Draft Requirements List: Accessing Byte-addressable NVM using OFI

#	Feature	Description	Comment
0	**General Requirements**		
0.1	Byte granular access	Generic byte granular interface which handles access to both DRAM and other types of byte addressable storage	
0.2	RDMA Read, Write, Send, Receive, atomics	Support all flavors of RDMA access semantics	May be limited per endpoint or device
0.3	Expose NVM specifics	Expose (and make use of) memory type specific access behavior (access latency, staged WRITE, persistency, …)	
0.4	Local and remote storage access	Access locally attached IO memory the same way as networked NVM or DRAM	See also 6.1
0.5	Application private access	Both privileged (OS level) and protected application level private end points	
1	**Work Request Execution Ordering**		
1.1	Strict Ordering	Strictly execute in order	
1.2	Lazy Ordering	Execute in any order/parallel but may signal completion in order posted	Efficient
1.3	Explicit Unordered	Execute in any order/parallel but signal effective completion order	Even more efficient
1.4	Ordering Selectable	Strict/non strict per WR, Explicit/Lazy per EP or device	
1.5	Fencing	Barrier: Halt execution of successive WR's until this done, execute all previous before	Per WR
1.6	Fusing	Execute 2 successive WR's in order, stop on first failure	E.g.: Atomic Compare + READ/WRITE, READ + TRIM,
2	**Write Completion Level (per WR)**		
2.1	Write/Send with lazy completion	Data have reached peer but may have not been written	May be lost if power outage
2.2	Write/Send committed	Data have been written to target storage device	Data in persistent store if NVM
3	**Read and Write Acceleration**		
3.1	Read Ahead	Explicit hint to pre-fetch data for successive READ's	Per WR
3.2	Write More	Explicit hint successive WRITEs are to be expected	Per WR
4	**Memory Registration and Addressing**		
4.1	Zero Based Addressing	Support zero based addressing for memory reference in WR	See 4.2 vs. 4.3 and 4.4
4.2	Registration by mapped VA	Allow NVM registration using VA from resource mapping	

4.3	Registration by opaque resource ID	Allow registration of memory resource by provider interpreted opaque handle	e.g. file name, path, requires ZBA
4.4	Re-registration by reservation key	Allow re-registration of previously registered persistent memory object	e.g., resume reservation after reboot
4.5	Resize memory registration	Shrink or grow memory reservation while maintaining same reservation key	
5	**NVM Specific Commands Support**		
5.1	TRIM support	Inline TRIM support per reservation key with offset and length	Implies fencing
6	**Accessing local NVM**		
6.1	Single EP for operations on local storage	Single EP to connect to local service to access local NVM	Embed 'local peer' as OS service
7	**Mix DRAM and NVM access**		
7.1	WR may reference any memory type	Per WR, referenced communication buffers might be of any memory type (requires single local key space)	Only if provider supported
7.2	SGL may reference a mix of storage and DRAM	In a single WR, individual SGE's may reference different types of storage (requires single local key space)	Only if provider supported