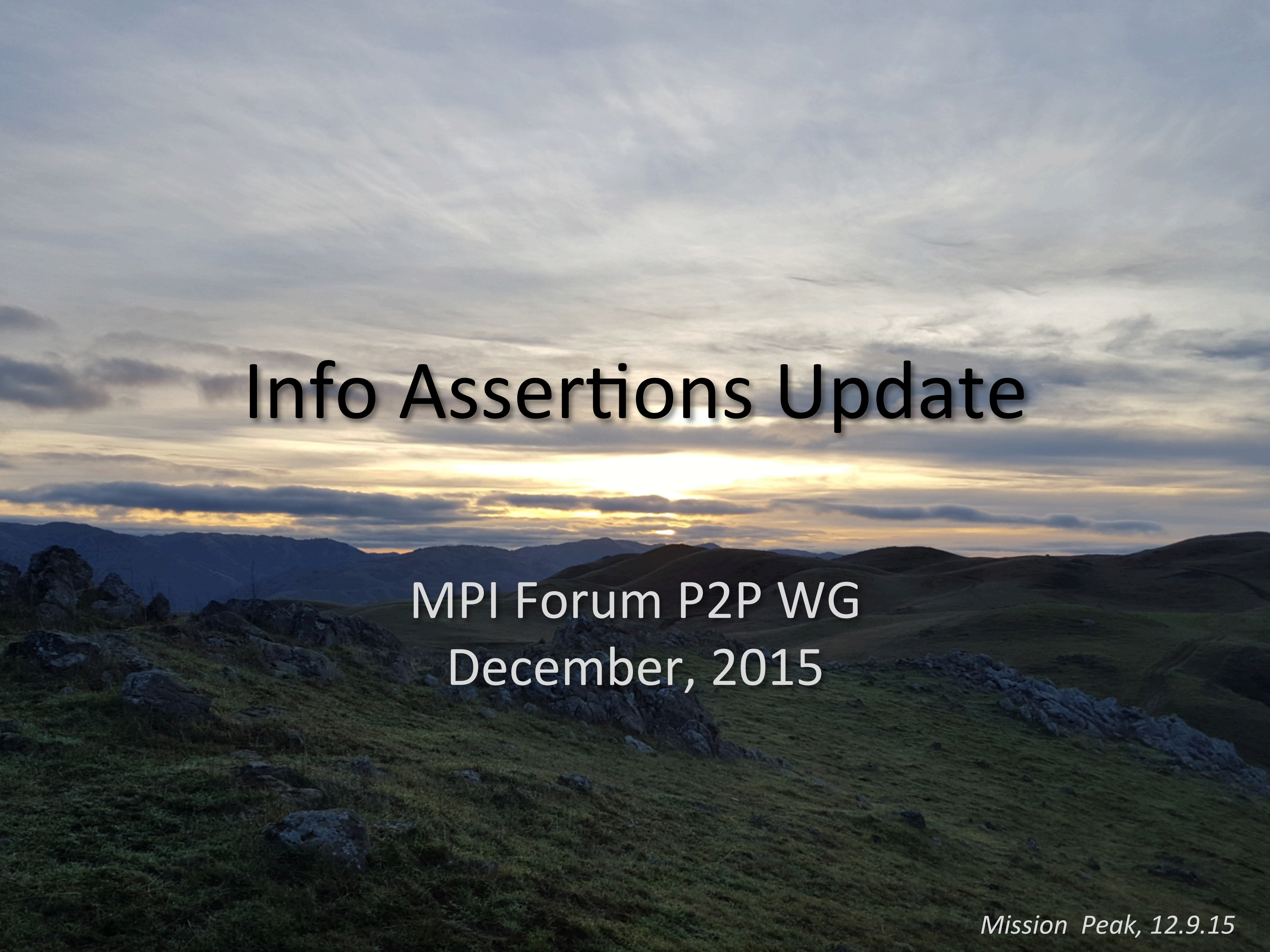
Info Assertions Update

MPI Forum P2P WG

December, 2015



Mission Peak, 12.9.15

Info Assertions Big Picture

- Goal: Allow application to provide guarantees about behavior
  - Guarantees about behavior should not be propagated
  - MPI library can ignore them, but application cannot
- MPI runtime can optimize using knowledge about application's behavior
- Examples of assertions on communicators:
  - No wildcards – optimize message matching
  - No message ordering – use adaptive routing
  - No underflow – optimize rendezvous protocol

Info Keys and Assertions Gotchas

- MPI 3.1, Section 6.4.4:
  - “Hints specified via info (see Chapter 9) allow a user to provide information to direct optimization. Providing hints may enable an implementation to deliver increased performance or minimize use of system resources. *However, hints do not change the semantics of any MPI interfaces.*”
  - Forum opinion was that this text means *both* application and MPI must be able to ignore info hints

- MPI_COMM_DUP also propagates info hints
  - Propagation was added in MPI 3.0
  - Could break libraries if they don’t follow application’s assertions
  - E.g. if a library is passed a communicator with no_any_source set, duplicates it, then uses MPI_ANY_SOURCE

