MCWrapper and the OSG

Thomas Britton
HallD Postdoc

The use case

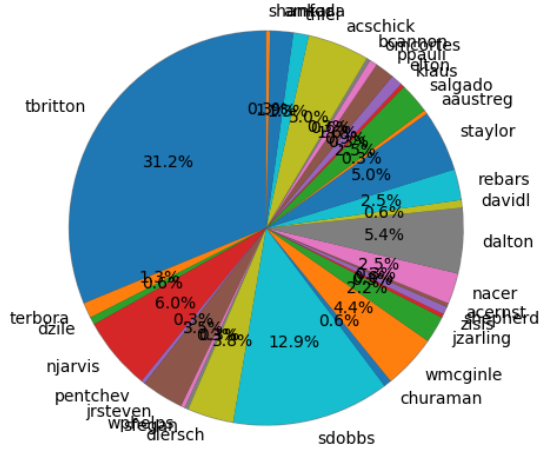
- The lowest hanging fruit is in Monte Carlo production
  - Very parallelizeable (each simulated event is independent)
- Had to fight for local resources
  - Higher priority given to data taking and reconstruction
- Desire to run buffered from localized issues
- Given everything above OSG seems an obvious target

Paved road

- Using the OSG opportunistically has freed HallD/GlueX local cycles for data analysis
  - Utilization of offsite resources is part of the plan to fulfill the computational needs of GlueX et al
- Users of MCwrapper/MCwrapper-bot accept some constraints on running (covers most use cases anyways) in exchange for not having to handle the production
  - Most users don’t care where any of it happens. Just that it does happen….

Statistics

- Since July 20 2018
  - 32 unique users (out of 116 members)
  - Produced over 47 TB of MC
  - In more than 700k core-hours



