

# Cluster Grid based Response-time analysis module for the PIPE Tool

Harini Kulatunga      Ashok Argent-Katwala  
William Knottenbelt

Department of Computing, Faculty of Engineering  
Imperial College London, United Kingdom  
{hkulatun, ashok, wjk}@doc.ic.ac.uk

## Abstract

*Generalized Stochastic Petri Nets (GSPNs) are a widely used high-level formalism used for modelling discrete-event systems. The Platform Independent Petri net Editor (PIPE) is an open source software project that allows creation, analysis and simulation of Petri Nets. This tool paper presents a PIPE module for response-time analysis of a Petri net's underlying Continuous Time Markov Chain (CTMC). Jobs are submitted via a web interface, from within PIPE or from a browser. The parallel computations are run using Grid Engine on a cluster hosted at Imperial College London.*

## 1. Introduction

The PIPE tool can be used for modelling GSPNs and includes many analysis modules and the ability to generate reachability graphs. This tool has been downloaded over 7200 times and is in use around the world for teaching and analysis. It is an active open source project on Sourceforge [5], with a number of contributors, regularly reported bugs and a features wish list. A user has the ability to compose a GSPN model using PIPE from which the underlying CTMC can be obtained.

With the advent of high performance parallel computing direct numerical analysis of full response-time densities in structurally unrestricted Markov models has become a reality in areas such as QoS metrics. Combining hypergraph partitioning and uniformization, as done in the Hypergraph-based Distributed Response-time Analyzer (HYDRA) [1], has enabled a parallel implementation for computing passage-time densities in Markov chains with very large state spaces.

It is the aim of this paper to bring together the popular GSPN tool, PIPE, and a high performance cluster

computing resource. This provides a new analysis capability that enables web-driven response-time performance analysis. A user submits their job over the web from with the PIPE tool, and the analysis is executed on a grid cluster computer.

## 2. Tool overview

The PIPE tool includes a Graphical User Interface (GUI) by which users are able to specify a Petri net which conforms the Petri net Markup Language (PNML) standard. The graphical editor is intuitive and can be used to create, save and load Petri nets. The application also contains an animator so that the user can manually fire enabled transitions within the Petri net. Pluggable C++ modules that operate on the PNML data layer can be integrated at runtime.

The Camelot cluster at Imperial College, funded by EPSRC grant *EP/D505933/1* for project 'GRID-Enabled Performance Analysis using Stochastic Logics' (GRAIL), is comprised of 16 dual processor, dual core nodes with 8GB of RAM and a file server of 4.7TByte. The nodes are connected with both gigabit ethernet and Infiniband (2.5Gbit/s) interfaces.

The cluster grid is the simplest form of a grid and a Distributed Resource Management (DRM) system enables this as a grid computing service [3]. The Sun Grid Engine (SGE) software is open source DRM software and a core component of the Sun Cluster Grid software. The HYDRA Markov chain analyzer [1], a C++ parallel tool developed by the Analysis, Engineering, Simulation & Optimization of Performance (AESOP) group, was implemented on the Camelot Cluster Grid. The implementation executes its parallel jobs within the SGE distributed environment. The front-end for the 'HYDRA cluster' is a web interface which can be invoked from the PIPE tool or accessed directly over the web at <http://dev.performdb.org/tools/grail/>. The interface also requires user input speci-

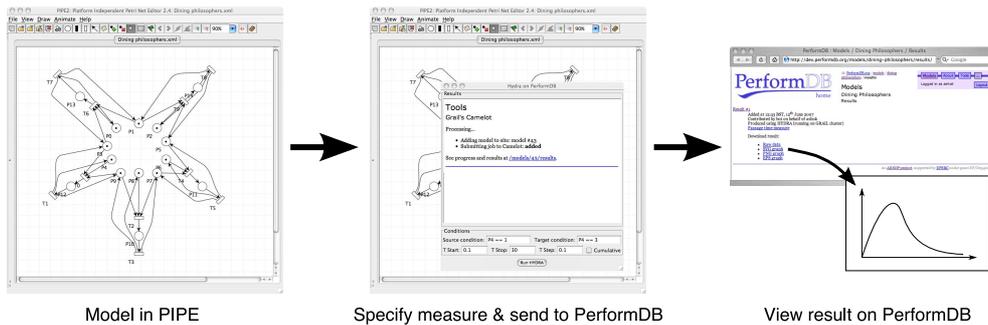


Figure 2. Performance analysis pipeline using the PIPE tool and PerformDB.

