Abstract. This paper presents a description of the concepts and functionalities of the DOSMOS-Trace monitoring tool. The designing choices lead us to introduce the concepts of Event Manager Processes and Meta-Objects. It allows to keep a good scalability of the system but also to get a low intrusive monitoring tool.

1 Introduction

By gathering the advantages of shared memory systems and distributed memory systems, Distributed Shared Memory (DSM) systems offer an intuitive and easy-to-use programming environment and therefore receive increasing attention. The total transparency of shared data accesses hides the behaviour of its application (and of the system) to the programmer. A monitoring tool would be a great help for him to better understand how the whole system (i.e. the application plus the DSM system) reacts. There are many monitoring tools but are essentially designed for message passing systems. In that framework, the relevant information is related to the sequence of "send" and "receive", but for DSM systems, it concerns the operations performed on shared data. Thus, this paper presents the concepts and the functionalities of our DSM monitoring tool, DOSMOS-Trace (section 3). This tool was developed to monitor the DOSMOS DSM system which is quickly described in section 2. Section 4 concludes this paper by giving the main direction of our future work.

2 The DSM System Model

Developed on top of PVM, DOSMOS (Distributed Objects Shared MemOry System) [1] already benefits from the portability property.

The four main features of DOSMOS are:

Variable sharing: The system allows the sharing of user defined variables in a totally transparent way. Basic types of data (i.e. integers, reals) but also static arrays can be declared.

The array splitting up: This feature is provided in order to reduce the contention on variable accesses in case of false-sharing.
Weak memory consistency model: The system allows variable replication. Thus, to ensure a consistency between the variable copies, we have implemented a weak consistency model (the release consistency) which permits to avoid expensive synchronizations and useless communications.

Hierarchic groups: The idea is to gather the processes that access regularly to a common set of variables. Only processes belonging to the group of a variable are allowed to get a copy. Due to the process clustering, the number of variable copies is limited and the cost of the consistency maintenance is bounded. This allows a good system scalability.

Figure 1 presents the software configuration of a DOSMOS application. It is composed by three types of processes:

- **Application Processes (AP)** which contain the code of the user application (in C) plus code for communications with the Memory Processes.
- **Memory Processes (MP)** which are in charge of the management of the shared objects. It answers the access requests sent by the Application Processes.
- **Link Processes (LP)** are gateways for a Memory Process which wants to access a variable not present in the same group.

3 The Monitoring Tool: DOSMOS-Trace

3.1 The Concepts

In order to provide a low intrusive and scalable monitoring tool, we choose to add a new kind of process in the system called Event Manager Processes (EMP). These processes receive information from the Memory Processes about variable accesses. Depending on the further use of the received trace data, they can be