Novel visualizations for optimization of parallel programs

M.S MABAKANE (1,2), M KUTTEL (2)

(1) Centre for High Performance Computing, CSIR Campus, 15 Lower Hope Street, Rosebank, Cape Town, South Africa

(2) University of Cape Town, 18 University Avenue, Upper Campus, Rondebosch, Cape Town, South Africa

CHPC CENTRE FOR HIGH PERFORMANCE COMPUTING

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

INTRODUCTION

Visualization tools are useful for profiling and analysing parallel software running on different supercomputing systems:

- Distributed-memory systems.
- Shared-memory machines.
- Vector systems.

The purpose of this study is to design visualization techniques to optimize parallel execution of programs and enable optimum performance of parallel programs on the supercomputers.

PROBLEMS AFFECTING PERFORMANCE OF APPLICATIONS

The performance of parallel applications is mainly affected by interplays of factors such as (Adve and Vernon, 2004; Ebnenasir and Beik, 2009):

- Limited network bandwidth.
- Unevenly distribution of message-passing.
- Slow read/write requests within the storage.
- Logic of the parallel code.
- High memory latency in the processing nodes.
- High processor utilisation in the execution nodes.

Adve, V.S., and Vernon, M.K. (2004). Parallel program performance prediction using deterministic task graph analysis. ACM Transactions on Computer Systems. 22(1): 94-136.

Ebnenasir, A., and Beik, R. (2009). Developing parallel programs: a design-oriented perspective. Proceedings of the 2009 IEEE 31st international conference on software engineering. Vancouver: IEEE Computer Society, pp. 1-8.

WHY PROGRAMS DO NOT PERFORM WELL ON SUPERCOMPUTERS?

- Limited parallel programming experience by developers.
- Continuous changes of hardware architecture.
- Application users do not have sufficient knowledge and skills of optimizing programs.

WHY IS IT DIFFICULT TO OPTIMIZE PARALLEL PROGRAMS

It is hard to optimize parallel programs due to the following factors:

- Complex structure of the parallel systems.
- Not easy to identify exact area that cause performance bottlenecks.
- Need thorough understand and analyse of activities such as:
  - Message passing tasks.
  - I/O performance.
  - Network communication.
  - Level of parallelism with the code of the application.

METHODS USED TO ANALYSE PARALLEL PERFORMANCE

- Some of the parallel computing users prefer to utilise visualization and analyses tools (Geimer et al., 2010; Subotic et al., 2010).

- On the other hand, others manually index and search the syntax of the code to increase performance and efficiency of the program (Schwartz-Narbonne et al., 2011).

Geimer, M., Wolf, F., Wylie, B.J.N., Abraham, E., Becker, D., and Mohr, B. (2010). The scalable performance toolset architecture. Concurrency and Computation: Practice and Experience. 22(6): 702-719.

Subotic, V., Sancho, J.C., Labarta, J., and Valero, M. (2010). A simulation framework to automatically analyse the communication-computation overlap in scientific applications. IEEE international conference on cluster computing, edited by: P. Kellenberger. New York: IEEE Computer Society, pp. 275-283.

Schwartz-Narbonne, D., Lui, F., Pondicherry, T., August, D., and Malik, S. (2011). Parallel assertions for debugging parallel programs. 9th IEEE/ ACM international conference on formal methods and models for codesign, edited by: J. Brandt. New Jersey: IEEE Computer Society, pp. 181-190.

TAU's CALLGRAPH VISUALIZATION TOOL

In this study, we investigate the design of TAU (Tuning and Analysis Utilities) callgraph system used to analyse the performance of parallel programs.

TAU's callgraph tool is a visualization system that presents the relationship between different parts (objects) of the program such as message passing functions, modules, routines and subroutines.

It further highlight the execution status of each object used to simulate the program.

OBJECTIVE OF THE STUDY

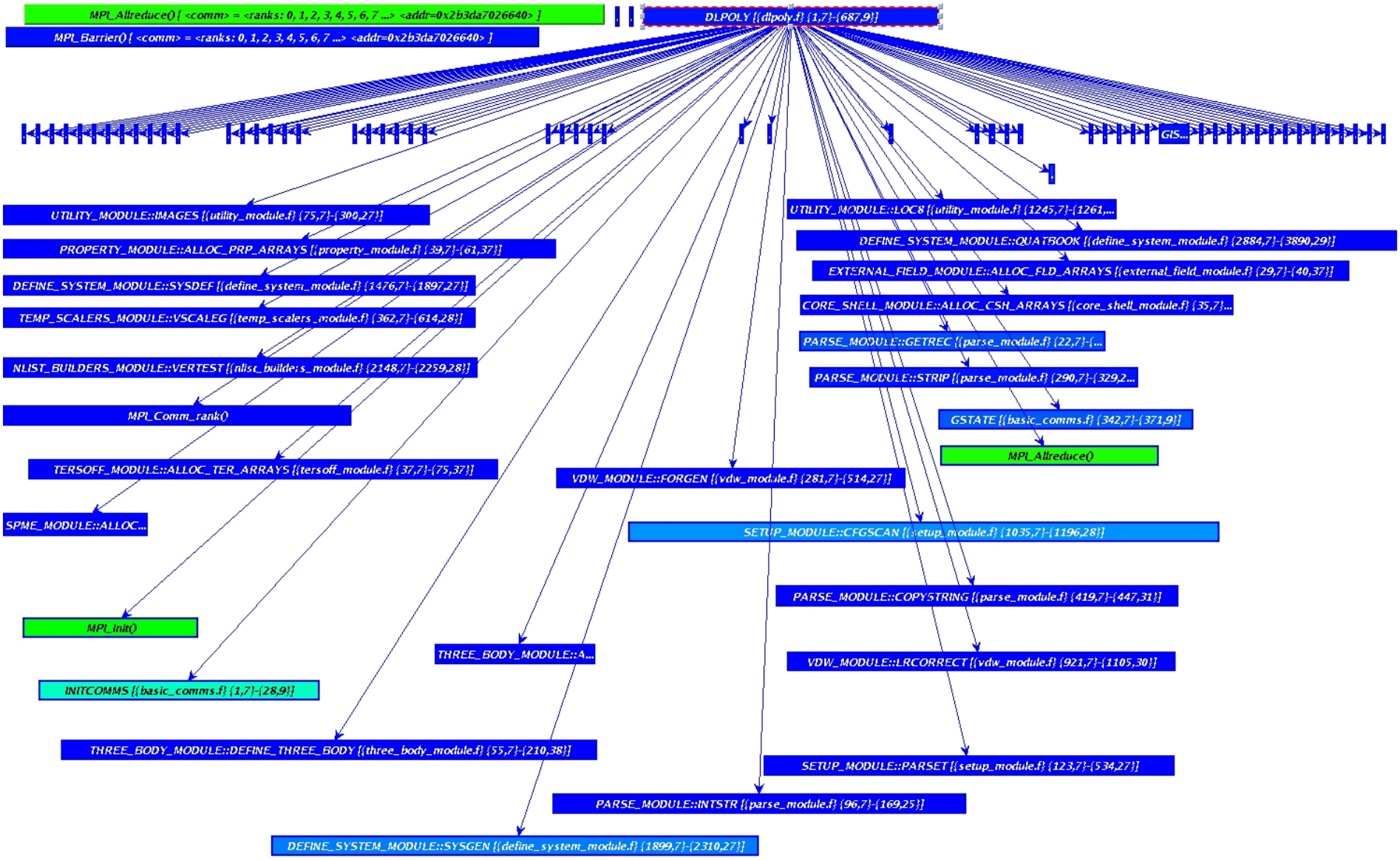
The aim of this work is to develop effective callgraph visualization system that will enable users to efficiently identify performance bottlenecks in parallel programs.

