

The VMI Definition & Query Language for VMRC

Instituto de Instrumentación para Imagen Molecular
Área de Grid y Computación de Altas Prestaciones
Universitat Politècnica de València
Camino de Vera S/N 46022 Valencia (SPAIN)

Revisión 08/04/2014

Latest version of the document: <http://www.grycap.upv.es/vmrc>

Content

Introduction	2
General Language Description	2
Specific Features to Describe the VMI Requirements.....	3
Language Examples	4
Describing a VMI	4
Specifying the Requirements to search for a VMI.....	5

Introduction

This document describes the language employed to express the capabilities of the Virtual Machine Images (VMIs) and the language employed to query the catalog to search for the appropriate VMIs that satisfy the given specified requirements.

General Language Description

The language is structured as a key-value one, where the properties and features of the VMI are expressed. To specify the requirements that a VMI has to meet, the following operators can also be employed ($<$, $<=$, $>$, $>=$, $\&\&$ and $||$). The keys are hierarchically structured where the dot (.) is employed as a separator between the category and the subcategory.

The values are not subject to a previously defined ontology. Instead, the user is free to assign the specific semantics to each attribute considering the application domain. This way, the language can be employed in different fields or future areas. As an example, restricting the possible values to name an Operating System (OS) would require modification of the language when a new OS appears.

The categories and their specific subcategories are explained:

- system
 - Includes general information about the VMI
 - Possible keys:
 - system.name: The name of the VI
 - system.hypervisor: The hypervisor for which this VMI was built.
 - system.location: An URL that indicates where this VMI can be located (it can be a file:// or an http:// or even a user-defined such as ec2://)
- cpu
 - Includes information about the CPU features of the VMI
 - Possible keys:
 - cpu.arch. The CPU architecture.
- disk
 - Includes information about disk of the VMI.
 - Possible keys:
 - disk.size: The size of VMI's disk.

- `disk.os.name` : Type of Operating System (i.e. Windows, Linux, Mac OS X)
- `disk.os.flavour` : Flavour of the OS (i.e. XP, Ubuntu, Leopard)
- `disk.os.version` : Version of the OS (i.e., SP1, 9.10, 10.5.4)
- `disk.os.credentials.user` : User name to access the VMI
- `disk.os.credentials.password` : Password of the account.
- `disk.applications` : The applications installed in the VMI

Use the following syntax to specify the applications:

disk.applications contains (name = com.mathworks.matlab, version = 8.0, path = /opt/matlab/bin/matlab8)

The path is an optional attribute.

Specific Features to Describe the VMI Requirements

To describe the application requirements, these can be classified in two groups by the user.

- **Hard requirements:** These are the requirements that must be satisfied by the VMI (i.e. Operating System)
- **Soft requirements:** These are optional features that would be interesting to be met by the VMI, but that they are not mandatory. For example, this is the case of specific application that could be already installed in the VMI. This is typically not a mandatory requirement, since contextualization can be employed to deploy the application afterwards.

By default, requirements are considered hard unless otherwise specified by the user. The soft requirements are ranked by the user. This rank is completely dependent on the user, who expresses the rank value assigned to each soft requirement. This way, the user can indicate which soft requirements are more important than others in order to perform matchmaking.

If you want to specify an application requirement, use the following syntax.

disk.applications contains (name = Java, version = 1.6)

In this case, it is possible to use the operators (<,<=,>,>=) to specify a greater range of possible values.

Language Examples

Describing a VMI

This section exemplifies the usage of the language. The first example (shown in Figure 1) provides the description of a VMI created by the KVM hypervisor with a CPU architecture i686, and a disk size of 10000 Mbytes. The operating system is Ubuntu Linux 11.10 with a user account created with the username jdoe and password janed0e!!. The VMI has the following applications installed (java-jdk version 1.6, ant version 1.8.1. and Apache Tomcat version 7.0.1).

```
system.name = MyImage7
system.hypervisor = kvm
system.location = /opt/vm_images/dummy_img.qcow2
cpu.arch = i686
disk.size = 5000
disk.os.name = Linux
disk.os.flavour = Ubuntu
disk.os.version = 11.10
disk.os.credentials.user = jdoe
disk.os.credentials.password = janed0e!!
disk.applications contains (name = org.apache.tomcat, version = 7.0.1 )
disk.applications contains (name = org.apache.ant, version = 1.8.1 )
disk.applications contains (name = com.java-jdk, version = 1.6, path = /bin/java )
```

Figure 1. Usage example to describe a VMI.

It is important to point out that the VMRC catalog doesn't check that the actual features of the VMI specified by the user actually match the existing VMI. The user is responsible for providing this correct information..

In addition, the meaning of the attributes can be altered for a given application domain. For example, the attribute system.disk might represent the amount of free available space (instead of the disk size). Of course, changes in the semantics must be taken into account by the user.

Additionally, declaring an application without version can be used as a simple and effective mechanism to tag a VM. These can be of importance during the matchmaking process in the case the users desires a specific VMI.

Specifying the Requirements to search for a VMI

The following example shows how the language should be employed to specify the requirements that a given VMI should met in order to be a candidate one to be selected by the matchmaking process. The VMI must have been created for the KVM hypervisor. The VMI must have a disk size greater than 5000 Mbytes.

Concerning the software, the OS should be Linux and, if it is possible, an Ubuntu Linux greater or equal to 9.10. The matlab application should be installed (no specific version). It also requires the Java JDK 1.6+. Additionally, the existence of the Apache Tomcat application installed (version greater than 7.0) should be ranked favorable.

```
system.hypervisor = kvm
cpu.arch = i686
disk.os.name = Linux
disk.os.flavour = Ubuntu
disk.os.version >=9.10
disk.applications contains (name = org.java-jdk, version >= 1.6)
soft 25 disk.applications contains (name = org.apache.tomcat, version > 7.0)
```

Figure 2. Specifying the requirements to search for candidate VMIs.