The anatomy of a submit file

Declare universe

what executes

arguments for exe

node requirements

container loc/behavior

job behaviors

needed locations

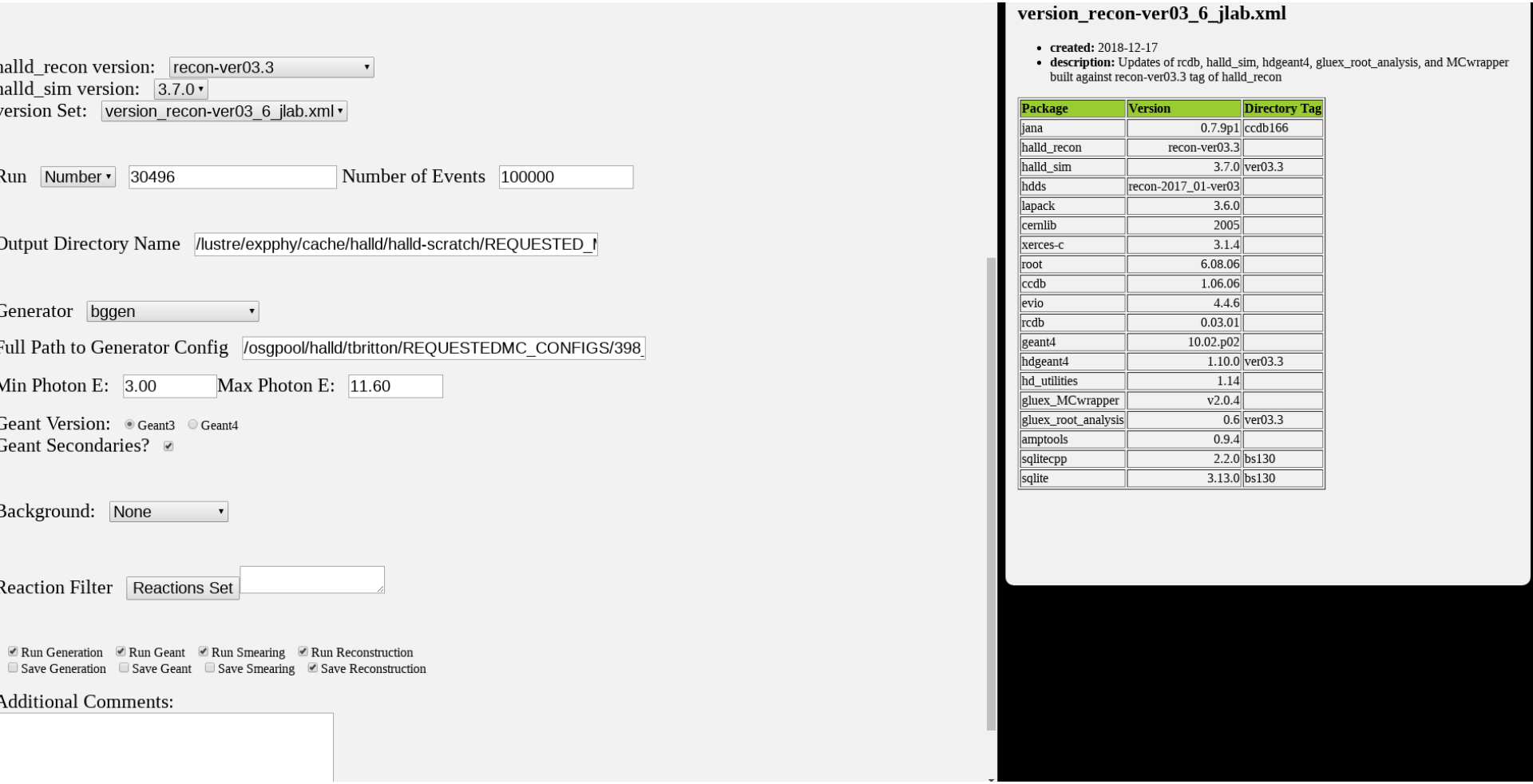
output

queue it

```
universe = vanilla
Executable = /osgpool/halld/tbritton/gluex_MCwrapper//osg-container.sh
Arguments  = ./MakeMC.sh 1 /srv/version_recon-ver03_6_jlab.xml /srv/gen_omega_rad
one file:/srv/403_jana.config 50000 ./ batch_default batch_default 0 rcdb rcdb re
Requirements = (HAS_SINGULARITY == TRUE) && (HAS_CVMFS_oasis_opensciencegrid_org
+SingularityImage = "/cvmfs/singularity.opensciencegrid.org/markito3/gluex_docker
+SingularityBindCVMFS = True
+SingularityAutoLoad = True
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
concurrency_limits = GluexProduction
on_exit_remove = true
on_exit_hold = false
Error      = /osgpool/halld/tbritton/REQUESTEDMC_OUTPUT/omega_rad_gend_2gamma_ana
output      = /osgpool/halld/tbritton/REQUESTEDMC_OUTPUT/omega_rad_gend_2gamma_an
log = /osgpool/halld/tbritton/REQUESTEDMC_OUTPUT/omega_rad_gend_2gamma_analyzed_2
initialdir = ./
transfer_input_files = /osgpool/halld/tbritton/gluex_MCwrapper//MakeMC.sh, /group
alld_sim-4.1.0/src/programs/Simulation/gen_omega_radiative/gen_omega_radiative_f1
jana.config
transfer_output_files = 30848_3
transfer_output_remaps = "30848_3=/osgpool/halld/tbritton/REQUESTEDMC_OUTPUT/omeg
queue
```

