The *PDBG* Process-level Debugger for Parallel and Distributed Programs*

João Lourenço José C. Cunha
{*jml, jcc*}*@di.fct.unl.pt*
Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
Portugal

On Debugging Parallel and Distributed Programs

In this paper we discuss several issues concerning the design and implementation of a debugger for parallel and distributed applications. This debugger is based on a client-server approach to isolate the debugging user-interface from the debugging services, and has a two-level structure.

A distinctive feature of our work is that we are following a two-level structured approach to achieve this goal:

1. *Component-level.* At this level, the performed debugging actions act upon individual application components, such as processes and threads.
2. *Coordination-level.* At this level, coordination groups of processes and/or threads is provided, and the inspection and control commands can observe and control distributed processes interactions.

In this paper we are focusing in the component-level (processes) of a parallel/distributed program. The main aspects we are addressing are:

- *Interfacing with other tools and environments.*
  The use of a debugging tool should be complemented by other software development tools, such as testing or visualization tools. The consistent integration of such tools provides an environment with new functionalities resulting from the exploitation of the interactions between tools.
- *Heterogeneity.*
  The components of a distributed application may span a large set of machines, with different hardware and operating systems, using multiple communication protocols and can rely on different programming models, of the logical, functional and procedural styles. A debugger for distributed applications must be flexible and adaptable in order to support all such levels of heterogeneity.

The *PDBG* Process-level Debugger

The PDBG debugger is a distributed debugger that provides the following classes of services:

1. Debugging session control services;
2. Process execution control services;
3. Process internal state inspection and modification;
4. Event processing and management services.

The interactions between client tools and PDBG have two distinct facets: synchronous and asynchronous:

- *Synchronous interactions.* For each procedure invocation a status code is returned indicating success or unsuccess in the processing of the requested service.
- *Asynchronous interactions.* To handle deferred commands, or commands that take an unpredictable amount of time to be executed, *e.g.* continue the execution of the process until a breakpoint is reached or the process terminates. Client tools can register their interest in being notified about what's happening with the target processes. This notification corresponds to *Events* that will activate associated handlers. As PDBG supports connections from multiple client tools, events are also used to support tool coordination, so that coherent views on the same target application can be provided.

PDBG Implementation

The current implementation runs over DAMS (*Distributed Applications Monitoring System*) which is a *Monitoring and Control layer* (see [2]), which support incremental development of services. For the implementation of PDBG, the DAMS system was extended with a new *Debugging Module* that implements the debugging services, and a set of coordination-level modules, such as *Replayer Module* or a *Race-detection Module* can extend both control and inspection functionalities of PDBG.

Conclusions

The most distinctive aspects of the our approach are the provision of a framework to support the experimentation with the following aspects:

- Incremental development of debugging functionalities;
- Support for heterogeneity;
- Tool interfacing and integration.

Due to its flexibility, PDBG can be used to implement debugging specifications such as HPDF [1].

References

[1] J. Brown, J. Francioni, and C. Pancake. White paper on formation of the high performance debugging forum. Available in "http://www.ptools.org/hpdf/meetings/mar97/whitepaper.html", February 1997.

[2] J. C. Cunha, J. Lourenço, J. Vieira, B. Moscão, and D. Pereira. A framework to support parallel and distributed debugging. In *Proceedings of the International Conference on High-Performance Computing and Networking (HPCN'98)*, Amsterdam, The Netherlands, April 1998.

*In "*The 2^{nd} SIGMETRICS Symposium on Parallel and Distributed Tools*," Welches, Oregon, USA.