OSG Engagement VO

Helping new users

Mats Rynge

Renaissance Computing Institute

Engagement Mission

1. Facilitate University Campus CI deployment, and interconnect it with the national organizations

2. Help new user communities from diverse scientific domains adapt their computational systems to leverage OSG

User Profile

- Small research groups
- Little or no CS/IT resources
- No knowledge / time / resources / need for a complex job handling system
- Example: Kuhlman lab at UNC
  - Protein folding
  - 1 job run (~5000 jobs, ~ 3 days)
  - 4 weeks in wet lab, using results from run

Engagements – Initial interactions with new users

- User describes executable, needed inputs and example on how to run the model
- Every user is different, but in general, Engagement team creates:
  - submit tool (creates jobs / dags)
  - job-wrapper (wraps model remotely)
  - job-success-check (checks stdout)

Biggest Challenges?

- Used to be security
- Big step to go from 1 job to 1000s
  - job / data management
- The black box (aka remote resource)
  - small differences hard to track down
  - environment configuration
  - file system configuration
  - available system utilities
  - network setup

Engagement VO Jobs

- Mostly very simple jobs
  - Embarrassingly parallel
  - No inter-job dependencies
  - Simple staging requirements
    - Inputs/outputs staged with job
- Job independence makes it easy to spread a run across many sites
- Great candidates for match making

What is Resource Selection?

- Well described jobs and resources
- **Automatically** match the jobs up against resources
- Additional features include
  - automatic retries of failed jobs
  - site verification

Job Requirements

- Can you list all the requirements for your jobs?
  - Memory usage?
  - Disk usage?
  - Dependencies?
- Most users have a hard time describing their own code

Additional Job Requirements for Resource Selection

- Job fails...
- Job is in the queue for too long...
- Job is running for too long...

$\left.\begin{array}{l}\\ \\ \\ \end{array}\right\}$ resubmit to another site

- When submitting to another site, do not submit to a site which we have already failed on