ANALYSES OF TAU'S CURRENT CALLGRAPHS

TAU CALLGRAPH ON DL_POLY_2.18

Below picture demonstrate the first view of the current callgraph used to visualize DL_POLY_2.18:

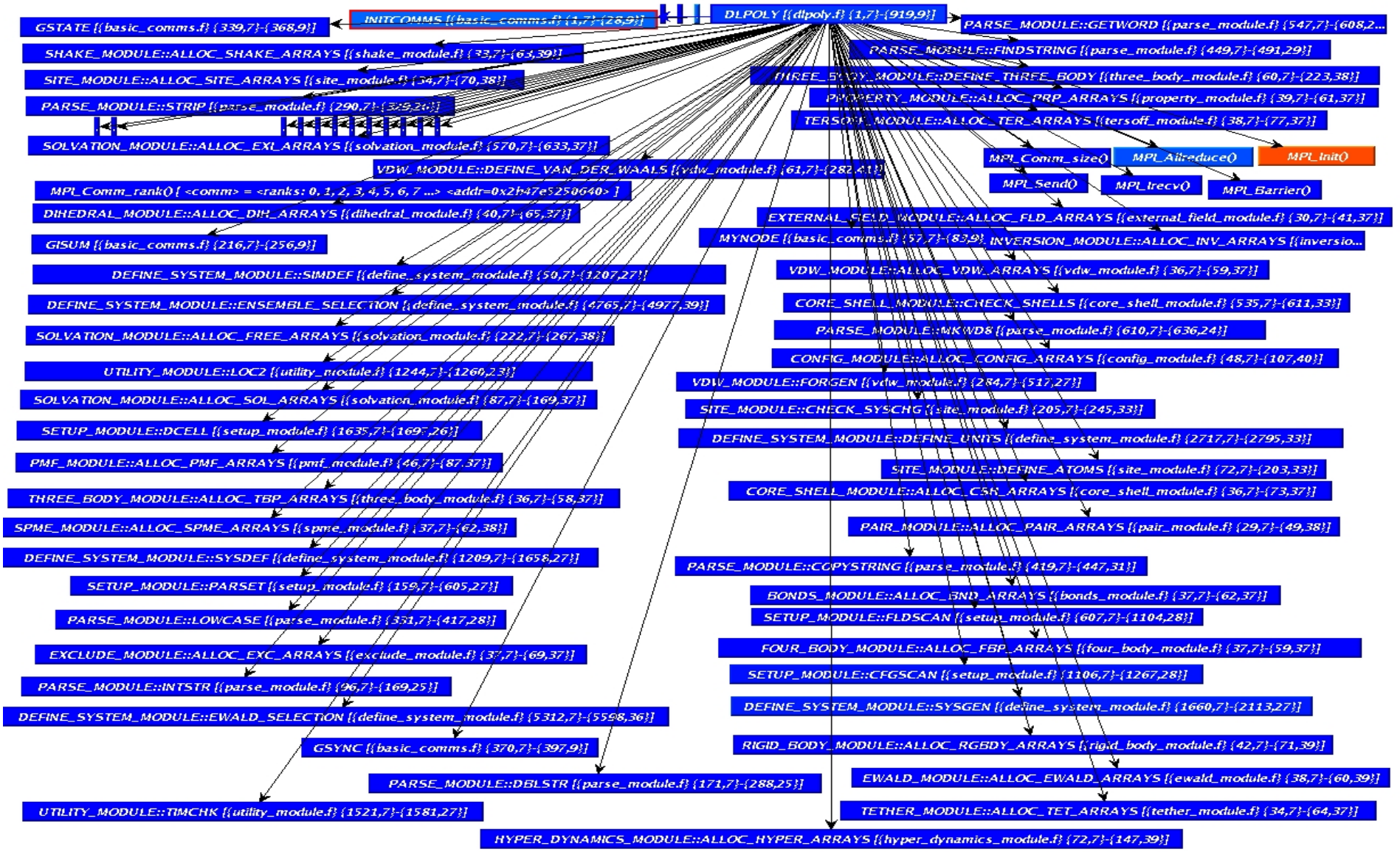
TAU CALLGRAPH ON DL_POLY_2.18 (cont…)



TAU CALLGRAPH ON DL_POLY_2.20

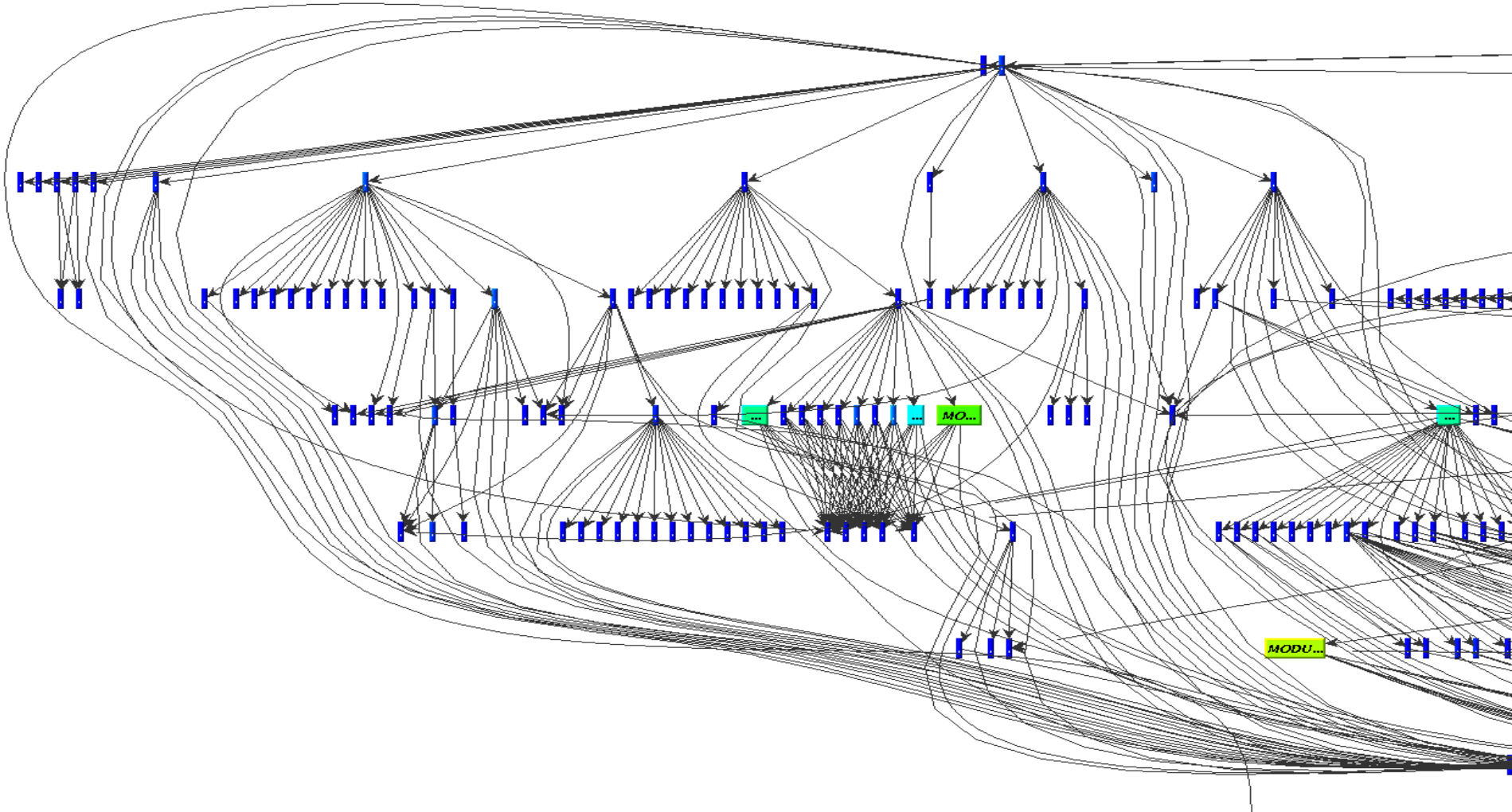
Below picture demonstrate the first view of the current callgraph used to analyse DL_POLY_2.20:

TAU CALLGRAPH ON DL_POLY_2.20 (cont…)

