Towards a Data-Driven Coordination Infrastructure for Peer-to-Peer Systems

Nadia Busi, Cristian Manfredini, Alberto Montresor, and Gianluigi Zavattaro

Dipartimento di Scienze dell’Informazione, Università di Bologna, Mura A.Zamboni 7, I-40127 Bologna, Italy, {busi,manfredi,montresor,zavattar}@cs.unibo.it

Abstract. Shared dataspaces, initiated by Linda since the beginning of the 80s, has been successfully adopted as a coordination model in a huge variety of systems and applications, going from parallel computing to web-based collaborative work. We point out several scalability problems which arise when trying to exploit the original Linda coordination model in peer-to-peer systems. The objective of this analysis is to produce some guidelines for the design of a data-driven coordination infrastructure suitable for the peer-to-peer scenario.

1 Introduction

The rapid evolution of computers and networks is calling for the development of middleware platforms responsible for the management of dynamically reconfigurable federations of devices, where processes cooperate and compete for the use of shared resources. In this scenario one of the most challenging topics is concerned with the coordination of the activities performed by the federated components.

Generative communication, realized by means of the insertion and withdrawal of elements from a shared multiset, is the peculiar feature of a family of coordination languages, of which Linda [Gel85] is the most prominent representative. Generative communication is based on the following principles: a sender communicates with a receiver through a shared data space (called tuple space, TS for short), where emitted messages are collected; the receiver can consume the message from TS; a message generated by a process has an independent existence in the tuple space until it is explicitly withdrawn by a receiver; in fact, after its insertion in TS, a message becomes equally accessible to all processes, but it is bound to none.

In the last decades, the shared dataspace approach has been successfully adopted in a huge variety of systems and applications, going from parallel computing to Web-based collaboration system. Recently, this communication mechanism has been adopted also by several proposals of coordination platforms (see, e.g., Sun Microsystems JavaSpaces [W+98] or the IBM T Spaces [WMLF98]) for the management of dynamically reconfigurable federations of devices, where processes cooperate and compete for the use of shared resources.
In this paper we investigate the scalability of this coordination approach to the realm of peer-to-peer systems.

Informally, peer-to-peer (P2P) systems are distributed systems based on the concept of resource sharing by direct exchange between peer nodes (i.e., nodes having the same role and responsibility). Exchanged resources include content, as in popular P2P file sharing applications [Shi01, Kan01, Lan01], and storage capacity or CPU cycles, as, for example, in computational and storage grid systems [And01, RD01, K+00].

Distributed computing was intended to be synonymous with P2P computing long before the term was invented, but this initial desire was subverted by the advent of client-server computing popularized by the World Wide Web. The modern use of the term P2P and distributed computing as intended by its pioneers, however, differ in several important aspects. First, P2P applications reach out to harness the outer edges of the Internet and consequently involve scales that were previously unimaginable. Second, P2P by definition, excludes any form of centralized structure, requiring control to be completely decentralized. Finally, and most importantly, the environments in which P2P applications are deployed exhibit extreme dynamism in structure, content and load. The topology of the system typically changes rapidly due to nodes voluntarily coming and going or due to involuntary events such as crashes and partitions. The load in the system may also shift rapidly from one region to another, for example, as certain files become “hot” in a file sharing system; or the computing needs of a node suddenly increase in a grid computing system.

2 Shared Dataspaces in Mobile Systems

The pervasiveness of the client-server architecture also affected the design of shared dataspaces based coordination infrastructures. Indeed, in most of the currently available implementations of Linda-like systems, the dataspaces metaphor is intended as a (centralized) repository service.

An interesting proposal breaking the client-server bias is represented by the transiently shared dataspaces metaphor introduced in Lime [PMR99].

Lime [PMR99] (Linda in a Mobile Environment) is a coordination middleware supporting both logical and physical mobility. It provides programmers with a Linda-like dataspaces, whose content is determined by the connectivity among mobile hosts.

It is reasonable to investigate the scalability of Lime to a peer-to-peer context, characterized by dynamically changing connectivity, significant autonomy for the processes and direct communication between them.

The following discussion takes into consideration three different aspects of coordination models: coordinables (what is coordinated), coordination rules, coordination medium.

As far as coordinables are concerned, in a Linda-like system, the coordinated entities are usually active programs called agents. In Lime, agents with their own local dataspaces reside on hosts and are able to logically move from a host to