ReSS

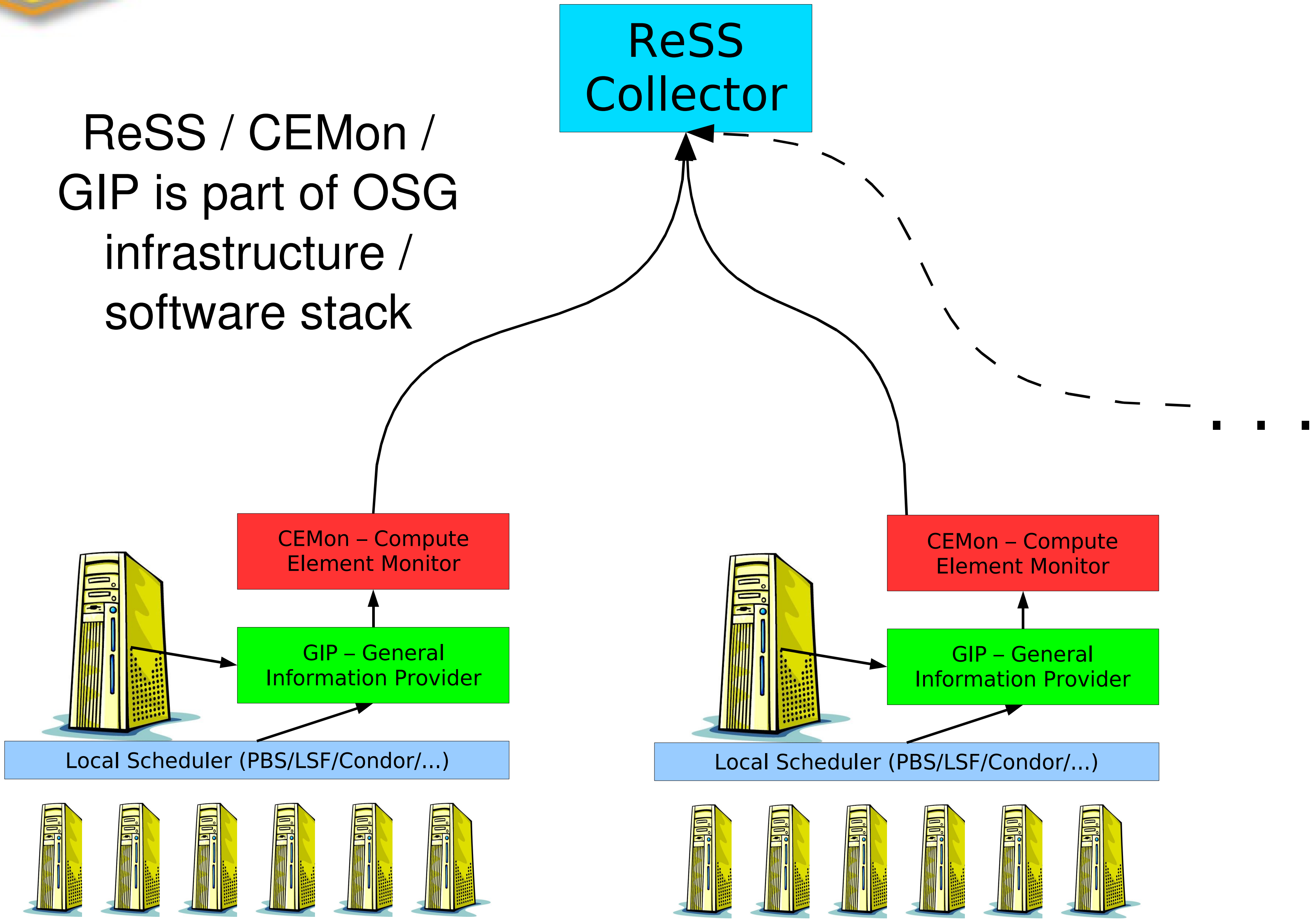
- Resource Selection Service
  – but is only really an information provider
- Developed at Fermi Lab and is part of OSG infrastructure
- Collects data from compute elements (clusters) and publishes the data in Condor ClassAd format

Information in ReSS

- OS name / version
- LRM information
  - Total number of job slots
  - Assigned slots
  - Open job slots
- Memory / CPU / Disk
- Network setup
- Storage configuration