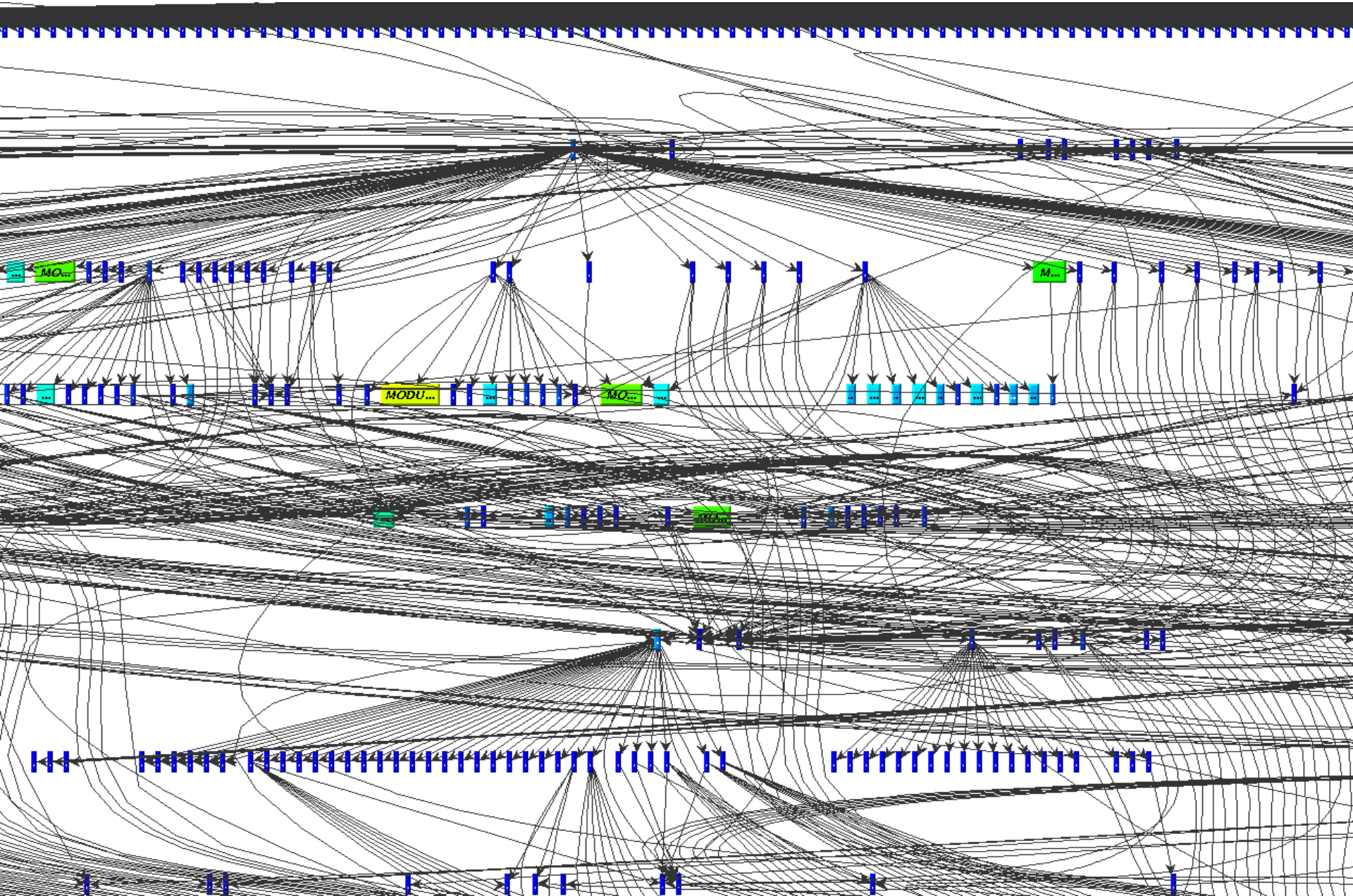


TAU CALLGRAPH (LEFT SIDE) ON WRF-3.5

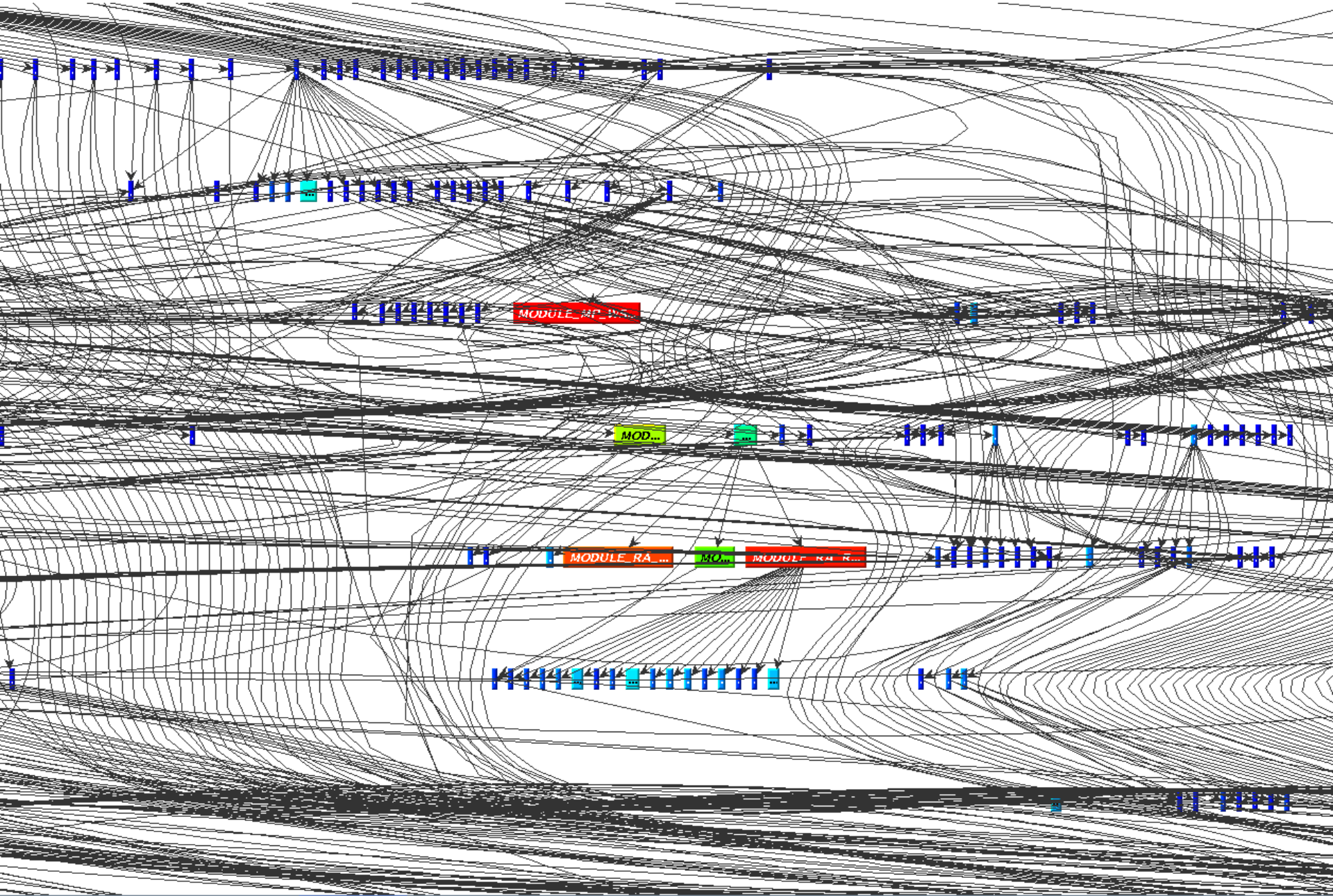
The following picture illustrates the left side of the current callgraph used to visualize WRF-3.5:



TAU CALLGRAPH (CENTRAL AREA) ON WRF-3.5



TAU CALLGRAPH (RIGHT SIDE) ON WRF-3.5



BASIC CRITIQUE OF TAU CALLGRAPHS

The following shows a summary of visual properties used to design the current callgraphs:

- White and black text on different colours of the nodes.
- Dots within the nodes
- Names displayed using italic text style.
- Similar colours (dark blue, light blue, light green and green) of the nodes.
- Blue and black network links that move over blue nodes.
- Different small sizes of the nodes.