ifying the passage-time parameters and number of processors in the cluster for analysis. The structure of the parallel response-time module for the PIPE tool is summarized in the block diagram of Figure 1.

The high-level model specification translates the PNML data files into a DNAmaca model file [6] that also contain the passage-time parameters of the calculation. The HYDRA tool receives the model file and number of processors from the web interface and produces a hypergraph representation which is used by the PaToH tool [4] for hypergraph partitioning of the underlying state-space. Next the MPI-based [2] uniformization in HYDRA implements the parallel response-time analysis. Finally the MPI jobs are submitted to the SGE Job Management System (JMS) which supports MPI parallel libraries. SGE handles queueing, scheduling, prioritizing, providing job status and execution of MPI jobs. All the results are returned to the master node and they are published on the Per-

formDB site. The user's view of this process is illustrated in Figure 2.

### 3. Response-time analysis of GSPNs

Calculation of response-time density contains two stages, the initial step is to uniformise the CTMC, which results in a 'new' CTMC with exponentially distributed state holding times with the same mean for all states. An extra absorbing state is added during this stage to ensure that only the first passage-time is calculated. Hence the density of time taken between two states is obtained by convolving sojourn time densities along all possible paths in between the two states. This results in an Erlang distribution.

The kernel of HYDRA's operation are matrix-vector multiplications. These are computed in parallel using a hypergraph partition to minimise communication between the nodes.

In this way, the HYDRA tool is capable of handling very large Markov state-spaces in the order greater than  $O(10^7)$ . Further details can be found in [1].

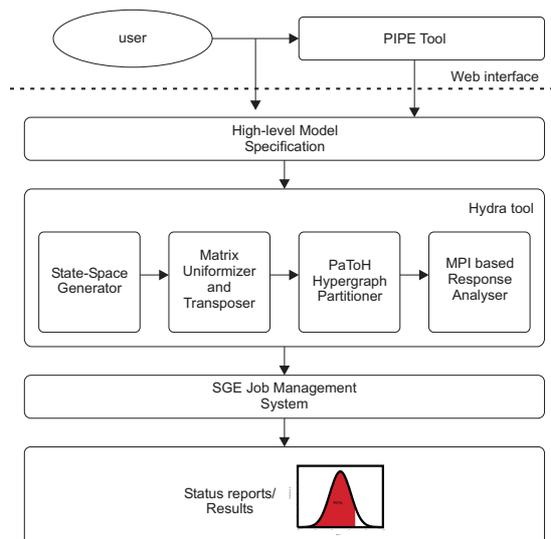


Figure 1. Block Diagram of the Tool

### References

- [1] N. J. Dingle, P. G. Harrison, and W. J. Knottenbelt. HYDRA: HYpergraph-based Distributed Response-time Analyser. In *the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2003)*, pages 215–219, June 2003.
- [2] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI Portable Parallel Programming with the Message Passing Interface*. MIT Press, Cambridge, Massachusetts, 1994.
- [3] Introduction to the Cluster Grid - Part I. <http://www.sun.com/blueprints>.
- [4] PaToH v3.0. <http://bmi.osu.edu/umit/software.html>.
- [5] PIPE Tool. <http://pipe2.sourceforge.net/index.html>.
- [6] W. Knottenbelt. *Parallel Performance Analysis of Large Markov Models*. PhD thesis, Imperial College London, United Kingdom, February 2000.