Active Agents Programming in HARNESS

Mauro Migliardi[♦]

Emory University, Dept. Of Math & Computer Science
1784 N. Decatur Rd. #100
Atlanta, GA, 30322, USA
Phone +1 404 727 4975
Fax +1 404 727 5611
E-mail om@mathcs.emory.edu

Vaidy Sunderam[♥]

Emory University, Dept. Of Math & Computer Science
1784 N. Decatur Rd. #100
Atlanta, GA, 30322, USA
vss@mathcs.emory.edu

[♦] Corresponding author.
[♥] Presenting author.

Active Agents Programming in HARNESS

Mauro Migliardi, Vaidy Sunderam

Emory University[1], Dept. Of Math & Computer Science
Atlanta, GA, 30322, USA
om@mathcs.emory.edu

Abstract. HARNESS is an experimental Java-centric metacomputing system based on distributed object oriented technolgy. Its main features are the capability to dynamically assemble, manage and dismantle virtual machines that are reconfigurable both in terms of resources enrolled and in therms of services offered, and the capability to mix and match heterogeneous programming models on demand. In this paper we describe how the HARNESS system supports the active, mobile agents programming model.

1 Introduction

Harness [1][2] is an experimental metacomputing system based upon the principle of dynamically reconfigurable object oriented [3] networked computing frameworks. Harness supports reconfiguration not only in terms of the computers and networks that comprise the virtual machine, but also in the capabilities of the VM itself. These characteristics may be modified under user control via an object oriented "plug-in" mechanism that is the central feature of the system. At system level, the capability to reconfigure the set of services delivered by the virtual machine allows overcoming obsolescence related problems and the incorporation of new technologies. In fact, a virtual machine model intrinsically incorporating reconfiguration capabilities addresses these issues in an effective manner. At application level the reconfiguration capability of the system allows the incorporation of new capabilities into applications directly at run-time. This capability is extremely useful for long running applications (e.g. simulations) that need to evolve to adapt to new data and constraints.

The native programming model of HARNESS is distributed, Object Oriented components. However, the modular, Object Oriented nature of the HARNESS system allows on demand mix and match of components or complete applications adopting programming models that are different from its native one (e.g. PVM [4]). This process is supported by means of compatibility suites that can be plugged into the system if the need for such a programming model arise. The feasibility of this approach has been already proved in past works [5].

In this paper we describe the compatibility suite for the active, mobile agents programming model [6][7].

The paper is structured as follows: in section 2 we give an abstract overview of the system architecture; in section 3 we describe our compatibility suite for active agents

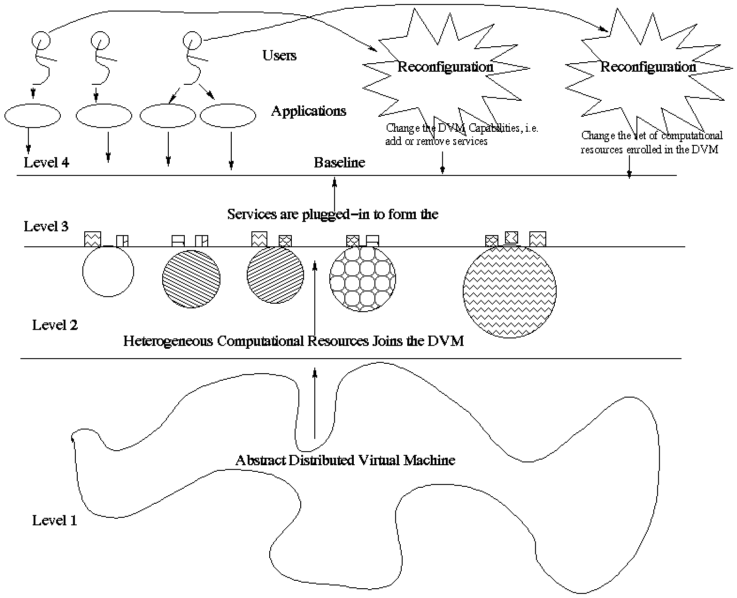
Users
Applications
Reconfiguration
Reconfiguration
Change the DVM Capabilities, i.e.
add or remove services
Change the set of computational
resources enrolled in the DVM
Level 4
Baseline
Services are plugged−in to form the
Level 3
Level 2
Heterogeneous Computational Resources Joins the DVM
Abstract Distributed Virtual Machine
Level 1

Fig. 1.

programming; finally, in section 4, we provide some concluding remarks.

2 The HARNESS Metacomputing System

The fundamental abstraction in the HARNESS metacomputing framework is the **Distributed Virtual Machine** (DVM) (see figure 1, level 1). Any DVM is associated with a symbolic name that is unique in the HARNESS name space, but has no physical entities connected to it. **Heterogeneous Computational Resources** may enroll into a DVM (see figure 1, level 2) at any time, however at this level the DVM is not ready yet to accept requests from users. To get ready to interact with users and applications the heterogeneous computational resources enrolled in a DVM need to load **plug-ins** (see figure 1, level 3). A plug-in is a software component implementing a specific **service**. By loading plug-ins a DVM can build a **service baseline** (see figure 1, level 4) that is consistent with applications' requrements. Users may **reconfigure** the DVM at any time (see figure 1, level 4) both in terms of computational resources enrolled by having them **join** or **leave** the DVM and in terms of services available by **loading** and **unloading** plug-ins.

