

UNIT - 5

① MPI Performance issues

→ Large communication overhead.
Thus,

① Choose algos with min. cross process dependencies.

② Combine multiple message into one.

→ MPI is static work distribution in nature.

→ Compute anything that doesn't need non-local variables while using non blocking MPI calls.

② MPI Safety Issues

→ Lot of low level details are left to programmer.

→ Both sending & receiving should match.

→ Deadlock & livelock are possible.

→ Mutual exclusion isn't needed cause no shared memory.

③ Unified Parallel C

→ Gives global view of address space.

→ 'Shared' arrays are distributed in cyclic or block cyclic manner.

→ ~~Ext~~ Extends C, supports pointers.

↳ Private
↳ Shared

→ 4 types of pointers

	Pointer	
	Private	Shared
Reference variable it is pointing to	Private	Shared
	Private - Private P_1	Private - Shared P_2
	Shared - Private P_3	S-S P_4

→ `upc_forall` : Distributes `for`.

④ Titanium (Ti)

- Extends Java
- Communication code based on one-sided communication library.
- Object oriented
- regions : safe performance oriented memory management.
- `foreach` : unordered iteration
- `forall` \longleftrightarrow Concurrent
- 'single' : variable shared & synchronized

⑤ Evaluating Existing Approaches

① POSIX threads

- Shared memory
- Any thread can interact with other threads' stuff.
- Does not encourage locality.
- Heavy dependency on locking.

② OpenMP

- Simple & easy to use
- Some forms of parallelism can't be implemented.
- Provides forall, barriers & reduce.

③ Java Threads

- Multiple levels
 - ↳ One is similar to POSIX
- Concurrent data structures

④ MPI

- Need to specify low level details
- Redundant specifications.
- Reduce, Scan are provided.
- Distributed memory programming model.

⑥ Hidden Parallelism

- Bit-parallel functional units
- Multiple functional units
- Pipelined execution
- out of order execution
- DMA controllers
- Prefetch units
- Trace caches
- Simultaneous multithreading
- Vector processors
- Chip multiprocessors
- Co-processors

⑦ Transparent Performance

- Ability to reason the performance encourages programmers to improve it.