**Key Topics**

* Time line to release
  + What activities have to happen to achieve rel 1.0?
  + What is the release timeline for intermediate releases leading up to rel 1.0?
* Simple walkthrough – Intel (Frank Berry)
  + Simple illustrations showing how an activity is performed using verbs, then how it is performed using libfabrics.
* Create pseudo-code walkthroughs
  + Objective is to think more deeply about the existing APIs as applied to well-known apps
  + Identify gaps
* Release requirements
  + Define the minimum set required to support rel 1.0
* Test framework
  + Support testing without a requirement for a specific job launcher
  + Identify specific frameworks (Unit test, Functional test, etc) that will be needed
* Models for implementation – LANL (Howard, Susan)
  + Apps writing to ibverbs tended to rely on existing micro benchmarks (e.g. perftest) as a model for implementing certain functions over verbs
  + Libfabrics has a similar need, but should not be based on microbenchmarks.
* Verbs extensions – MLNX
* Management and Addressing
  + Discuss assumptions related to e.g. endpoint identification, assumptions related to how much information an end user app must maintain
* Continue detailed API review and walkthroughs

**Proposed Agenda**

|  |  |  |  |
| --- | --- | --- | --- |
| Tuesday Morning – 8/19 | | | |
| **Start** | **Duration** | **Topic** | **Notes** |
| 8:00 | :30 | Gathering |  |
| 8:30 | :30 | Introductions, ground rules, agenda bashing |  |
| 9:00 | :30 | Timeline to 1st release | * What has to happen to achieve rel 1.0 * Intermediate releases leading to rel 1.0 |
| 9:30 | 1:00 | Simple walk through: verbs, libfabrics | Intel (Frank Berry) |
| 10:30 | :15 | Break |  |
| 10:45 | 1:15 | Set-up a pseudo-code exercise |  |
| 12:00 | 1:00 | lunch |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Tuesday Afternoon - 8/19 | | | |
| **Start** | **Duration** | **Topic** | **Notes** |
| 1:00 | 1:30 | Pseudo-code small group session | Break into small teams |
| 2:30 | :15 | break |  |
| 2:45 | 1:30 | Review pseudo-coding results | Full group |
| 4:15 | :45 | Test framework discussion |  |
| 5:00 |  | Adjourn |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Wednesday Morning – 8/20 | | | |
| **Start** | **Duration** | **Topic** | **Notes** |
| 8:00 | :30 | Gathering |  |
| 8:30 | :30 | Review, agenda bashing |  |
| 9:00 | 1:00 | Requirements for release 1.0 – group discussion | 1. What is the minimum set of APIs, 2. How is this choice validated, 3. How to confirm that each is properly constructed |
| 10:00 | :30 | break |  |
| 10:30 | 1:00 | Models for implementation |  |
| 11:30 | 1:00 | Management and addressing assumptions |  |
| 12:30 | :30 | lunch |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Wednesday Afternoon – 8/20 | | | |
| **Start** | **Duration** | **Topic** | **Notes** |
| 1:00 | :90 | Verbs extensions - mlnx |  |
| 2:30 | :15 | break |  |
| 2:45 | 1:00 | Continue detailed API reviews |  |
| 4:00 |  | Adjourn |  |
|  |  |  |  |
|  |  |  |  |

Raw notes

* Timeline –
  + What are the characteristics of what is first released?
  + Potential release schedule? What are the intermediate steps that lead us to rev 1.0. Should include a “release candidate” that allows application-level experimentation and a final release.
  + When will applications be ready to deploy?
* Walk through simple example(s) of how an application (examples: SHMEM, PGAS, MPI) would actually use libfabrics relative to libibverbs.
  + Here’s how it works with verbs, here’s how it would work using libfabrics.
  + Not planned to be a detailed description e.g. of MVAPICH or Open MPI rather a simple ‘stage setting’ exercise.
  + Since its comparative, it won’t illuminate the use of libfabrics-unique features.
  + **AR Frank Berry** – pull together at least a simple example, even if not quality code.
* Define the set of requirements for a successful first release
  + which requirements can we hit, which will we miss?
  + May require voting
* Pseudo-code walk throughs
  + Avoids the problem that there prototype providers don’t exist.
* Systematically go thru the APIs one-by-one. Parameters, sizes, functions, behaviors, etc.
* Review specific code and/or man pages, identify gaps
* Work on ideas that have been raised during these meetings including work on verbs extensions?
  + Most of the concepts so far map nicely onto verbs extensions
  + **AR Liran Liss** – work on agenda for verbs extensions discussions for the meeting in two weeks
* **F-2-F** **Date 8/19, 20, Location – Hillsboro, OR**
* Suggestion: Add a discussion of test frameworks to allow testing without having to rely on someone’s job launcher and to identify a specific framework (Unit test, Functional test, etc) that would need to be built.
* For existing verbs, testing is done as a proprietary function of vendors. In the case of ibverbs, people used the existing microbenchmarks (e.g. perftest does some basic verbs testing) as the model for writing their applications.
* We need both, example code implementations, and some sort of unit test or validation.
* AR – Howard to verify he can travel, if so he may be able to lead this.
* Susan Coulter has also volunteered to take up this topic.
* Frank Berry has an RCG who is digging into some examples, and will be able to provide some illustrations a) how this is done in verbs, and b) now this is how it’s done in libfabrics. The parallels are imperfect because verbs does not have some capabilities (e.g. RD) that are available in libfabrics.