The availability of services to heterogeneous computational resources derives from three different properties of the framework: *i)* the portability of plug-ins, *ii)* the portability of the runtime system and *iii)* the presence of multiple searchable repositories. HARNESS implements these properties leveraging two different features of Java technology, i.e. the capability to layer a homogeneous architecture such as the Java Virtual Machine [8] over a large set of heterogeneous computational resources, and the capability to customize the mechanism adopted to load and link new objects and libraries.

For further details about the HARNESS metacomputing framework we direct the attention of the interested readers to [2].

3 Active Agents Programming Support

Active mobile agents programming is a very promising approach to Internet computing and distributed computing in general. Among the advantages of the mobile agents paradigm with respect to traditional distributed computing programming such as message passing we can cite its inherent adoption of Object Oriented programming and its capability to move the computation to the data instead of the traditional reverse approach. The latter property offers advantages both in the minimization of the bandwidth required as well as in its capability to survive to intermittent network partitioning by moving the computational entity at the site where the data reside. For further details about the agents paradigm we direct the intersted reader to [6] and [7].

To support the agents programming paradigm in HARNESS we have developed the HARNESS active mobile agents compatibility suite. A compatibility suite is a pluggable execution environment that can be loaded into the HARNESS system on demand to support additional programming paradigms. The Java-centric nature of HARNESS together with its capability to soft install both Java classes and native code retrieved from trusted repositories suits very well the mobile active agents programming paradigm. The HARNESS active mobile agents compatibility suite provides to users:

- the capability to load, initialize and start an agent;
- a system-wide unique naming for agents;
- the capability to perform a weak migration of agents;
- the capability to track the movements of an agent through an event system;
- the capability to obtain an RMI reference of an agent;
- a coordination system based on Sun’s JavaSpaces [9].

We adopted a weak approach to the problem of agents migration in our compatibility suite because the current definition of the Java platform does not provide an architecture independent format for the execution stack of a thread. Thus our compatibility suite does not service a request for migration until all the threads of the agent have acknowledged that they are ready to migrate.

The capability to track the movements of an agent allows other agents and applications to retrieve RMI references of RMI enabled agents and interact with them in a direct, synchronous way. The coordination system based on JavaSpaces allows agents to interact asynchronously.

The compatibility suite is currently in alpha testing, however, our first tests showed that the time required to perform migration between two Sun Ultra60 workstations connected over a 100 Mbs ethernet LAN is 100 milliseconds.

4 Concluding Remarks

The modular, object oriented nature of the HARNESS system allows on demand mix and match of components or complete applications adopting programming models. This process is supported by means of compatibility suites that can be plugged into the system if the need for such a programming model arise. In this paper we have described the HARNESS active mobile agents compatibility suite. This compatibility suite allows the execution of agents based applications on a HARNESS virtual machine. The mobile agents programming paradigm is one of the most promising new approaches to distributed programming and is extremely well suited to Internet computing. Our plug-in based approach leverages the capability of the system to support weak migration of agents and multiple agents interaction mechanisms. The initial testing of our implementation shows a very low overhead for agents migration.

References

1 M. Migliardi, V. Sunderam, A. Geist and J. Dongarra, Dynamic Reconfiguaration and Virtual Machine Management in the Harness Metacomputing Framework, proc. Of ISCOPE98, Santa Fe (NM), December 8-11, 1998.

2 M. Migliardi and V. Sunderam, Heterogeneous Distributed Virtual Machines in the Harness Metacomputing Framework, Proc. of the Heterogeneous Computing Workshop, IEEE Computer Society Press, pp. 60-73, 1999.

3 G. Booch, Object Oriented Analysis and Design with Applications, Second Edition, Rational, Santa Clara, California, 1994.

4 Geist A., A. Beguelin, J. Dongarra, W. Jiang, B. Mancheck and V. Sunderam, PVM: Parallel Virtual Machine a User’s Guide and Tutorial for Networked Parallel Computing, MIT Press, Cambridge, MA, 1994.

5 M. Migliardi, V. Sunderam, PVM Emulation in the Harness Metacomputing System: a Plug-in Based Approach, Proc. of 6th EuroPVM-MPI99, Lecture Notes in Computer Science, Vol. 1697, pp. 117-124, Springer Verlag, 1999

6 J. White, Mobile Agents Whitepaper, MIT Press, Menlo Park, 1996.

7 Proc. of the International Workshop on Mobile Agents, Lecture Notes in Computer Science, N. 1219, Springer Verlag, 1997.

8 T. Lindholm and F. Yellin, The Java Virtual Machine Specification, Addison Wesley, 1997.

9 Sun MicroSystems, Jini Starter Kit documentation package, available on line at http://developer.sun.com/developer/products/jini/jinidoc.html