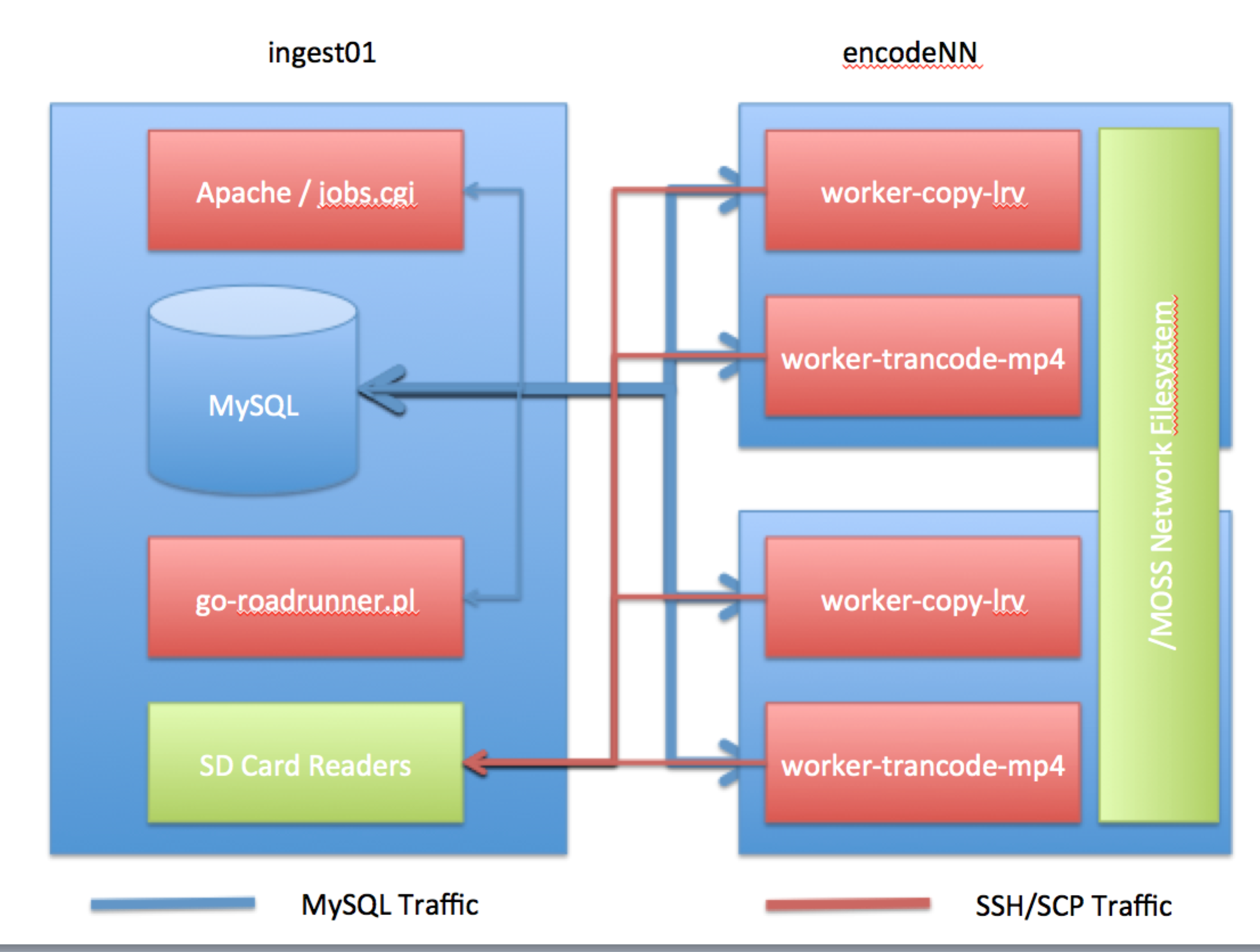
**Fleye Roadrunner Operations Manual**

Roadrunner is a video file ingestion, transcoding, and indexing system built upon open source software including Linux, MySQL, Apache, Perl, and FFmpeg. The system is designed to ingest video files from GoPro HD Hero 2 camera, index them, and prepare them for remote transcoding and storage somewhere across the Internet. The goal of this system is to streamline the process of processing SD cards from a Fleye event in order for video files to be ready to be used on the Fleye website.

