Getcams System Overview -- 2018

# Background and Goals

This document describes the getcams system and its role among the PRP (Pacific Research Platform, prp.ucsd.edu) infrastructure benefits to HPWREN. Included are requirement for porting getcams.

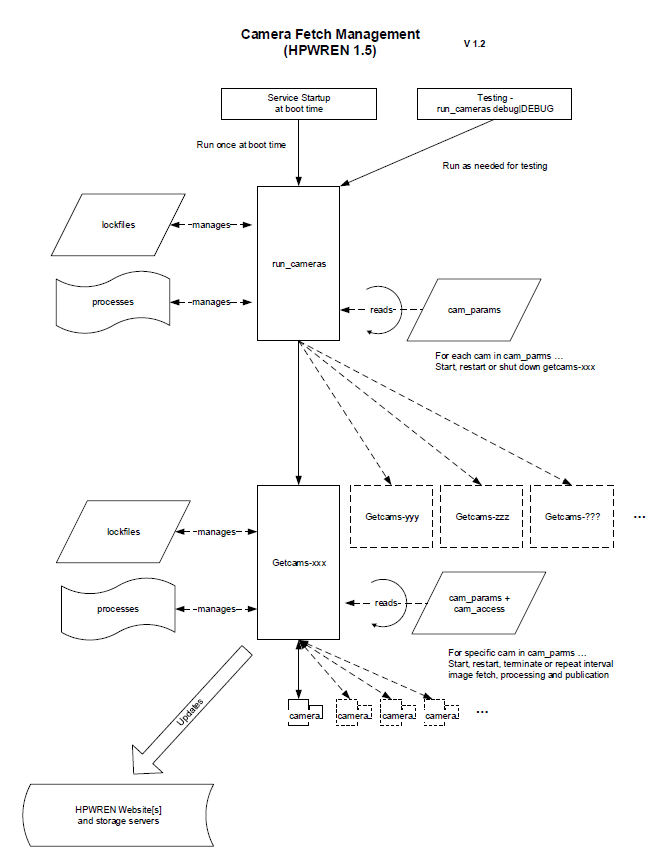
Getcams is a new system for the fetching, processing, publishing, and archival of HPWREN (hpwren.ucsd.edu) camera image data. It is one part of HPWRENs engagement with the PRP project. As a PRP applications driver and use case, HPWREN has received PRP servers and services in support of making HPWREN data more available, easier to access, and more user friendly, redundant and scalable. To date, this support includes 7 (“Fiona”) servers and close to 750TB of storage at UCSD, SDSU and UCI, and is now growing by 500TB to over a PB (Petabyte) total storage. The added storage is being installed across regions of CA (adding Los Angeles, Santa Cruz and Merced). PRP staff are also supporting the HPWREN adoption of container based applications and CEPH distributed storage.

Originally, the fetching, processing, publishing, and archival of HPWREN (hpwren.ucsd.edu) camera image data were performed by a collection of FreeBSD based “archive” servers each of which handled the above services for a subset of HPWREN cameras. These half-dozen or so legacy servers also provided web services and access to sensor and weather data. Over time, longer term archival storage on these machines became impractical due to storage limitations. PRP Fiona servers are now providing HPWREN archival storage as well as camera image animations processing (of the real-time images sent over by the legacy servers). These newer production systems extended the life and utility of the legacy servers and provided multiple backup web and storage servers in the event the HPWREN servers became unavailable. Getcams was designed to ultimately take over the remaining camera image fetching and processing roles of the legacy servers. It could now do so (if run on c1 or c2) if any of the legacy servers failed. It currently runs experimentally on c0 using PRP provided CEPH storage. In this mode, it is limited to fetching and processing about 30 camera images per minute before being limited by current CEPH storage throughput. Storage upgrades are now in progress to remove that bottleneck.

# Design

Getcams is a system consisting of a single “run\_cameras” daemon (a bash script), started at boot time, which reads a camera control file and spawns individual camera drivers (“getcams-\*” perl scripts) accordingly. The control file tells the system which cameras to fetch from, how frequently to fetch images, and any camera or vendor specific information, such as which getcams driver to invoke or which parameters of choice to use for the fetch. The daemon monitors the fetching, kills or restarts camera drivers as needed, and handles system logging and process management, rereading the control file regularly for changes. Camera drivers are currently available for Iqeye, Mobotix, and Axis cameras (getcams-iqeye.pl, getcams-mobo.pl and getcams-axis.pl).

The getcams system process logic is illustrated on the image below.



# Implementation

The getcams system began development under CentOS 7 on Fiona servers (c1 and c2). Development continued on the Centos 7 VM c0 (camaqc1) and is now near completion and operable on c0.

The latest stable version of getcams is available from <https://github.com/ghidley/getcams>

The getcams system consists primarily of the following executables and control files:

run\_cameras: daemon that starts and maintains running system, reading cam\_params control file and starting getcams-\*.pl as needed

getcams-\*.pl: camera drivers, one each for Iqueye (getcams-iqeye.pl) , Mobotix (getcams-mobo.pl) and Axis cameras (getcams-axis.pl)

cam\_params: camera control file for indicating camera fetch parameters

cam\_access: camera login and password information (not in git repository)

The above files install to ~hpwren/bin/getcams, via the getcams Makefile.

Getcams fetches images from cameras enabled in cam\_params and stores images in location specified within run\_cameras:

Storage areas used by getcams system (from Makefile):

ARCHDIR=/Data/archive #(for long term archival)

CDIR=$(ARCHDIR)/incoming/cameras #(for last fetched images)

DATADIR=/Data

INCOMING=$(ARCHDIR)/incoming/cameras/tmp

RUNDIR=~hpwren/bin/getcams

LOCALDIR=/Data-local/scratch

SYSLOCAL=/var/local/hpwren

LOCKDIR=$(SYSLOCAL)/lock

LOGDIR=$(SYSLOCAL)/log

Storage set locally win getcams-\* files:

$ADIR="/Data/archive"; # Archival image location  
$TDIR="/Data-local/scratch"; # Temp/local faster location for interim processing  
$CDIR= "$ADIR/incoming/cameras"; # Current image location (for web page collage)

$TDIR location requires up to 40MB per camera imager (currently we have about 180 imagers from which we fetch images … thus using about 8GB temp processing space

# Requirements for porting

Suitable server or container with the following attributes:

Need routing to access # 172.16.0.0/16 HPWREN-local private network

Need storage access to camera image archival area (using about 240TB currently)

Need /etc/hosts file (or DNS services) with camera base names and IP addresses

Need user and group hpwren:

ghidley@c1 ~$ grep hpwren /etc/passwd /etc/group\*

/etc/passwd:hpwren:x:30001:30001:HPWREN role account:/home/hpwren:/bin/bash

/etc/group:hpwren:x:30001:hwb,ghidley,geoff,davis,jmeyer,lirichards,abrust

Need the following software packages:

perl (with , Log::Log4perl and [File::Copy](File:///\\:Copy) qw(copy) libs)

python, python-swiftclient

swift, s3cmd

ppmlabel, ppmarith, convert, pnmscale (from ImageMagick/netpbm ports)