LPJS - Lightweight, Portable Job Scheduler

Contents

Chapter 1

Introduction

LPJS is a resource manager and job scheduler for running batch jobs on one or more computers. It can be used on a single machine in order to maximize utilization of CPUs and memory without oversubscribing the system, or on multiple networked computers organized as an HPC (high performance computing) cluster or HTC (high throughput computing) grid.

Unlike most similar tools, LPJS is designed to be small, simple, easy to install and configure, and easy to use. It provides an intuitive user interface, including menu-based operation for common tasks.

This manual is aimed at the systems manager, covering installation and configuration of LPJS. For a user's guide on LPJS use, including general information on HPC clusters and HTC grids, see the Research Computing User's Guide at https://acadix.biz/-publications.php.

Chapter 2

Installation

Installation should be performed using a package manager, such as FreeBSD ports or pkgsrc. We maintain a FreeBSD port for use on FreeBSD and Dragonfly BSD, and a pkgsrc package that should work on almost any other POSIX platform, including other BSDs, most Linux distributions, macOS, Solaris-based systems, etc. Other package managers may be supported by third parties. If you would like to add LPJS to your favorite package manager, see the instructions for packagers in the README at https://github.com/outpaddling/LPJS/.

LPJS uses **munge** (https://github.com/dun/munge) to encrypt and authenticate messages between nodes. Munge is installed automatically by the package manager when installing LPJS.

However, munge must be configured and enabled as a service as well. Munge requires all nodes to have a shared munge key file, which is unique to your installation. It must be generated by you (see the munge documentation) and distributed to all computers that are part of your cluster or grid. THE MUNGE KEY FILE MUST BE KEPT SECURE AT ALL TIMES ON ALL NODES. Use secure procedures to distribute it to all nodes, so that it is never visible to unauthorized users.

Chapter 3

Configuration

LPJS is designed to require minimal configuration. For example, compute node specs such as available cores and memory are determined automatically, and need not be specified in configuration files.

Most configuration can be done entirely using **lpjs-admin**, a menu-driven admin tool. Simply run **lpjs-admin**, select an item from the menu, and answer the questions on the screen.

For the sake of understanding what **lpjs-admin** does, some basic information is provided below.

The head node requires a configuration file, which in its simplest form merely lists the complete host names (FQDNs) of the head node and each compute node, e.g.

```
head    myhead.mydomain
compute compute001.mydomain
compute compute002.mydomain
...
```

The FQDN (fully qualified domain name) must match the name reported by the **hostname** command, or the `gethostname()` standard library function.

Each compute node requires a similar configuration file, but it need only list the head node. It can be the same configuration file used on the head node, in which case the compute node entries are ignored.

Chapter 4

Starting Daemons

All nodes in the cluster or grid must be running **munged**, using the same munge key. The head node must run **lpjs_dispatchd**, and all compute nodes must run **lpjs_compd**.

Appropriate services can be configured by running **lpjs-admin** on each node.

Note The head node can also serve as a compute node, though this is not generally recommended. The head node should remain lightly loaded so that it can respond in real time to events that occur such as new job submissions and job completions. The head node need not be a powerful machine. A laptop or low-end desktop machine will work just fine.

Jobs can be submitted from any node with the same version of LPJS and the shared munge key installed. It need not be running LPJS daemons, but it does require a configuration file point to the head node and munge to authenticate requests. Hence, other computers on the network can act as submit nodes, even if they are not part of the cluster/grid.

The lpjs_dispatchd and lpjs_compd commands are normally run as a service. You can run lpjs-admin as the root user and use the menus to configure a machine as a head node or compute node with the appropriate services enabled.

It is also possible to use LPJS without having admin access. Simply start the daemons manually by running **munged** and **lpjs_dispatchd** on the head node, and **munged** and **lpjs_compd** on each compute node. Note that if lpjs_compd is not running as root, the compute node will only be able to run jobs under the same user name that submitted them.

Chapter 5

Advanced configuration

TBD