



Contribution ID: 119

Type: Oral presentation to parallel session

## Running jobs in the Vacuum

*Thursday, 17 October 2013 11:22 (22 minutes)*

We present a model for the operation of computing nodes at a site using virtual machines, in which the virtual machines (VMs) are created and contextualised for virtual organisations (VOs) by the site itself. For the VO, these virtual machines appear to be produced spontaneously “in the vacuum” rather than in response to requests by the VO. This model takes advantage of the pilot job frameworks adopted by many VOs, in which pilot jobs submitted via the grid infrastructure in turn start job agents which fetch the real jobs from the VO’s central task queue. In the vacuum model, the contextualisation process starts a job agent within the virtual machine and real jobs are fetched from the central task queue as normal. This is similar to ongoing cloud work where job agents are also run inside virtual machines, but where VMs are created by the virtual organisation itself using cloud APIs.

An implementation of the vacuum scheme, *vac*, is presented in which a VM factory runs on each physical worker node to create and contextualise its set of virtual machines.

With this system, each node’s VM factory can decide which VO’s virtual machines to run, based on site-wide target shares and on a peer-to-peer protocol in which the site’s VM factories query each other to discover which virtual machine types they are running, and therefore identify which virtual organisations’ virtual machines should be started as nodes become available again, and which virtual organisations’ virtual machines should be signaled to shut down. A property of this system is that there is no gate keeper service, head node, or batch system accepting and then directing jobs to particular worker nodes, avoiding several central points of failure.

Finally, we describe tests of the *vac* system using jobs from the central LHCb task queue, using the same contextualisation procedure for virtual machines developed by LHCb for clouds.

**Primary author:** MCNAB, Andrew (University of Manchester (GB))

**Co-authors:** STAGNI, Federico (CERN); UBEDA GARCIA, Mario (CERN)

**Presenter:** MCNAB, Andrew (University of Manchester (GB))

**Session Classification:** Distributed Processing and Data Handling A: Infrastructure, Sites, and Virtualization

**Track Classification:** Distributed Processing and Data Handling A: Infrastructure, Sites, and Virtualization