Open Science Grid

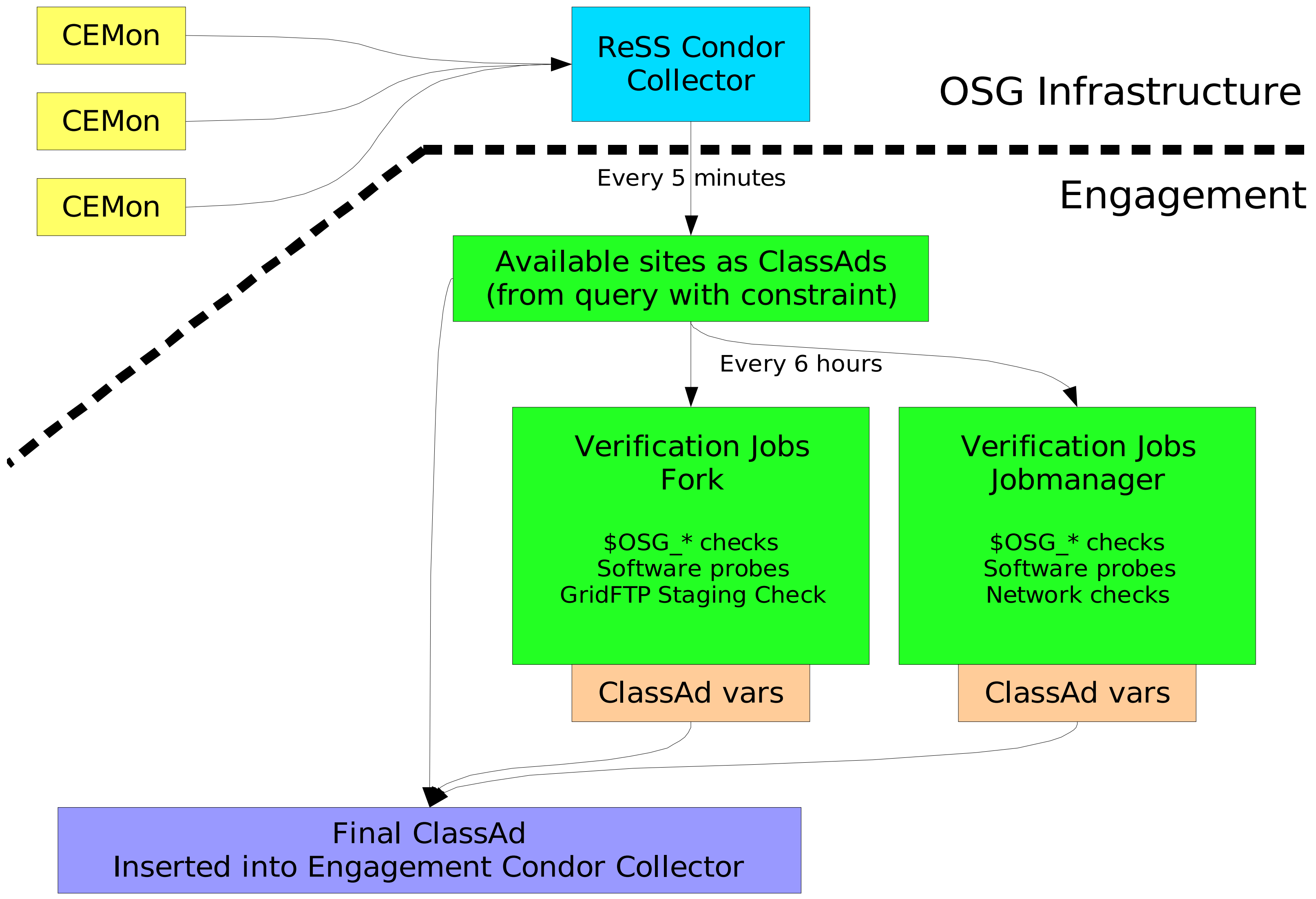
ReSS / CEMon / GIP is part of OSG infrastructure / software stack


ReSS Collector
. . .
CEMon – Compute Element Monitor
GIP – General Information Provider
Local Scheduler (PBS/LSF/Condor/...)
CEMon – Compute Element Monitor
GIP – General Information Provider
Local Scheduler (PBS/LSF/Condor/...)

Engagement ReSS Layer

- Retrieve base ClassAds from ReSS
- Validate the sites with probe jobs
- Determine the current state of the system by looking at current job states and success rates (continuous system feedback)
- Merge the information, and insert into local Condor system

Open Science Grid



Verification: File Systems

- $OSG_WN_TMP
  - most of the time local, similar to /tmp

    Exist? Write permissions?
- $OSG_DATA
  - shared, read/write from worknodes

    Is it mounted? Permissions? Does it have the data we staged earlier?
- $OSG_APP
  - shared, read-only from worknodes

    Is it mounted? Permissions? Does it have the applications we expect?

ReSS + Site Verification

- ReSS information + site verification:
  - – The result is a set of resource classads
    - Well formed
    - Verified information
  - – Users' can trust the information
  - – Increases job success rates

More Information

- ReSS (Resource Selection Service): https://twiki.grid.iu.edu/twiki/bin/view/ResourceSelection/WebHome
- OSG Engagement VO https://twiki.grid.iu.edu/twiki/bin/view/Engagement/WebHome
- Questions?
  - Email: osg@renci.org