TAU's NEW CALLGRAPH VISUALIZATIONS

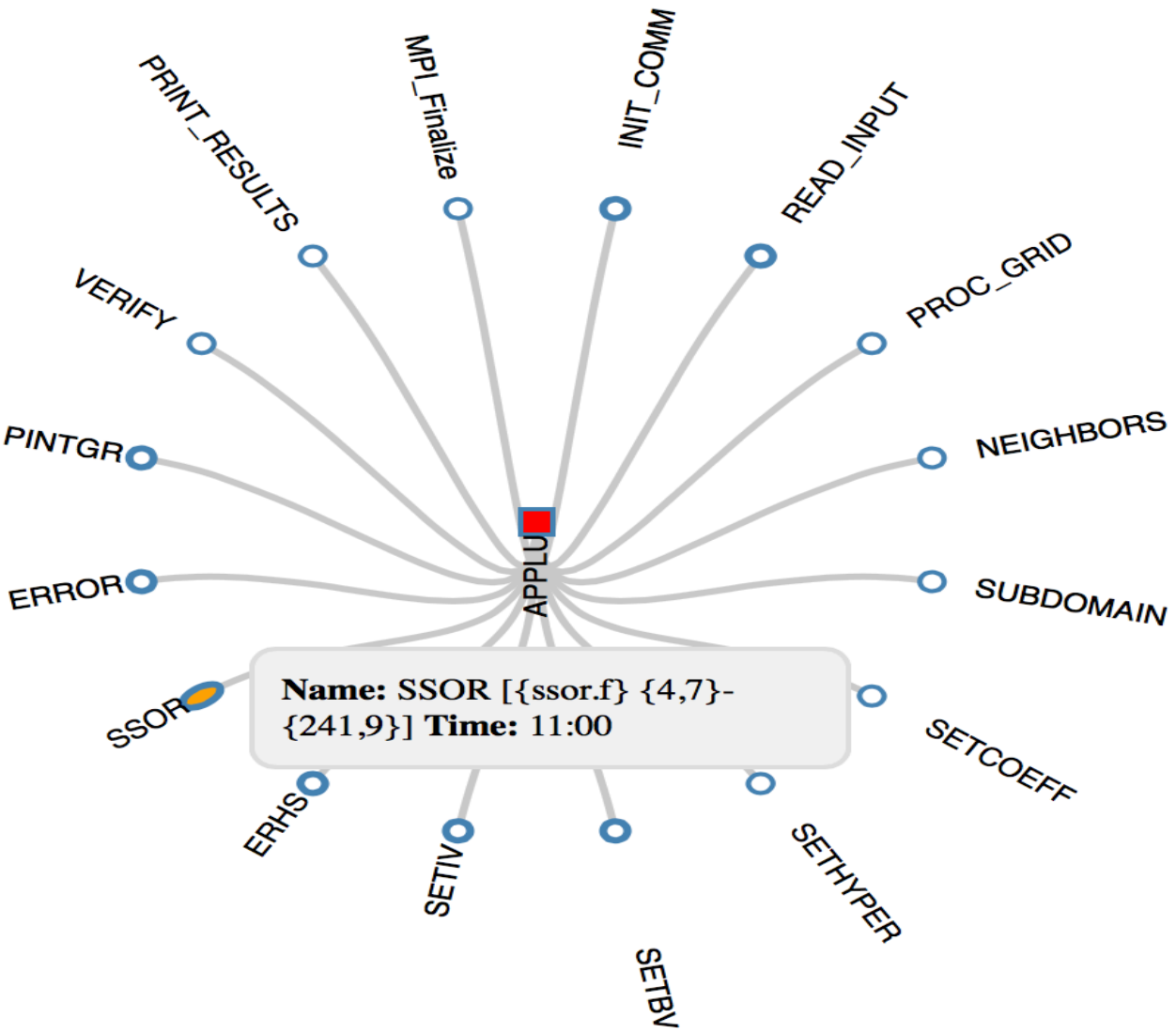
TAU'S NEW CALLGRAPH VISUALIZATIONS

D3.js JavaScript visualization library was used to design the following new interactive callgraph visualization prototypes:

- Star expandable design.
- Star collapse design.
- Tree expandable design.
- Tree none-expandable design.

STAR EXPANDABLE DESIGN

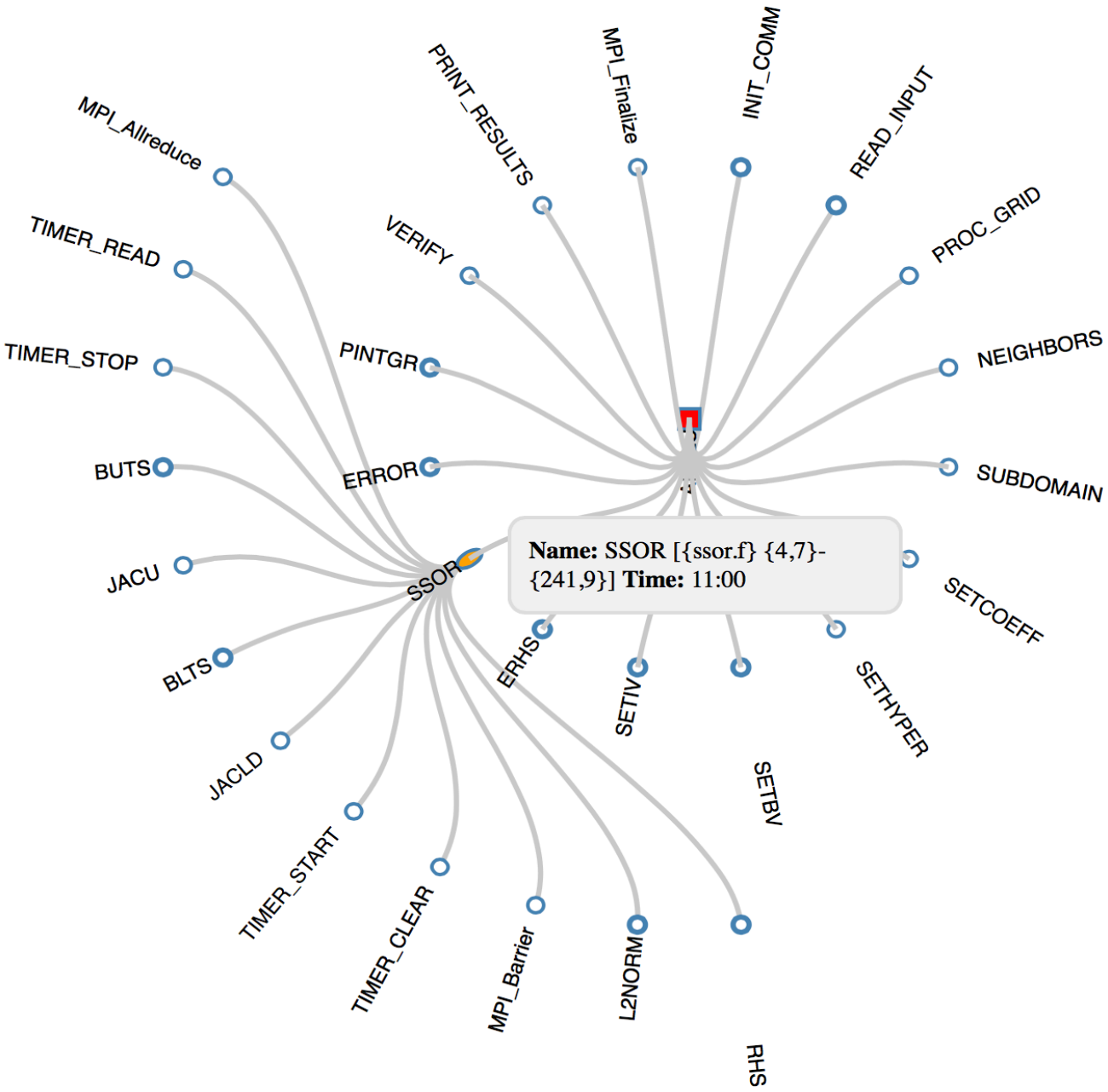
The below picture shows the first view of the star expandable design:



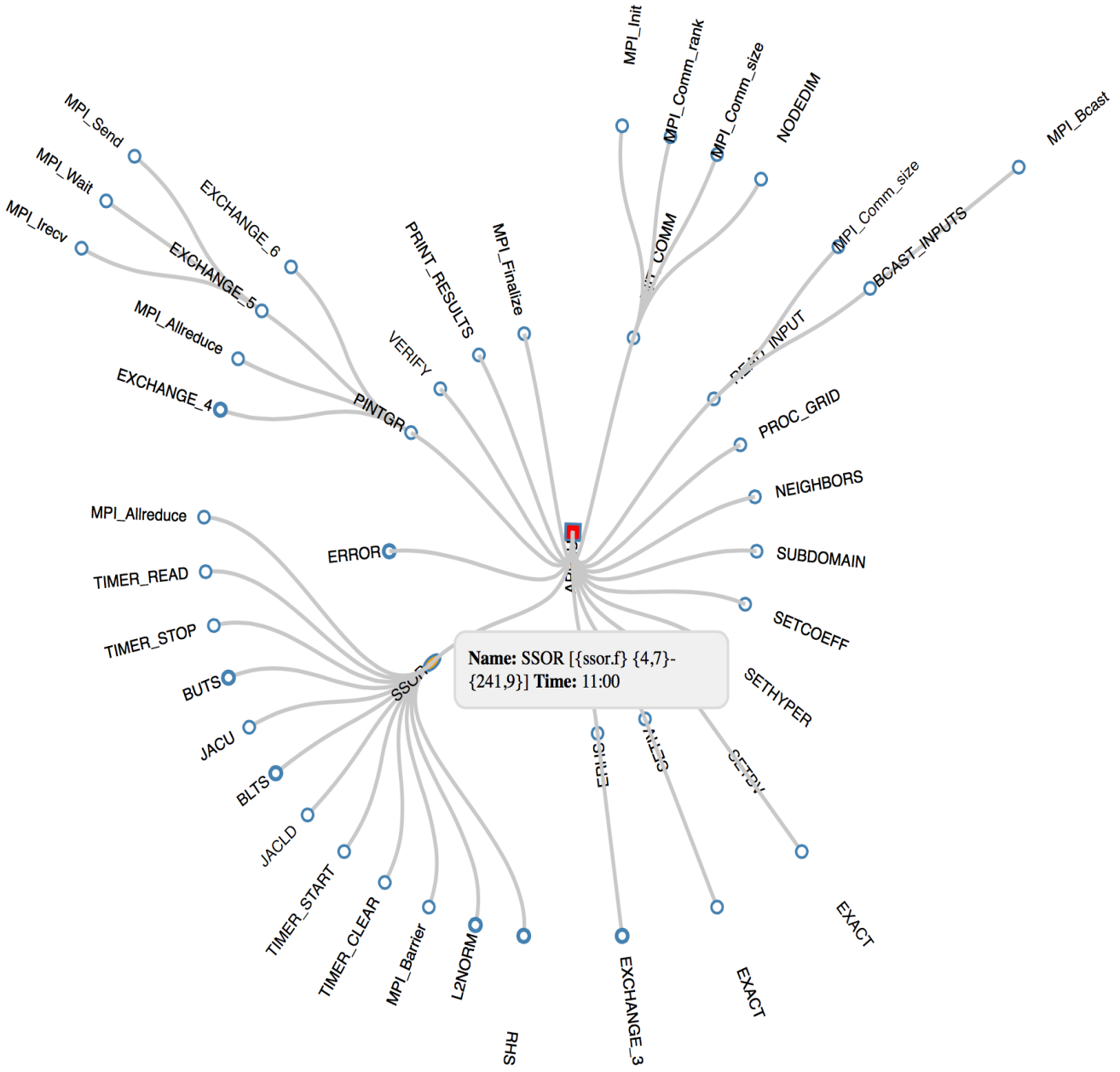
CHPC CENTRE FOR HIGH PERFORMANCE COMPUTING

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

STAR EXPANDABLE DESIGN (cont…)



STAR EXPANDABLE DESIGN (cont…)

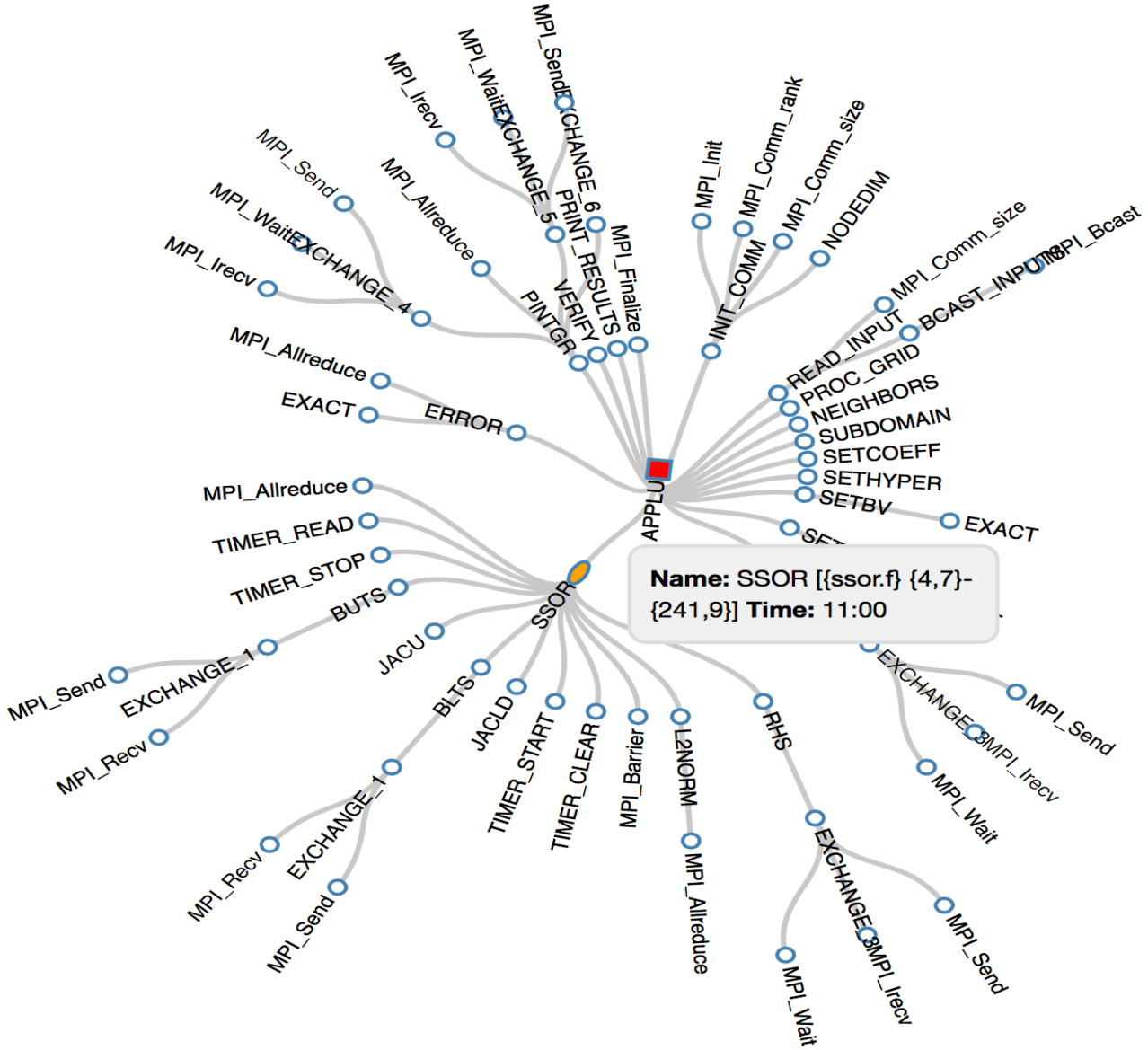


CHPC CENTRE FOR HIGH PERFORMANCE COMPUTING

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

STAR COLLAPSE DESIGN

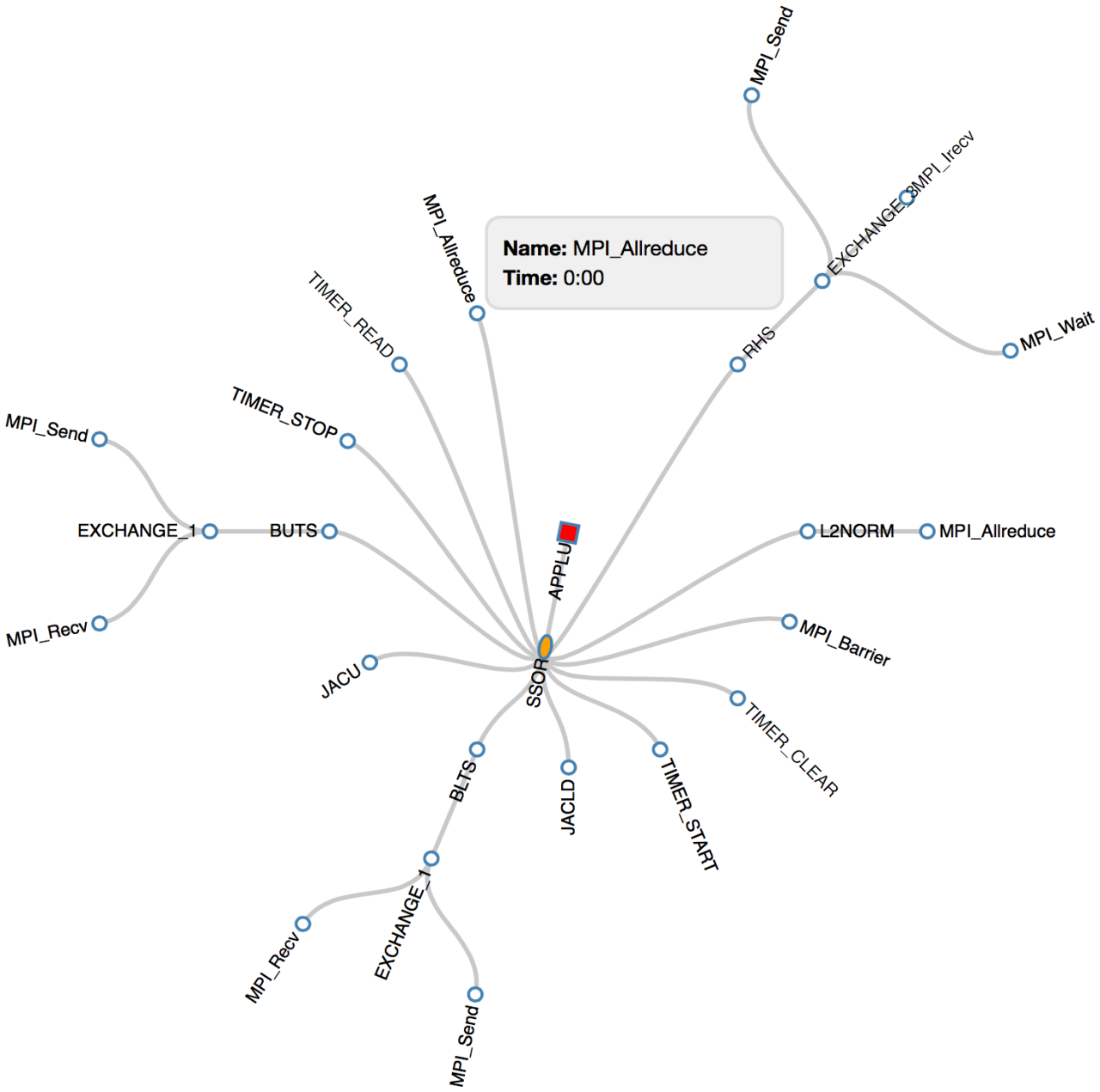
The below picture shows the first view of the star collapse design:



