

# DESIGN and IMPLEMENTATION of SOS: Scalable Observation System

---

## Observation, Introspection, and Awareness for Distributed Scientific Work Flows at Scale

Chad Wood  
Department of Computer Science University of Oregon Eugene, OR 97403-1202 USA  
and  
Daniel Ellsworth  
Department of Computer Science University of Oregon Eugene, OR 97403-1202 USA  
and  
Allen Malony

Department of Computer Science University of Oregon Eugene, OR 97403-1202 USA

The SOS project implements a distributed runtime tool-kit purposed with providing observation, introspection, and awareness across heterogeneous systems in exascale HPC environments. In addition to the development of software artifacts, this project contributes descriptions of the essential structures, patterns, and information flow required to facilitate effective dynamic system control, self-description, and data stores able to provide useful information from a variety of perspectives.

■

### Contents

Keywords:	1	<b>1 Introduction</b>	<b>3</b>
1.1	Purpose . . . . .		3
3			
1.2	Motivation . . . . .		3
3			
1.3	Design Map . . . . .		3
3	<b>2 Design Considerations</b>		<b>3</b>
2.1	Work Flows and Performance . . . . .		3
2.2	The View From Nowhere . . . . .		3
3			
2.3	Risks and Volatile Areas . . . . .		3
2.4	HPC Libraries as Input . . . . .		3
<b>3</b>	<b>Architecture</b>		<b>3</b>
3.1	Overview . . . . .		3
3	LIBRARY3 DATA STRUCTURES4 DAEMON4		
DATABASE4	<b>4 High Level Design</b>		<b>4</b>
4.1	Some Component Title . . . . .		4
4.2	Some OTHER Component Title . . . . .		4
<b>5</b>	<b>Low Level Design</b>		<b>4</b>
5.1	Some Implementation Detail . . . . .		4
5.2	Some OTHER Implementation Detail . . . . .		4
<b>6</b>	<b>Experiments and Results</b>		<b>4</b>
6.1	Some Experiment . . . . .		4
6.2	Some Other Experiment . . . . .		4
6.3	Recommendations . . . . .		4
<b>7</b>	<b>Conclusion</b>		<b>4</b>
7.1	Summary of Efforts . . . . .		4
7.2	Lessons Learned . . . . .		4
7.3	Planned Future Work . . . . .		4
4			

## 1. INTRODUCTION

### 1.1 Purpose

1.1.0.1 . Present day HPC environments impose varied and acute design constraints on the applications they host, limiting the ability of complex heterogeneous tasks to coordinate their action, introspection, and control with a common framework, spanning local and global scopes. Scientific work flows often are data-driven (dynamic) across their entire life cycle, and consequently it is unproductive to Engaging this challenging environment, the SOS project represents an effort to develop analytical models enunciating useful constraints and principles for coordinating scientific work flows. Along with basic research, practical software tools will be created for integration into existing HPC projects.

### 1.2 Motivation

1.2.0.2 . Some motivating notions.

### 1.3 Design Map

1.3.0.3 .

## 2. DESIGN CONSIDERATIONS

### 2.1 Work Flows and Performance

### 2.2 The View From Nowhere

2.2.0.4 . The 'central' point of view of the functioning of an exascale system is anywhere you decide to look.

### 2.3 Risks and Volatile Areas

### 2.4 HPC Libraries as Input

## 3. ARCHITECTURE

### 3.1 Overview

3.1.0.5 . SOS is implemented primarily in ANSI standard C as a library of functions, data structures, and a run-time daemon. SOS also features a centralized data store built with one of several SQL server solutions, or a write-only .CSV 'dump' for highly-constrained work flow execution environments that do not support heterogeneous application launches.

3.1.0.6 *LIBRARY.*

3.1.0.7 *DATA STRUCTURES.*

3.1.0.8 *DAEMON.*

3.1.0.9 *DATABASE.*

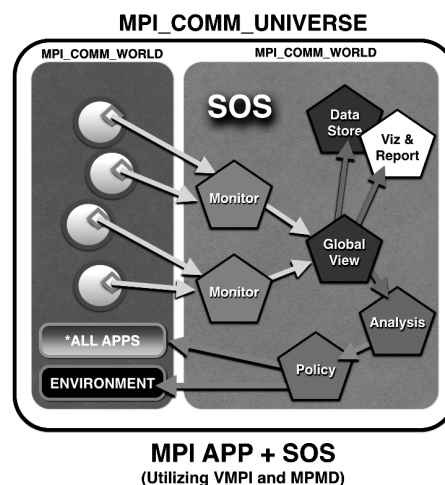


Fig. 1. An image as an example.

## 4. HIGH LEVEL DESIGN

### 4.1 Some Component Title

### 4.2 Some OTHER Component Title

## 5. LOW LEVEL DESIGN

### 5.1 Some Implementation Detail

### 5.2 Some OTHER Implementation Detail

## 6. EXPERIMENTS AND RESULTS

### 6.1 Some Experiment

### 6.2 Some Other Experiment

### 6.3 Recommendations

## 7. CONCLUSION

### 7.1 Summary of Efforts

### 7.2 Lessons Learned

### 7.3 Planned Future Work

7.3.0.10

—An example of a list.

—Another item in that list.

## REFERENCES

- K. A. Huck, A. D. Malony, S. Shende, and A. Morris. Taug: Runtime global performance data access using mpi. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pages 313–321. Springer, 2006.
- A. D. Malony and K. A. Huck. General hybrid parallel profiling. In *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, feb 2014.