**System Design**

Roadrunner is split across three types of server instances:

* Admin: The admin server (note, singlular) run a central MySQL server which provides job coordination amongst ingest and encode machines. Each component requires access to the MySQL server. Additionally, the admin server runs a web application designed to provide status reports on work in progress.
* Ingest: These servers are where SD cards are physically attached. One ingest server can support approximately 6 encode servers. Ingest servers run the application “go-roadrunner.pl”
* Encode: These servers copy files from the ingest servers, transcode them using FFmpeg, and place them on network storage. Currently, the system is designed to drop files on Cari.Net’s MOSS file storage product. Encode servers run two applications: “worker-copy-lrv.pl” and “worker-transcode-mp4.pl”
* Note: Currently, admin and ingest functions are shared on the same common machine (ingest01, 216.177.0.43). Additional ingest machines can be setup to communicate with the cluster but additional ingest machines do not require the admin functions to be setup. All ingest and encode instances should be configured to communicate with MySQL on ingest01.



**System Setup**

* Steps for admin node:
  + Install linux, mysql, apache, and git.
  + Fetch the roadrunner code from GitHub: “git clone git://github.com/tomdalynh/fleye fleye”
  + Install code locally: “cd fleye/roadrunner && sudo make prereq install”
  + Secure MySQL permissions, create “roadrunner” user and password, create “fleye\_roadrunner” database, import “fleye/roadrunner/sql/schema.sql” schema into database.
  + Configure apache with “fleye/roadrunner/www/web-roadrunner.conf” in /etc/http/conf.d.
* Steps for ingest node:
  + Install linux, fuse-exfat.
  + Copy code via SCP from admin node.
  + Create local user “system”
  + Install code locally: “cd fleye/roadrunner && sudo make prereq install”
* Steps for encode node:
  + Install linux and setup MOSS.
  + Copy code via SCP from admin node.
  + Create local user “system”. Setup SSH key pair for user. Append public key (id\_rsa.pub) to each ingest node’s /home/system/.ssh/authorized\_keys file.
  + Install code locally: “cd fleye/roadrunner && sudo make prereq install”
* Optional: Install ganglia across nodes for system level monitoring.

**Processing Cards / Process Workflow**

To process cards from a Fleye event, the following workflow is used:

* One or more encode instances should be launched in the cari.net cloud. For each encode instance started:
  + Mount the MOSS network filesystem via NFS to /MOSS. For configured machines, typing “sudo mount –a” should be sufficient.
  + Deploy the roadrunner application to the machine to ensure the most recent version is running:
    - SCP current version from ingest01:/home/tom/fleye to a working directory.
    - Install using make: “cd fleye && sudo make all install clean”
  + Start the worker-copy-lrv process: “sudo –u system /usr/local/fleye/roadrunner/bin/worker-copy-lrv.pl –d”. The “-d” flag will daemonize the process. Omitting the flag will run it in the foreground. The process will connect to admin MySQL to poll for work to do.
  + Start the worker-transcode-mp4 process: “sudo –u system /usr/local/fleye/roadrunner/bin/worker-transcode-mp4.pl –d”. The “-d” flag will daemonize the process. Omitting the flag will run it in the foreground. The process will connect to admin MySQL to poll for work to do.
* Now, connect to ingest01 to launch the process.
  + Ensure that MySQL is running.
  + Attach SD cards to be processed to ingest01.
  + Ensure that the go-roadrunner job queue is empty: “sudo /usr/local/fleye/roadrunner/bin/go-roadrunner.pl –t”. “The “-t” flag tells the system to truncate the current running job queue.
  + Use go-roadrunner to mount and process cards: “sudo /usr/local/fleye/roadrunner/bin/go-roadrunner.pl –m”. The “-m” flag tells the system to search and mount SD cards, and to process the contents of the card. The process will locate files, and index them into MySQL for processing. Remote workers will connect via SCP to copy files.
* Now, monitor the progress.
  + Ensure that Apache is running on ingest01.
  + Navigate to <http://216.177.0.43/roadrunner/cgi-bin/jobs.cgi> to monitor job processing.
  + Navigate to <http://216.177.0.43/ganglia2/> to monitor system load and network throughput.
* When the job queue is finished, unmounts the cards from ingest01: “sudo /usr/local/fleye/roadrunner/bin/go-roadrunner.pl –u”. The “-u” flag tells the system to unmount cards from the system.

