

# SLURM: Simple Linux Utility for Resource Management<sup>\*</sup>

Andy B. Yoo, Morris A. Jette, and Mark Grondona

Lawrence Livermore National Laboratory  
Livermore, CA 94551  
{ayoo,jette1,grondona1}@llnl.gov

**Abstract.** A new cluster resource management system called Simple Linux Utility Resource Management (SLURM) is described in this paper. SLURM, initially developed for large Linux clusters at the Lawrence Livermore National Laboratory (LLNL), is a simple cluster manager that can scale to thousands of processors. SLURM is designed to be flexible and fault-tolerant and can be ported to other clusters of different size and architecture with minimal effort. We are certain that SLURM will benefit both users and system architects by providing them with a simple, robust, and highly scalable parallel job execution environment for their cluster system.

## 1 Introduction

Linux clusters, often constructed by using commodity off-the-shelf (COTS) components, have become increasingly popular as a computing platform for parallel computation in recent years, mainly due to their ability to deliver a high performance-cost ratio. Researchers have built and used small to medium size clusters for various applications [3, 16]. The continuous decrease in the price of the COTS parts in conjunction with the good scalability of the cluster architecture has now made it feasible to economically build large-scale clusters with thousands of processors [18, 19].

---

<sup>\*</sup> This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes. This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48. Document UCRL-JC-147996.

An essential component that is needed to harness such a computer is a resource management system. A resource management system (or resource manager) performs such crucial tasks as scheduling user jobs, monitoring machine and job status, launching user applications, and managing machine configuration. An ideal resource manager should be simple, efficient, scalable, fault-tolerant, and portable.

Unfortunately there are no open-source resource management systems currently available which satisfy these requirements. A survey [12] has revealed that many existing resource managers have poor scalability and fault-tolerance rendering them unsuitable for large clusters having thousands of processors [14, 11]. While some proprietary cluster managers are suitable for large clusters, they are typically designed for particular computer systems and/or interconnects [21, 14, 11]. Proprietary systems can also be expensive and unavailable in source-code form. Furthermore, proprietary cluster management functionality is usually provided as a part of a specific job scheduling system package. This mandates the use of the given scheduler just to manage a cluster, even though the scheduler does not necessarily meet the need of organization that hosts the cluster. Clear separation of the cluster management functionality from scheduling policy is desired.

This observation led us to set out to design a simple, highly scalable, and portable resource management system. The result of this effort is Simple Linux Utility Resource Management (SLURM<sup>1</sup>). SLURM was developed with the following design goals:

- *Simplicity*: SLURM is simple enough to allow motivated end-users to understand its source code and add functionality. The authors will avoid the temptation to add features unless they are of general appeal.
- *Open Source*: SLURM is available to everyone and will remain free. Its source code is distributed under the GNU General Public License [9].
- *Portability*: SLURM is written in the C language, with a GNU *autoconf* configuration engine. While initially written for Linux, other UNIX-like operating systems should be easy porting targets. SLURM also supports a general purpose *plug-in* mechanism, which permits a variety of different infrastructures to be easily supported. The SLURM configuration file specifies which set of plug-in modules should be used.
- *Interconnect independence*: SLURM supports UDP/IP based communication as well as the Quadrics Elan3 and Myrinet interconnects. Adding support for other interconnects is straightforward and utilizes the plug-in mechanism described above.
- *Scalability*: SLURM is designed for scalability to clusters of thousands of nodes. Jobs may specify their resource requirements in a variety of ways including requirements options and ranges, potentially permitting faster initiation than otherwise possible.

---

<sup>1</sup> A tip of the hat to Matt Groening and creators of *Futurama*, where Slurm is the most popular carbonated beverage in the universe.