History of the Proposal

- For a while, disagreement within the Forum
  - P2P WG was asked to develop alternative proposals
  - All options were pretty unappealing
    - Separate assertions API, MPI_T Cvars hack, …
- Several RMA info keys already change MPI behavior
  - RMA: no_locks, accumulate_ordering, accumulate_ops, alloc_shared_noncontig
  - Spawn: soft, appnum
- Primary issue is propagation
  - Forum guidance: Removing propagation in MPI_Comm_dup poses little risk of breaking backward compatibility

Proposed Info Changes

- Update info semantics
  - Allow hints to convey application behavior

- Update to communicators
  - Remove propagation in MPI_Comm_dup/idup
  - Add idup_with_info to allow propagation in nonblocking API

- Add communicator info assertions:
  - mpi_assert_no_any_source
  - mpi_assert_no_any_tag
  - mpi_assert_exact_length
  - mpi_assert_allow_overtaking

Can Apps Use These Assertions?

(Simple grep of CORAL, NPB, and Sequoia benchmarks)

		MPI_ANY_SOURCE	MPI_ANY_TAG	MPI_Get_count
CORAL				
Datacentric	BigSort-20130808	N	N	N
Datacentric	kmi_hash	Y	N	Y
~~Micro~~	~~HACCmk~~	~~N~~	~~N~~	~~N~~
~~Micro~~	~~MILCmk-v1~~	~~N~~	~~N~~	~~N~~
~~Micro~~	~~UMTmk1.2~~	~~N~~	~~N~~	~~N~~
~~Micro~~	~~amgmk-v1.0~~	~~N~~	~~N~~	~~N~~
~~Micro~~	~~nekbone_kernel-2.0~~	~~N~~	~~N~~	~~N~~
~~Micro~~	~~stassuij~~	~~N~~	~~N~~	~~N~~
Science	HACC	Y	N	Y
Science	LSMS_3_rev237	Y	Y	N
Science	nekbone-2.3.4	Y	N	Y
Science	qball_r140	N	N	N
Skeleton	ALCF_MPI_Benchmark_v1.01.BGQ	N	N	N
Skeleton	HACC_IO_KERNEL	N	N	N
Skeleton	IOR	N	N	N
Skeleton	LCALS-v1.0.1_Benchmark	N	N	N
Skeleton	MADNESS	Y	N	Y
Skeleton	STRIDE_v1.1	N	N	N
Skeleton	XSBench	N	N	N
Skeleton	clomp_v1.2	N	N	N
Skeleton	ftqV110	N	N	N
Skeleton	pynamic-1.3	Y	Y	Y
Throughput	AMG2013	PROBE	N	Y
Throughput	UMT2013	N	N	N
Throughput	homme1_3_6	Y	Y	N
Throughput	lulesh	N	N	N
Throughput	mcb-20130723	N	N	N
Throughput	miniFE_openmp-2.0-rc3	N	N	N
Throughput	qmcpack-coral	Y	N	Y
Throughput	snap-v1.04	N	N	N

		MPI_ANY_SOURCE	MPI_ANY_TAG	MPI_Get_count
NPB	BT	N	N	N
	CG	N	N	N
	DT	N	N	N
	EP	N	N	N
	FT	N	N	N
	IS	N	N	N
	LU	N	N	N
	MG	N	N	N
	SP	N	N	N
Sequioa	AMG2006	Y	Y	Y
	AMGmk_v1.0	N	N	N
	CrystalMk_v1.0	N	N	N
	IOR-2.10.1_sequoia-1.0	Y	Y	N
	IRSmk_v1.0	N	N	N
	STRIDE_v1.1	N	N	N
	UMT_v1.1	N	N	N
	UMTmk_1.1	N	N	N
	clomp_v1.0	N	N	N
	irs.1.0	N	N	N
	lammps-22Jun07	Y	N	Y
	phloem-1.0.0	N	Y	N
	pynamic_v1.0	Y	Y	Y
	sphot_v1.0	Y	Y	N

*Note: Did not look at libraries, CORAL apps use FFTW, HDF5, MKL

Can Implementations Use Assertions?

- No wildcards (mpi_assert_no_any_source, mpi_assert_no_any_tag)
  - *The process will not use MPI_ANY_TAG/SOURCE on the given communicator*
  - Enables message matching optimizations
    - Use hash tables for posted receive and unexpected message queues
    - Reduce overheads from managing MPI_ANY_SOURCE operations when separate shared memory / network queues are used

- No message ordering (mpi_assert_allow_overtaking)
  - *Point-to-point comm. does not require operations to match in the order posted*
  - Enables network ordering optimizations
    - Use adaptive routing for networks that use ordered mode for envelope info
    - Reduce overheads for networks that do receiver-side reordering prior to matching

- No underflow (mpi_assert_exact_length)
  - *Lengths of messages received equal lengths of the receive buffers*
  - With underflow, receiver does not know if sender will use eager or rdzv.
  - Allows receiver to know ahead of time and optimize xfer protocols
    - Can handle eager/rendezvous through separate mechanisms

Logistics

- Used old-style markup to generate color PDF
  - PR contains the markup
  - If reading is successful, WG will generate a clean PR prior to voting

- Without further ado …