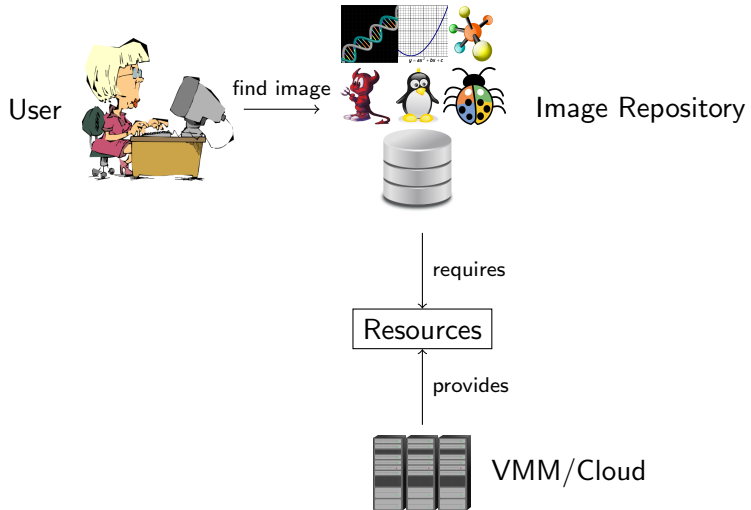
Motivation



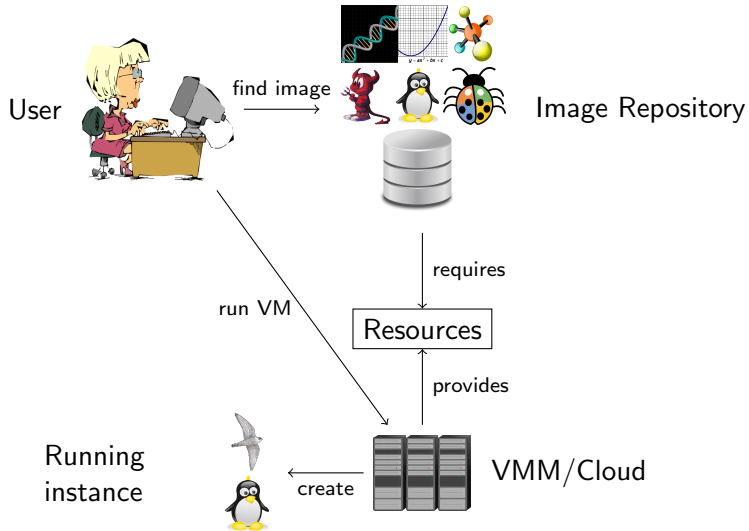
Motivation



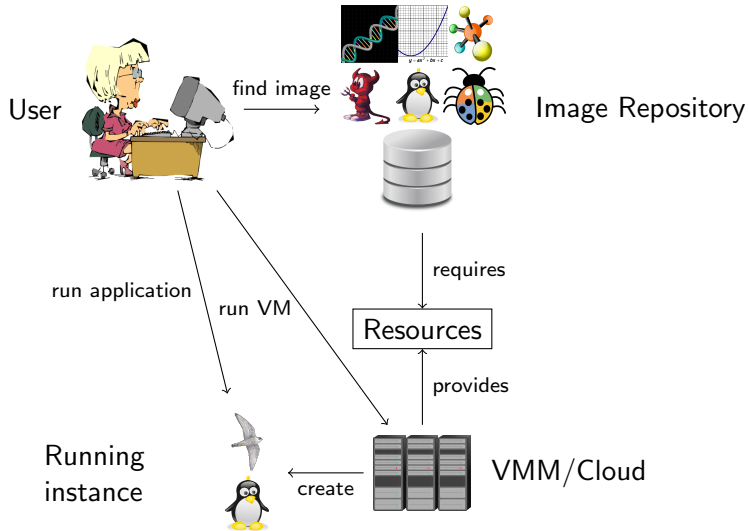
Motivation



Motivation



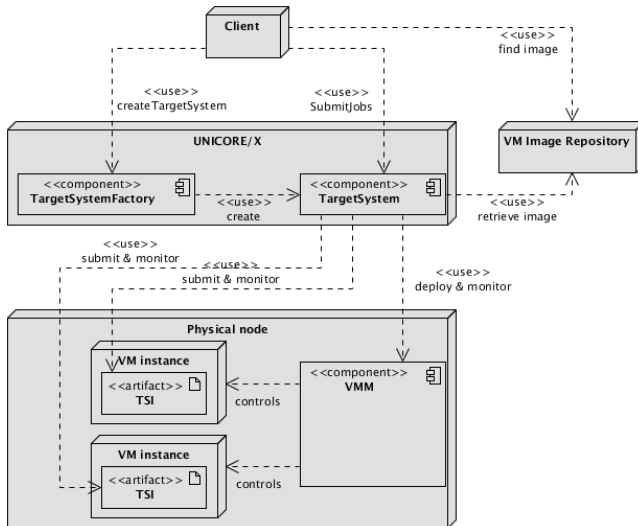
Motivation



Use Cases

- Specialized software setup
 - e. g. conflicting with other software configurations
- Specific runtime environments
 - Compiler suite
 - system libraries
 - Kernel
 - OS distribution
 - Applications in general

Fitting the Model



Required Changes

- TargetSystemFactory
 - Create TargetSystem according to requirements and given VM image
- TargetSystem/XNJS
 - Use a somewhat dynamic configuration taken from the image repository's metadata
 - Dynamically connect to the TSI inside the running instance
 - Keep track of running instance's health (Expose status)
- Client
 - Query image repository
 - Monitor state of VM instance
 - Provide parameters for instatiating VM images

Abstraction of the VMM

- Numerous virtualization solutions available
 - Xen, QEMU, KVM, VirtualBox, VMware, ...
- libVirt can connect to many of them
- additionally, libVirt does remote management of VMMs
- virtual networks
- storage



VM Image Repositories

- Need to store images plus metadata
 - Operating System
 - Applications provided
 - Requirements
- Will use UNICORE MMF for this purpose
 - Associate metadata with each image
 - Portions of IDB
 - Possibly TSI configuration
- The images themselves will contain the TSI

Image Metadata

- Explicitly set by image creator
- Read by user or orchestration service
- Contents
 - IDB configuration
 - TSI setup, e. g. port numbers