**Program Information:**

There are three critical programs to the Roadrunner system: go-roadrunner.pl, worker-copy-lrv.pl, and worker-transcode-mp4.pl. Each program requires a common set of database credentials stored in “fleye/roadrunner/etc/db.conf”.

**go-roadrunner.pl**

go-roadrunner.pl is responsible for managing the content of SD cards, and making them ready for coping. go-roadrunner.pl will design the file hierarchy on the remote encode side for all files found. By default, the program with simply query the database for running jobs and display the number of jobs to be run.

This program needs to run as “root” on the ingest server due to needing permissions to mount cards with fuse-exfat. This program requires access to the MySQL server specified in db.conf.

Switches below cause additional behavior:

|  |  |
| --- | --- |
| Switch Name / Short Version |  |
| --db\_config\_file / -c | Specify an alternate database config file, defaults to '/usr/local/fleye/roadrunner/etc/db.conf' |
| --local\_path\_glob / -l | Used for testing. Specify a pattern to glob local paths from and process their contents. Useful for processing from a local file system directory rather than an SD card that needs mounting. E.X. '/local-' |
| --mount\_cards / -m | Mount cards and look for files on attached card readers. Index and queue for processing all .LRV files and .MP4 files. |
| --unmount\_cards / -u | Umount all cards. Checks to make sure no jobs are running. |
| --truncate\_jobs / -t | Truncates the jobs table; stops all processing once current remote copy\_lrv and transcode\_mp4 jobs are running. Checks to make sure no jobs are in queue before running. |
| --force / -f | Combine with -u or -t; forcefully unmount or truncate. |
| --status\_report / -s | Prints a quick status report. |
| --help / -h | Prints help. |

**worker-copy-lrv.pl**

worker-copy-lrv.pl is responsible for processing LRV files found on the SD cards. The process polls the jobs table in MySQL for “copy\_lrv” jobs, and upon locating one, will copy the LRV file from the ingest machine to the encode’s /MOSS directory, appropriately renaming the file as needed. Failure to copy a file will result in it being placed in the back of the job queue.

This program needs to run as “system” on the encode server due to needing SSH keys from the encode server to the ingest server. This program requires that /MOSS be mounted. This program requires access to the MySQL server specified in db.conf.

The program will run in the foreground by default. Switches below cause additional behavior:

|  |  |
| --- | --- |
| Switch Name / Short Version |  |
| --db\_config\_file / -c | Specify an alternate database config file, defaults to '/usr/local/fleye/roadrunner/etc/db.conf' |
| -- daemonize / -d | Run as a daemon, fork to background. Logging will be placed in /home/system/worker-copy-lrv. |
| --kill / -k | Find any running daemon and exit. |
| --help / -h | Prints help. |

**worker-transcode-mp4.pl**

worker-transcode-mp4.pl is responsible for processing MP4 files found on the SD cards. The process polls the jobs table in MySQL for “copy\_lrv” jobs, and upon locating one, will copy the MP4 file from the ingest machine to the encode’s /MOSS directory, appropriately renaming the file as needed. It will then launch asynchronously FFmpeg to transcode the MP4 file from 1080p to 360p. Failure to copy a file or transcode it will result in it being placed in the back of the job queue.

This program needs to run as “system” on the encode server due to needing SSH keys from the encode server to the ingest server. This program requires that /MOSS be mounted. This program requires access to the MySQL server specified in db.conf.

The program will run in the foreground by default. Switches below cause additional behavior:

|  |  |
| --- | --- |
| Switch Name / Short Version |  |
| --db\_config\_file / -c | Specify an alternate database config file, defaults to '/usr/local/fleye/roadrunner/etc/db.conf' |
| -- daemonize / -d | Run as a daemon, fork to background. Logging will be placed in /home/system/worker-transcode-mp4. |
| --kill / -k | Find any running daemon and exit. |
| --help / -h | Prints help. |