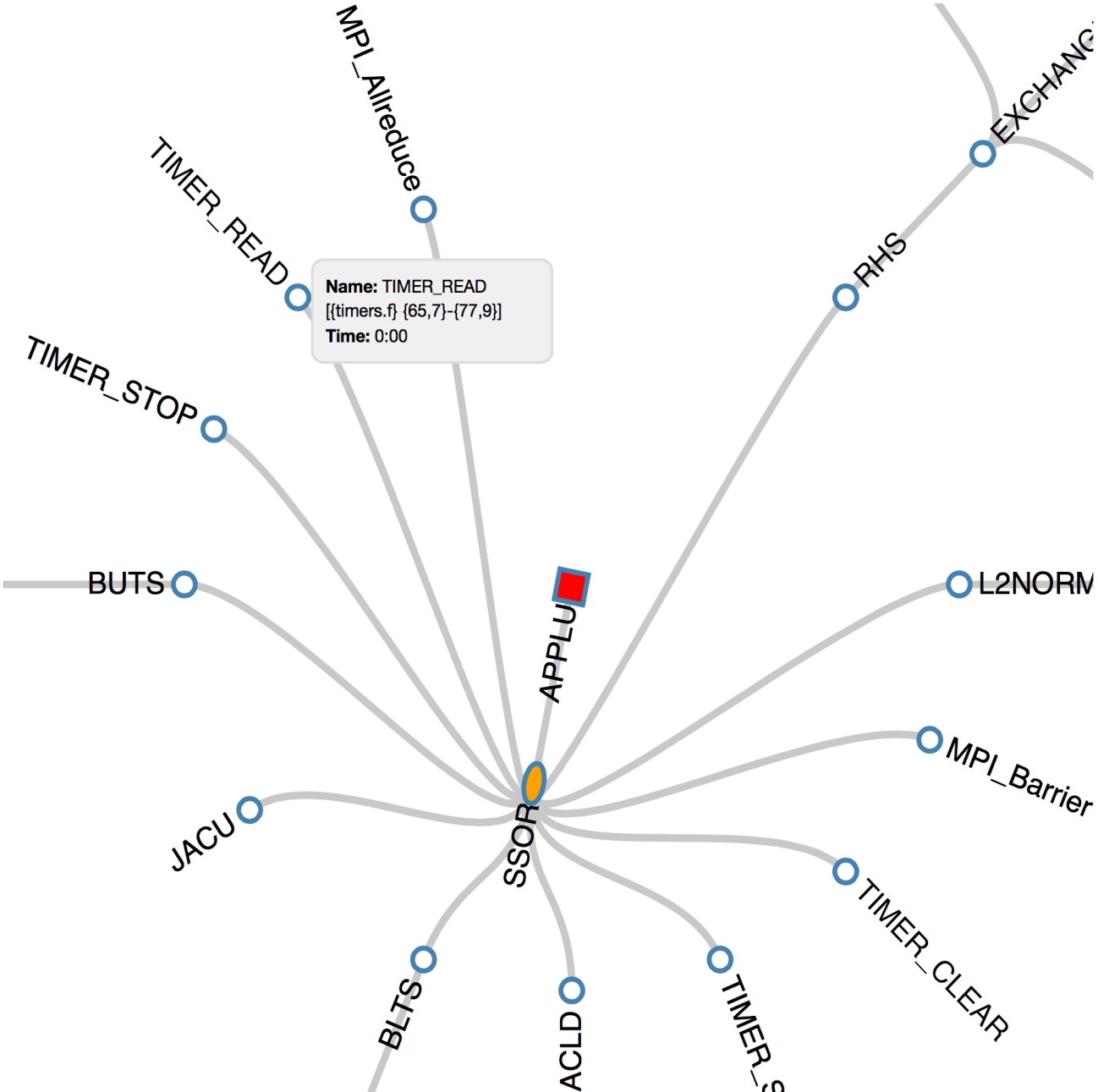
CHPC CENTRE FOR HIGH PERFORMANCE COMPUTING

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

STAR COLLAPSE DESIGN (cont…)



STAR COLLAPSE DESIGN (cont…)