Resource Requirements

VM Image



VMM/TSF

Resource Requirements

VM Image



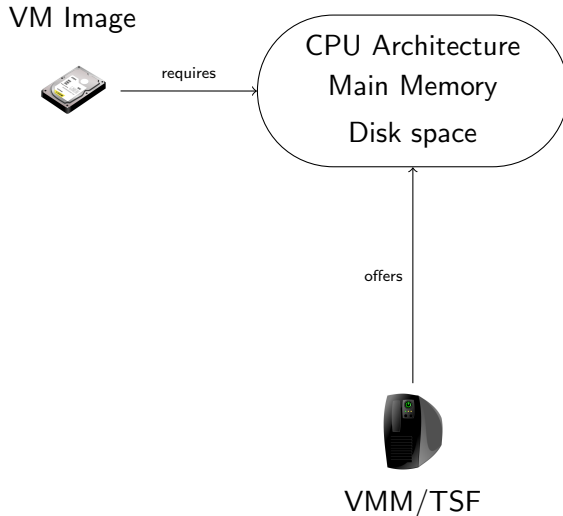
requires

CPU Architecture
Main Memory
Disk space

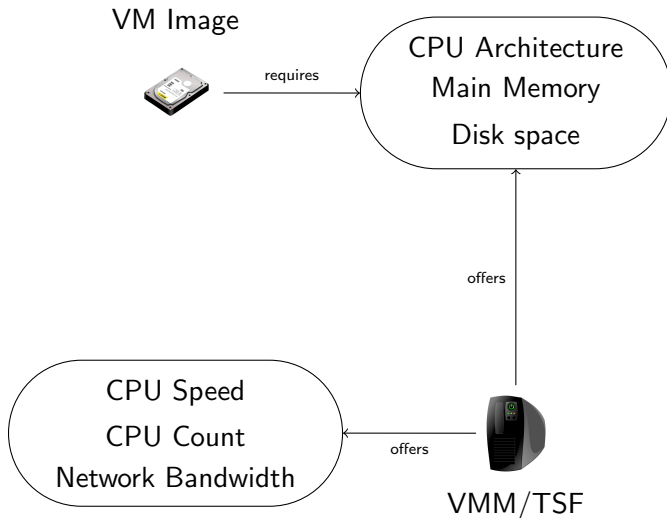


VMM/TSF

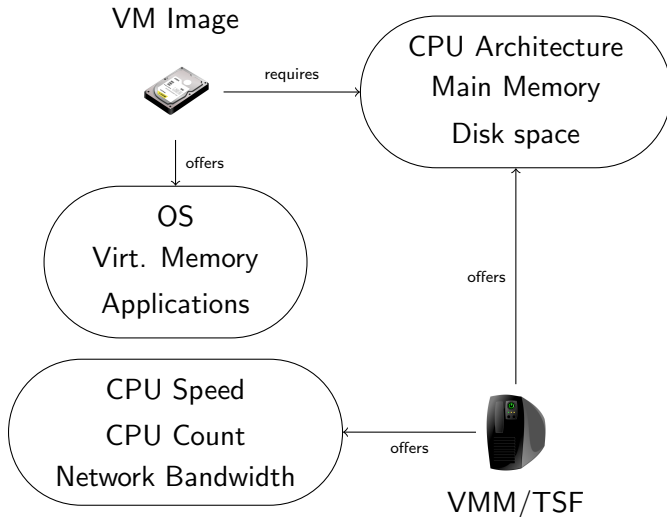
Resource Requirements



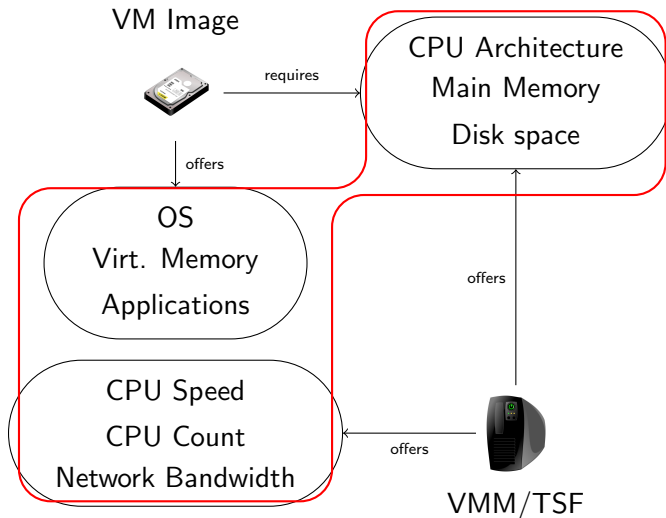
Resource Requirements



Resource Requirements



Resource Requirements



IaaS - PaaS - SaaS

IaaS



- Infrastructure assumed to be available

IaaS - PaaS - SaaS

PaaS



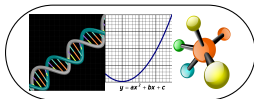
IaaS



- We'll be able to cover "Platform as a service" with our implementation
- Infrastructure assumed to be available

