Compute Provisioning ideas

Andrew McNab

University of Manchester

GridPP, DUNE, LHCb

Compute Provisioning ideas

- Some **preliminary** ideas about this area:
  - Scope / definitions
  - Timescales
  - Medium timescale
    - Request descriptions
  - Short timescale
    - Generic pilot factories?
  - Other opportunities

“Compute Provisioning” ???

- I’m using “Compute Provisioning” to mean obtaining compute resources on behalf of an experiment
- So this is **not** fabric / infrastructure provisioning
  - Not something like Foreman or OpenStack
- It’s also **not** workload management systems
  - Not PanDA, DIRAC WMS, GlideinWMS, …
- But it **does** include systems which sit between the sites and the WMS
  - Harvester, HEPCloud, Vcycle, CloudScheduler, …

Timescales: short/medium/long

- Useful to think about three timescales:
  - **Short**: make a reservation now that lasts ~days
    - “local job slot” / pilot job / pilot VM
  - **Medium**: making a request for capacity in the future (~months away) on ~existing hardware
    - ?JSON? files describing future plans?
  - **Long**: making a request which may require purchasing new hardware
    - C-RRB requests and pledges

Medium timescale provisioning

- Currently we have a programmatic way of accessing downtimes (via GOCDB)
- But we don't have a programmatic way for experiments to publish their compute requirements for the next ~year
  - We have talked about it on and off for a while
- How would we do this? Some kind of standard JSON file?
- Would need to describe "job slot" parameters
  - Processors, memory, duration, …?

Medium timescale benefits

- Can automatically warn of conflicts between planned downtimes and requests
- Sites can plan provisioning during the year
  - Experiments typically have plenty of jobs to run outside the high priority processing
  - Danger now is that share would get used up on low priority jobs - so sites tend to maintain roughly constant allocations during the year
- Emails/talks explaining plans are ok for ~4 experiments: won't scale to 20 with SKA, LSST, …
- Of course, **not going to be anywhere near perfect**, but better than guessing / assuming ~flat consumption

“Slots” in request descriptions

- To specify what is being requested we will need a general way of describing Job Slots / VM Slots
  - Processors, memory, duration, (GPUs, …) …
- Lists / ranges of what is acceptable?
  - Some things may need to be ratios: GB/processor
- Similar language to parts of the Compute Resource Records?
- Use JSON?

Short timescales

- "Short timescale" is where we already have working systems
  - The "pilot job" and its variations are ways of requesting short term reservations
  - We also have this for cloud (a VM is kind of reservation)
  - And HPC backfill jobs are jobs too, and for HPC, reservations are their normal way of working

Short timescales

- Some systems we have already
  - Glidein factories / Harvester / Vcycle / DIRAC SiteDirectors / HEPCloud / CloudScheduler
- All different though and tied to particular workload systems in many cases
- More than one per experiment also!
  - Even though we are “encouraged” to have common tools
- Can we stop proliferation as more projects join?