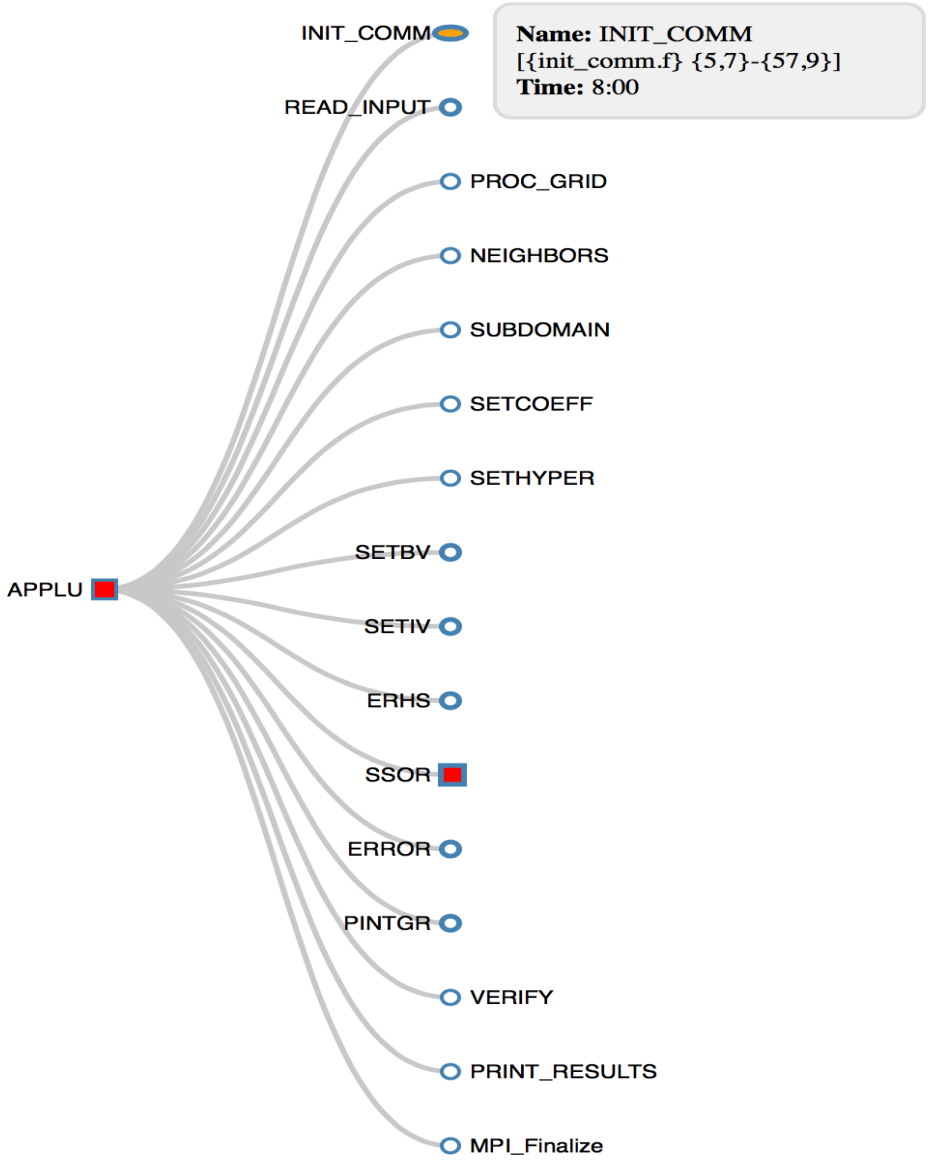


CENTRE FOR HIGH PERFORMANCE COMPUTING

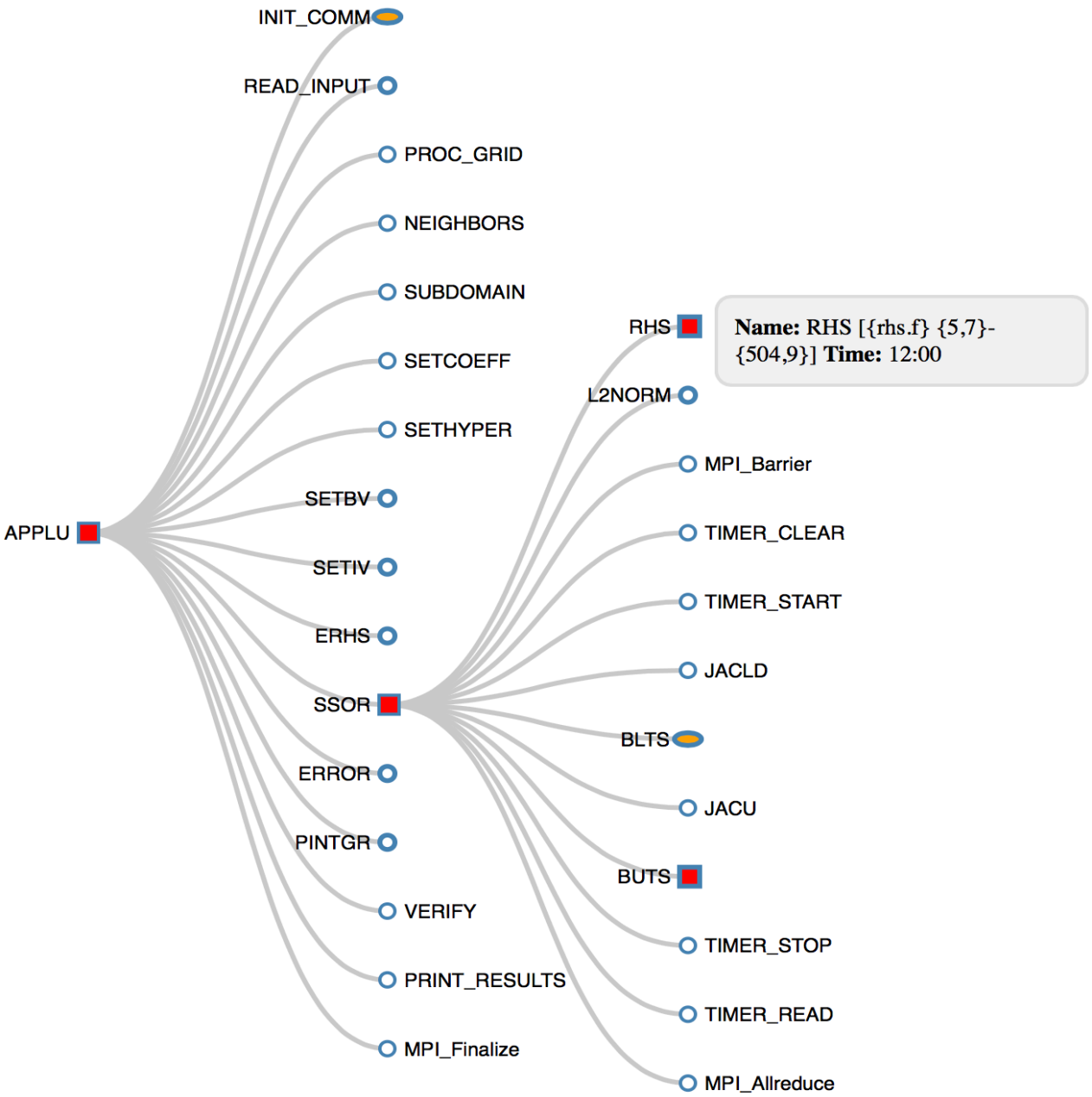
University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

TREE EXPANDABLE DESIGN

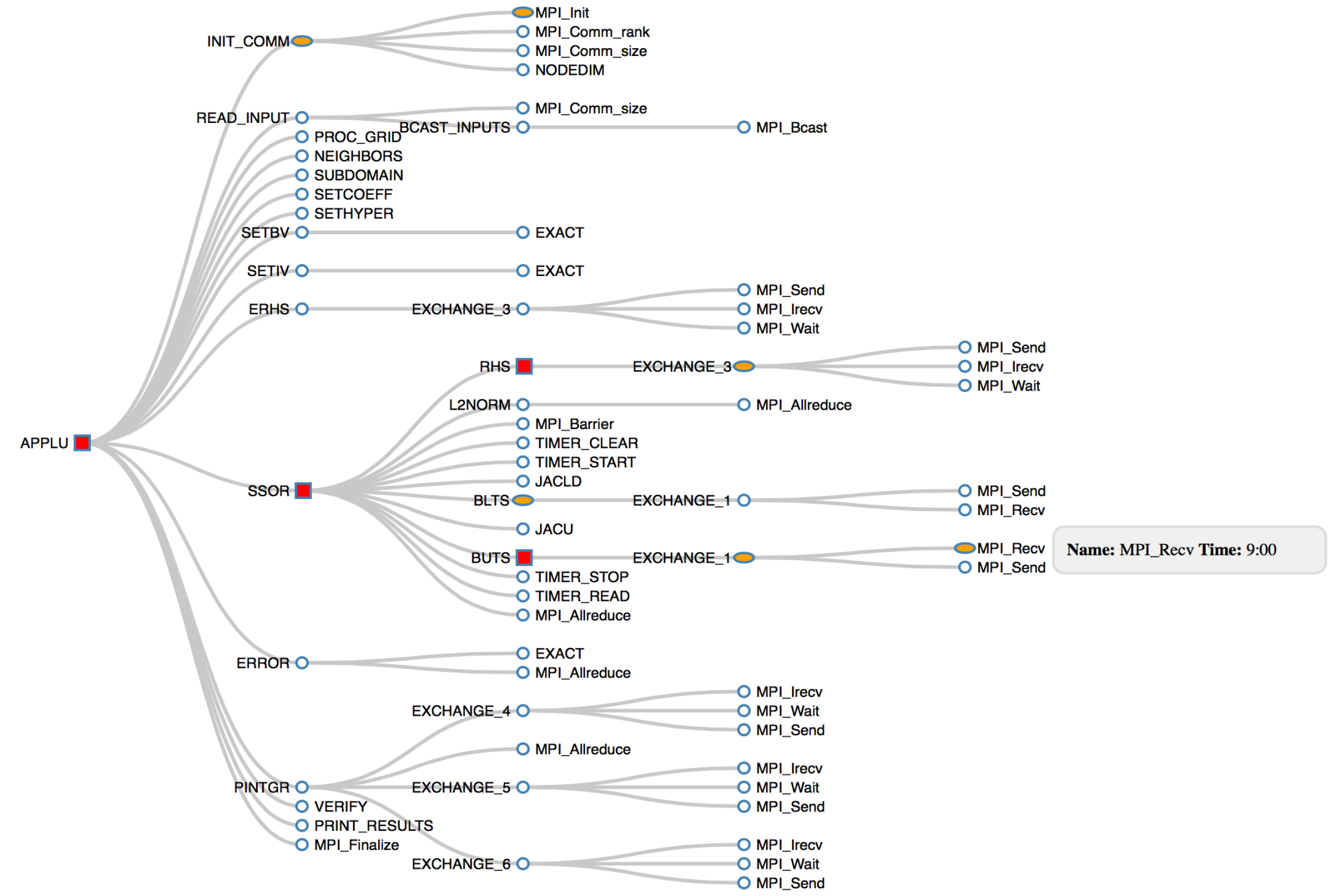
The following is the first view of the tree expandable design:



TREE EXPANDABLE DESIGN (cont…)