IaaS - PaaS - SaaS

SaaS



PaaS



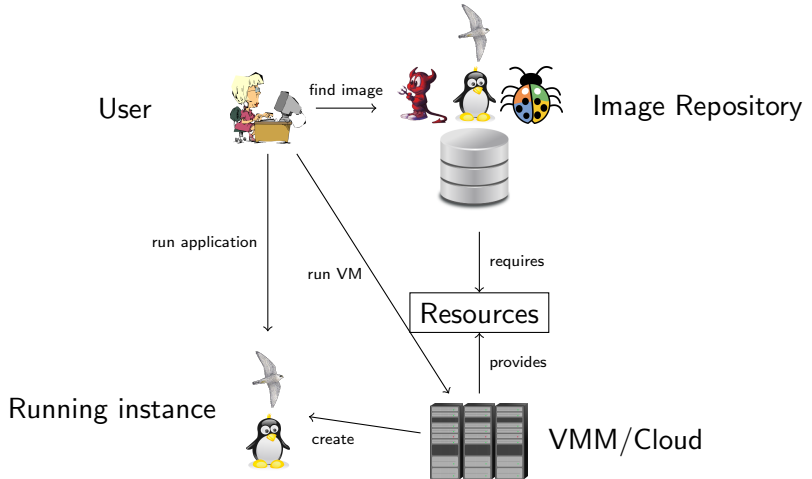
IaaS



- It doesn't take much to add "Software as a service" on top
- We'll be able to cover "Platform as a service" with our implementation
- Infrastructure assumed to be available

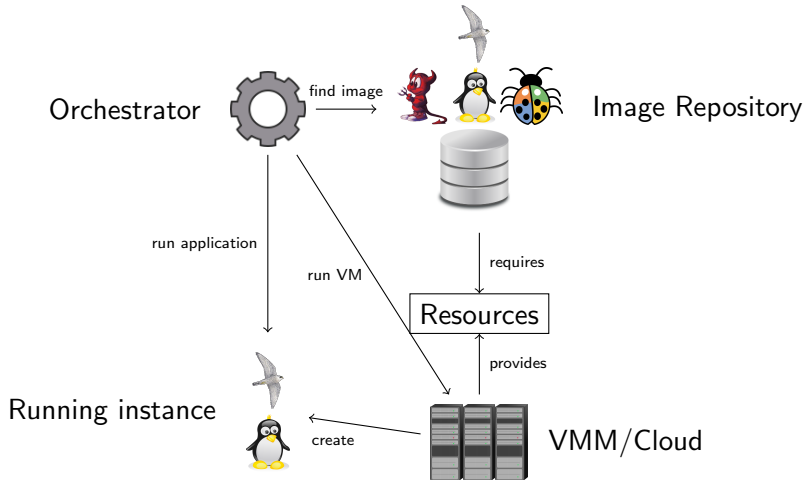
Software as a Service

In order to achieve full SaaS



Software as a Service

In order to achieve full SaaS



Related Work

- WNoDeS – Worker Nodes on Demand Service
 - LRMS integration
 - gLite Worker Nodes
 - Dynamic Provisioning of Virtual Worker Nodes
 - Use of Grid Resources through Cloud (IaaS) interfaces
- Manageable Dynamic Execution Environments on the Grid Using Virtual Machines
 - Sai Srinivas Dharanikota and Ralf Ratering, 2006
 - similar approach
 - Only OS relevant for image selection
 - Use of Software deployment service
 - abandoned prototype

Summary

- Motivation
 - flexibility in available applications
 - VO specific OS images
 - availability of virtualized hardware
- Design
 - Fitting the UNICORE model
 - Required changes
 - Abstraction of VMM
 - Image Repositories
- Metadata
- Relation to IaaS, PaaS, and SaaS
- Related work

Future Work

- Implementation currently ongoing
- Develop an orchestration service capable of providing full SaaS scenario
- Entire clusters of virtual nodes
 - 10,000 cores possible, would have ranked #114 in 2010 Top 500 list
 - <http://bit.ly/gouqdi>
 - Will need more complex setup
 - Multi-core nodes will be possible easily