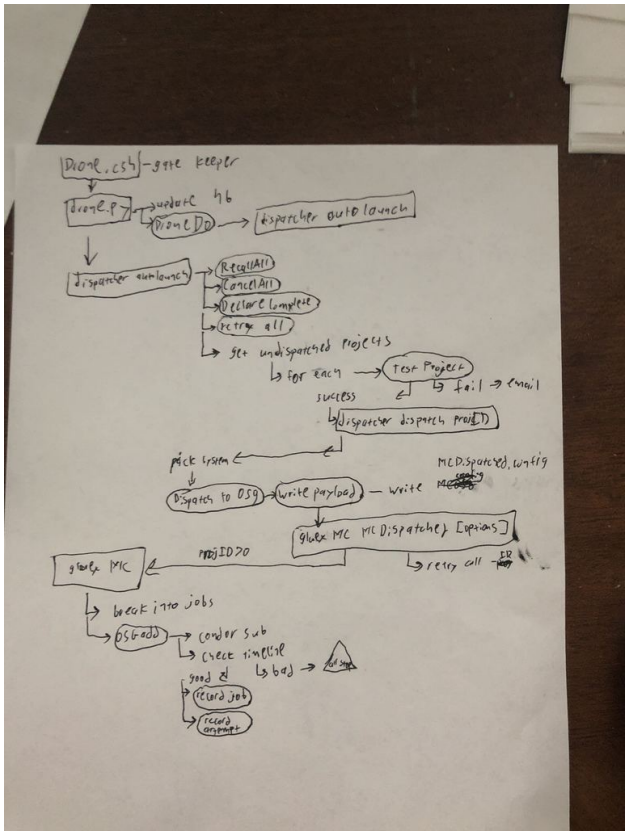
BACKUP

What the user sees

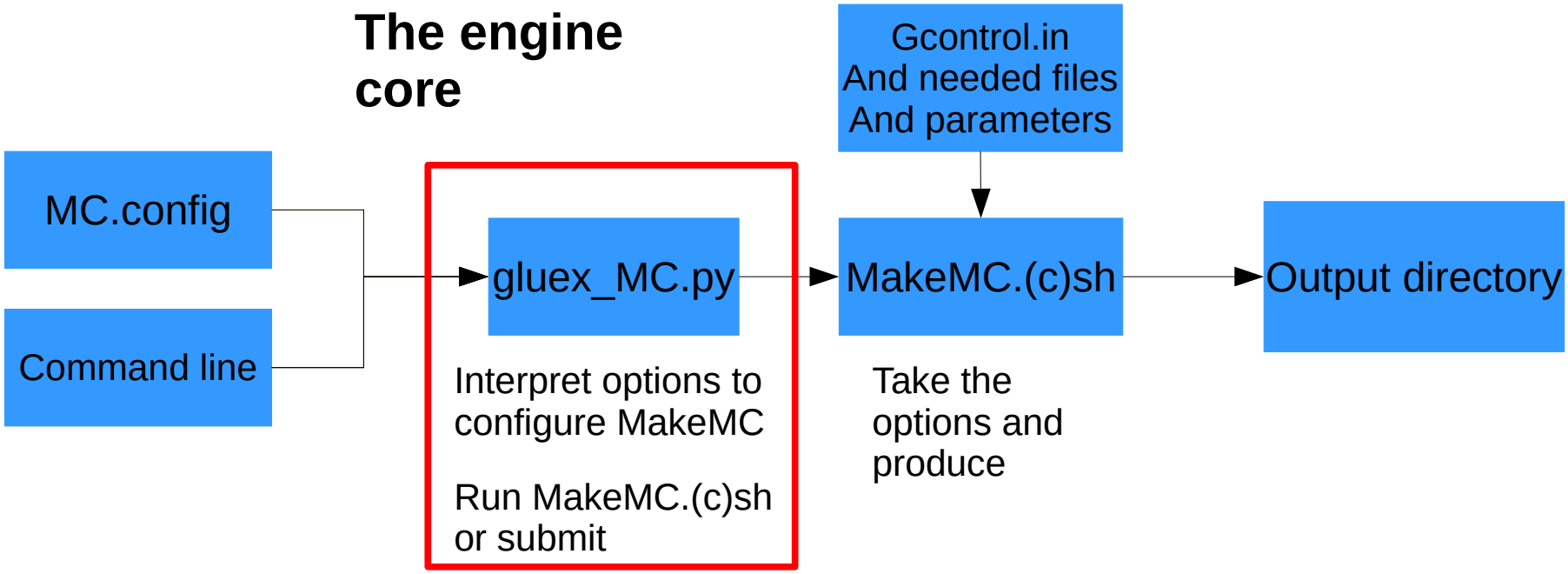


What really happens

- Once submitted a Rube Goldberg springs into action
  - Many moving pieces
    - Html/css/JS
    - Php/MySQL
    - Python
    - Bash/cshell
- Requests are tested
  - In the environment they should run
    - Not fool proof yet
- Requests are broken into jobs
  - Empirically optimized for OSG
- Jobs are submitted to the OSG



What really happens



- gluex_MC is responsible for interpreting the user and “instructing” MakeMC to Make MC. This includes configuring submit files

The Future

- Fix the potholes
  - Increase throughput and utilization of the OSG
- Prototype reconstruction/analysis on the OSG
- If opportunistic running is the name of the game. Opportunity discovery will be key