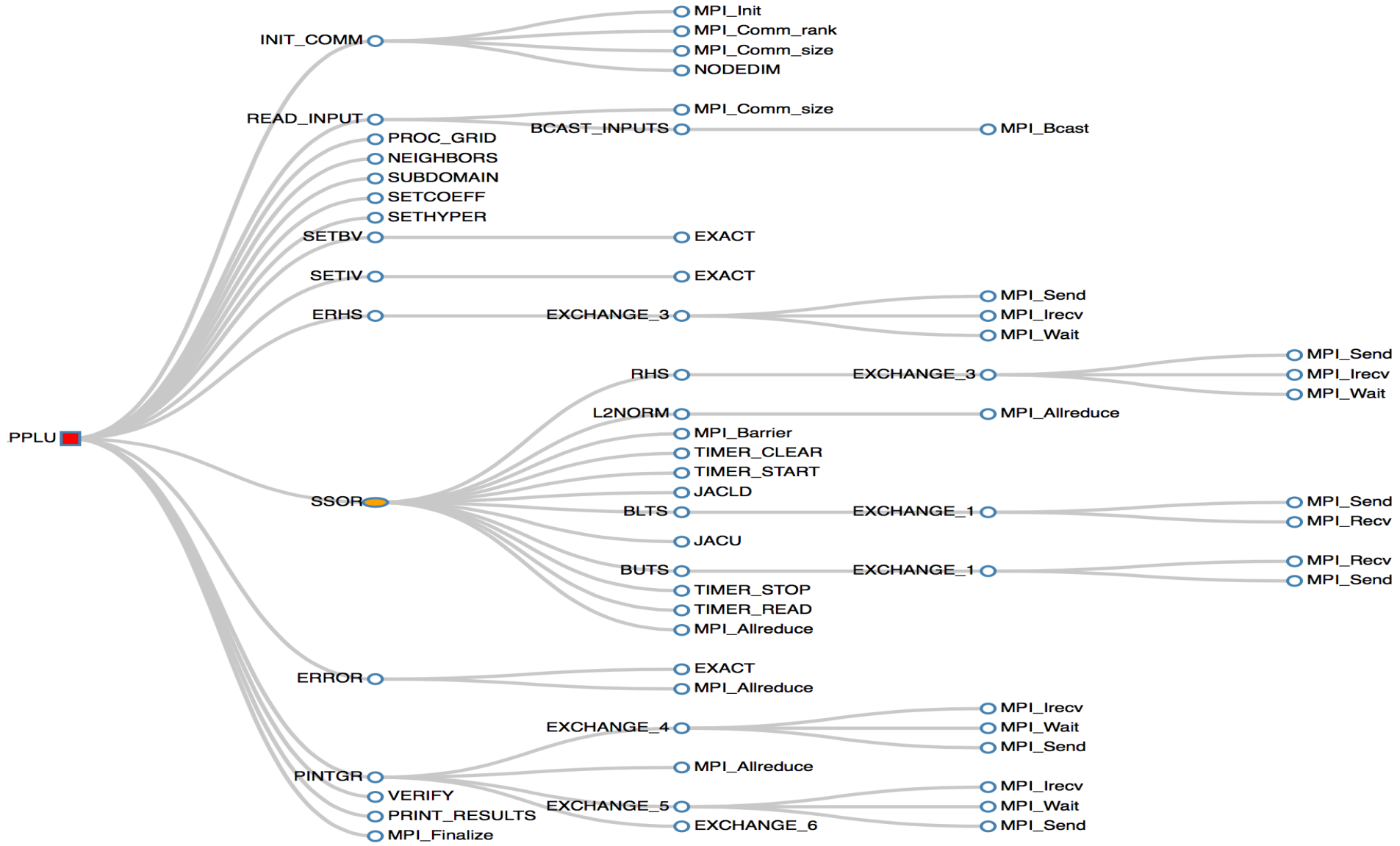


TREE EXPANDABLE DESIGN (cont…)

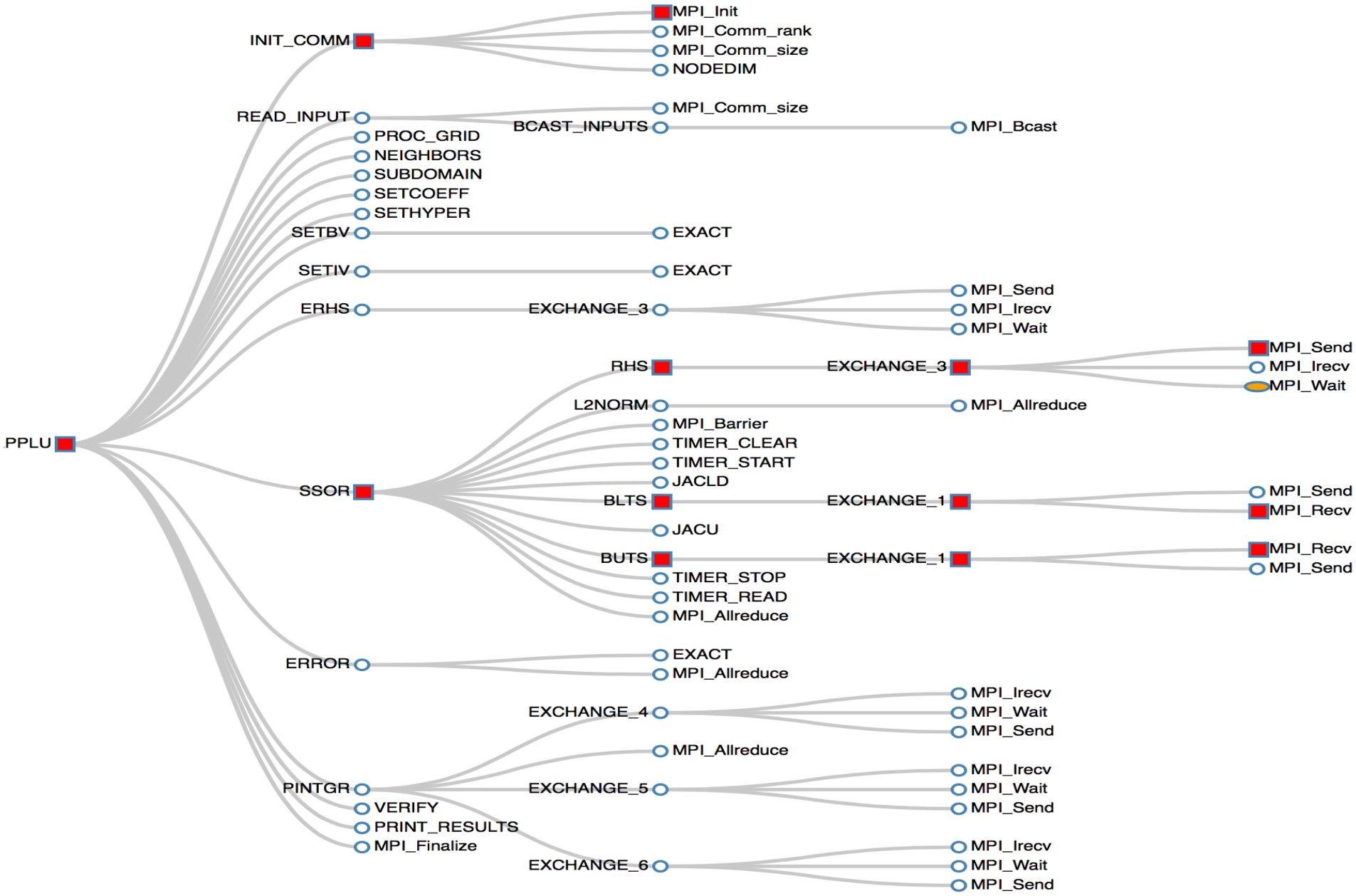


TREE NONE-EXPANDABLE DESIGN

The picture shows the tree none expandable design:



TREE NONE-EXPANDABLE DESIGN (cont…)



CONCLUSIONS

Parallel program users needs effective callgraphs to increase the performance of applications through optimization.

The current callgraphs are complicated to analyse the performance of parallel programs.

On the other hand, new designs of the callgraphs are useful for optimizing the performance of parallel applications simulated on the parallel system.

ACKNOWLEDGEMENTS

The authors acknowledges the following:

Council for Scientific and Industrial Research (CSIR) for funding the study.

Centre for High Performance Computing (CHPC) for providing computational resources to visualize parallel applications.

University of Oregon, Performance Research Lab for their assistance in providing data to tests the new callgraph visualizations.