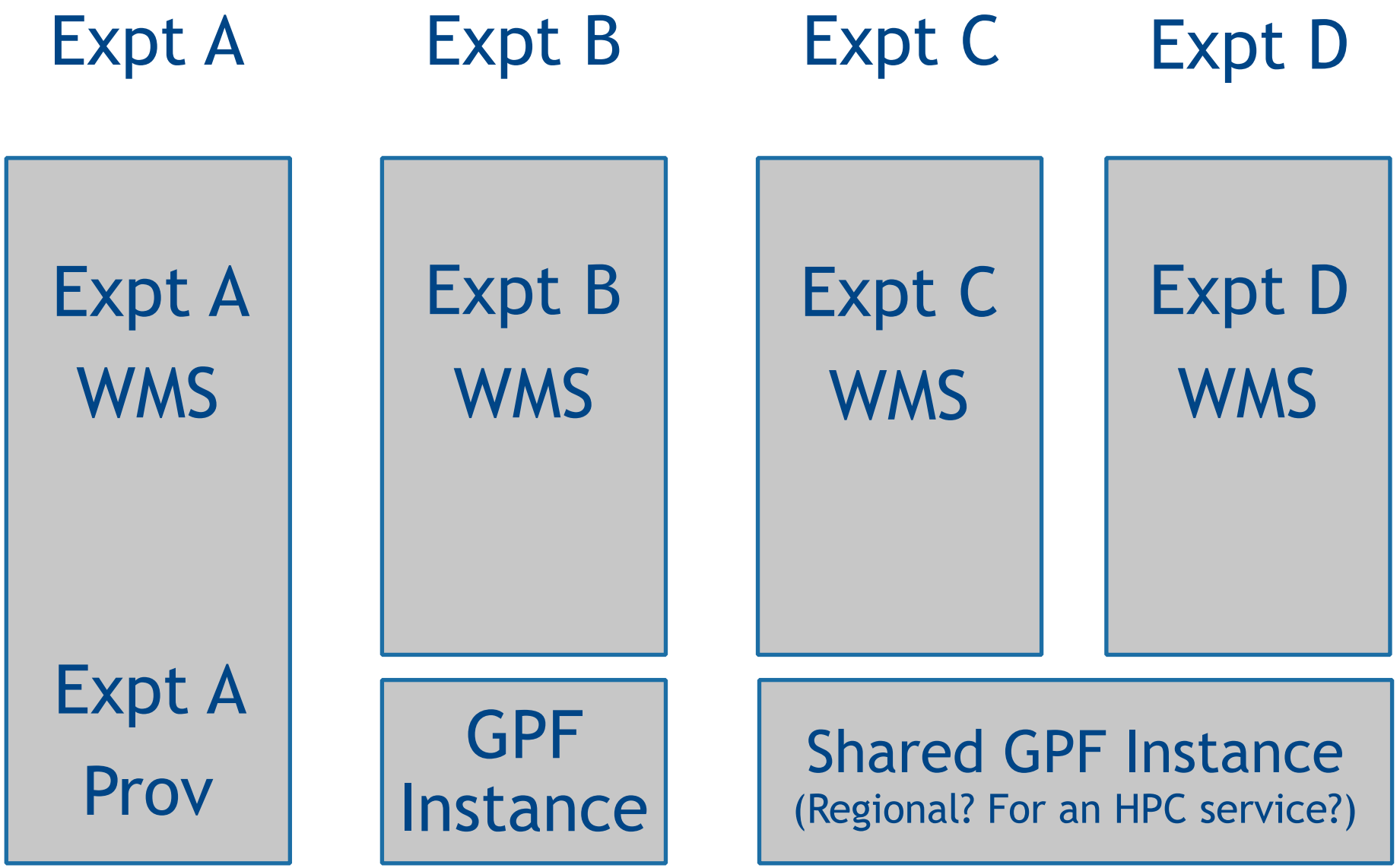
Generic pilot factories?

- Can we agree APIs which make it easy for a pilot factory to be used to do provisioning for different workload systems?
- Push or pull? “I need this” vs “They have this”?
- It would need some way of describing offered / requested Job/VM slots
  - Like the slot descriptions in the medium timescale JSON files! Perhaps.
- I’m **not** suggesting we all use the same pilot factories
  - But it would be good to have **some more** commonality in the workload area

More opportunities

- Are there other opportunities in this sector?
- What about reusable modules?
  - Can we avoid every experiment needing to code up support for each “edge case” site
  - eg each HPC service?
  - eg each commercial cloud provider?
- Could compute provisioning be done by regions or (large) sites in some cases?
  - eg another way of applying funding-driven allocations?

Generic Pilot Factory API in the mix



Conclusion

- This area of Compute Provisioning has some opportunities
  - To agree APIs to allow sites to work more efficiently
  - To scale up to more experiments/projects
  - To share tools (eg generic pilot factories)
  - To prioritise work more effectively
- Are people interested in discussing this?
  - Eventually a working group or preGDB?