HPC Profile:

No argument to closing out use case document

- Hiro wants GGF copyright

Andrew: don’t like the use of the word “scheduling” w.r.t. BES

Marvin: revised BES and revised JSDL
To be addresses in BES:

- extensible state model,
- extension to base case that are in the specification (should they be removed)
- info model (what should be in the base case)
- e.g. Windows is not POSIX compliant, so POSIX application description is an issue
- activity operation (e.g. creation parameters)
- scheduler operations (should they be in BES or a separate interface)

In JSDL:

- can’t run a program without the POSIXApplication element
- perhaps an HPC base extension that would have the parts of current POSIX app that are common to Windows and POSIX
- no data staging
- which resource elements are needed
- applications elements instead of program description

Andrew will add Trackers for these BES issues

Marvin: no batch operations submit or query
Dave: batch out for submission, but in for query
Currently, can’t profile BES to remove these array ops (could only restrict size of array)
Chris: don’t need batch ops for base case, but do need them for one of first extensions
Marvin: batch, notification, idempotence are among the first 3 extensions

Dave: extensible state model is in hand, batch to be discussed in BES session

Restriction means that HPC profile can pick and choose (in its profile of BES) whether to use a given feature (e.g. if single and array operations are separably defined in BES, HPC Profile can pick the single case)
BES does not have to *remove* the operations not used by HPC profile

Data staging is gone from BES’s state model and so BES does not include data staging anymore (only JSDL does)

Dave: BES has resolved to put all management operations in a separate port type, so this is in hand for HPC profile

Dave: extensions to the state model will involve inclusion of operations that operate on those added states (i.e. no more RequestActivityStateChange operation)

Marvin: extensions to state model will include new operations that will move the job to that state – operation will fail if movement from current state to the new state is not allowed

Ravi: what happens if there is no suspend state when a job is submitted?
Andrew: job runs when container is ready to run it
Marvin: expect suspend (or hold state) to be part of initial set of extensions to the base case

Ian: Notifications and idempotence are optional in the current BES spec and so HPC profile is ok

Marvin: createInSuspendState parameter to BES's create method doesn't make sense without a suspend state in the state model
Dave: pass this on as an issue to be figured out by BES (BES has not decided how to handle optional arguments)

Marvin: multiple renderings (WSRF and resource-model free)
Dave: largely agreed upon though BES doc needs to be redone to reflect this

Marvin: must decide how to describe/publish the extensions a BES container supports (and what those extensions are)
Andrew: we want to do this
Dave: can HPC profile do this without modifying BES?
Glenn: yes, HPC profile could define its own port types
Marvin: ok

Marvin: JSDL restrictions include having no elements that are meaningless in the Windows world
Jay: some elements should be in a "base case" and some elements should stay in POSIX

Andrew: data staging is not a UNIX of Windows concept
Marvin: its elements like "mount source" and "mount point" that are the problem
Chris: it's really the file system elements that are the issue
Marvin: no, it is more than that - some scheduling systems don't implement the notion of stage-in / stage-out
Andrew: is removing stage-in/out making the system non-useful (even if lowest common denominator)?
Ravi: I'm ok with taking out data staging
Glenn: restrictions to JSDL are part of the HPC profile and don't effect JSDL spec right?
Marvin: yes
Chris: HPC profile will say "these are the required JSDL elements" in an HPC profile compliant JSDL description

Andrew: but will they be allowed?
Chris: they can be, and you can throw a fault for anything your implementation doesn't understand
Marvin: it is therefore useful to have a means of discovering supported extensions so that you don't have to send a service a message and see if you get a fault

Dave: you can profile JSDL 1.0, or profile the emerging JSDL 2.0 spec
Andreas: could also use the JSDL 1.1 spec

Chris: perhaps JSDL should admit that "mount point" is something that is not going to be implemented by anyone

Andreas: perhaps easier to add a BasicHPCExecution element extension to JSDL

Mark: have services advertise which JSDL elements they support

Ravi: to summarize, BES is moving toward separate port types for operations which makes profiling easier
Is every method is in a different port type, won't there be dependencies that each new profile will want to change in the main spec?
Jay: no
Ravi: advice to any low level spec authors is to make their specifications highly granular
Marvin: yes, that's good sw engineering and so it applies here

Marvin: JSDL may not need info model, but HPC profile does need it
Andrew: is it really required for base case? What if all jobs can be assumed to run on all resources?
Marvin: good point, but info model is really useful
Marvin: maybe there is a simpler info model than the info model group is currently pursuing – akin to the base + extensions model of the HPC profile
Jay: maybe something simpler than XQuery
Ellen: how about SQL – it's got more experts than XQuery
Jay: I was thinking simpler is based on how large a software component you need to import (XQuery vs. SQL processing is not much simpler)

Dave: OGSA secure channel profile might be sufficient for base case of passing user credentials
Andreas: will send email to HPC profile list telling people to look at this

Marvin: session is closed