

Session Archaeology

Introduction

Around 2013, we started running a nightly cron script to crawl nanoHUB user home directories and collect run files from closed tool sessions. The original idea was to archive these run files for later post processing and analysis.

The cron method that we employed was not perfect. For instance, if a user purged their expired session folders, or run files in sessions that predated the cron harvester – would both result in “holes” in our collection. Over the last 5+ years, we have collected hundreds of thousands of run files, all compressed and stored in a mounted file system accessible by my (Nathan Denny) nanoHUB user account (schcats).

The nightly cron harvest job was obviated after the “ionhelper” updates that went live on 2019-01-06. For sessions after this date, runfiles are automatically deposited into the Instant On archive and can be directly accessed via a RESTful URL. Furthermore, the association of runfiles to sessions can be easily determined via the narwhal.squidlog table, which is updated in real-time.

The focus of this page, will be on those sessions that predate the ionhelper changes. We will call those sessions “archaic”, as compared to sessions after ionhelper which we will label “contemporary.” Of course, the study of archaic objects is “archaeology”, hence the title: “Session Archaeology”

Data Structure

Archaic sessions are compiled into compact sqlite3 database files. Each tool/revision pair (e.g. pntoy_r55) is compiled into its own compact file. The design of the compact files is to be a self-contained collection that includes run file content, post processing information like SQuIDs and file hashes, as well as session context, and compilation metadata. Most questions regarding a tool session should be able to be answered solely from within the compact file.

```
“CREATE TABLE IF NOT EXISTS promises (sessnum INTEGER PRIMARY KEY, username TEXT, appname TEXT, queued TEXT, promised TEXT, concluded TEXT, agent TEXT, status INTEGER, results BLOB)”
```

```
“CREATE TABLE IF NOT EXISTS runfiles (sessnum INTEGER, UUID TEXT, name TEXT, osize INTEGER, zsize INTEGER, mtime REAL, format TEXT, sha1 TEXT, squid TEXT, content BLOB, PRIMARY KEY (sessnum, name))”,
```

```
“CREATE TABLE IF NOT EXISTS sessions (sessnum INTEGER PRIMARY KEY, start REAL, username TEXT, remoteip TEXT, appname TEXT, walltime REAL, cputime REAL)”
```

Data Repository

Currently, the compact files are stored as read-only files hosted at db2.nanohub.org in the folder `/var/lib/arklite`

The compact files are readable by anyone in the “users” group on db2 and can be processed on the host (db2) itself, or downloaded and processed separately via `scp`, `sftp`, etc.