Design and Development of Prototype Components for the Harness High-Performance Computing Workbench

by

Ronald Baumann

March 16, 2006

Supervisors:
Prof. V. Alexandrov, University of Reading
G. A. Geist, Oak Ridge National Laboratory
Contents

- Introduction
- Definition of Parallel Plug-ins
- Motivation and Features
- Types of Parallel Plug-ins
- Scientific Applications and Parallel Plug-ins
- System Design of a Prototype Parallel Plug-in Suite
- Fault-Tolerant Design
- Conclusions
Parallel computing is used to solve large-scale problems

Enormous amounts of computational performance for simulations and modeling in aerospace, medicine, nanotechnology and material science

Development of parallel algorithms and software

Software organizes the resources and handles the communication of application components

Two well-known parallel computing packages, Parallel Virtual Machine (PVM) and Message Passing Interface (MPI)
Introduction – Previous Work

- Harness is a follow-on to PVM
- An ongoing, collaborative work between the Oak Ridge National Laboratory (ORNL), University of Tennessee Knoxville (UTK), and Emory University
- Provides a pluggable, heterogeneous Distributed Virtual Machine (DVM) environment
- Organizes programs and services by using plug-in software modules
- Consists of a lightweight kernel (runtime environment)
Introduction – Requirements and Objectives

- Parallel plug-ins assemble applications and provide services

- The main objectives of this Master thesis project were
  - Gaining experiences in the research area of parallel plug-ins
  - Using the available functions provided by Harness
  - Exploring the new aspects of fault tolerance that arise from the use of dynamic parallel plug-ins.

- Plug-ins must meet different requirements, such as inter plug-in communication, plug-in (un)loading and fault tolerance
Definition of Parallel Plug-ins

Parallel Plug-ins

Harness

Daemon Process
- External Process
- Startup and Control:
  - Process Manager
  - Worker Threads:
    - Thread Pool
  - Dynamically Loaded Plug-Ins:
    - Plug-In Loader
    - Some Plug-In

Running Processes:
- External Process
- Forker Process

Loadable Plug-Ins:
- Some Plug-In
- Distributed Control Plug-In
- FT-MPI Plug-In

Today’s plug-in technology

- Plug-ins interact with other programs providing certain, specific functions
- Extends software to provide new features
- Usually integrated over a well-defined interface
- Possible way to overcome bulky software
Motivation and Features

- New design attempts to use the advantages of both technologies
- Providing a design pattern for parallel plug-ins and basic parallel plug-in frames
- Parallel plug-ins build up entire, modularized, pluggable applications
- Facilitating the modularization of huge and complex software systems
- Parallel plug-ins are independent from other modules of the complete system
- Benefits: code reuse and lightweight software
Types of Parallel Plug-ins

- Distributed parallel plug-in
- Replicated parallel plug-in
- Service plug-in
Parallel Plug-ins and Scientific Applications

Monte Carlo Integration

- Used in many simulations in physics or other scientific disciplines
- Based on numerically generated pseudo random numbers
- Simplifies the calculation of complex integrals

\[ I = \int_{a}^{b} g(x) \, dx \]

\[ I \approx I_{MC} = \frac{b-a}{n} \sum_{i=1}^{n} g(x_i) \]
Parallel Plug-ins and Scientific Applications

Image Processing Pipeline

- Needs a lot of processing power for the calculation of certain filter functions
- Mostly, images are processed with a variation of filters
System Design of a Prototype Parallel Plug-in Suite

Architecture of the Parallel Plug-in Prototype Suite

Design of the Prototype Suite Components
System Design – Parallel Plug-in for Integral Computation

Integration Application

Fault-tolerant Design
System Design – Parallel Plug-in for Image Processing

Multiple Image Processing Pipeline with Acknowledgment and Error-Handling
Fault-Tolerant Design

- Continue the operation of a system after a failure occurred
- Partial success: not all parallel plug-in components were loaded
- PPM handles partial success depending on the loaded parallel plug-in
- Redistribution of missed integral intervals
- Restoration of a broken image processing pipeline
Conclusions

- Plug-in technology was successfully joined with Harness
- Various facets of parallel plug-in technology were presented
- Three major problems or features of parallel plug-ins were investigated and realized
  - (Un)loading of parallel plug-ins
  - Inter plug-in communication
  - Fault-tolerant design
- Scalability and reliability through fault-tolerant design, including partial success
- The final version of the paper draft is sent to the HPCC 2006 in Munich