

MPI Apps: How We Use MPI

**Next-generation scalable applications:
When MPI-only is not enough**

**Richard Barrett
Scientific Computing group
Oak Ridge National Laboratory**

**Bishop's Lodge,
Santa Fe, NM
June 3, 2008**



Managed by UT-Battelle for the
U. S. Department of Energy

UNCLASSIFIED // FOR OFFICIAL USE ONLY



MPI View of the Universe

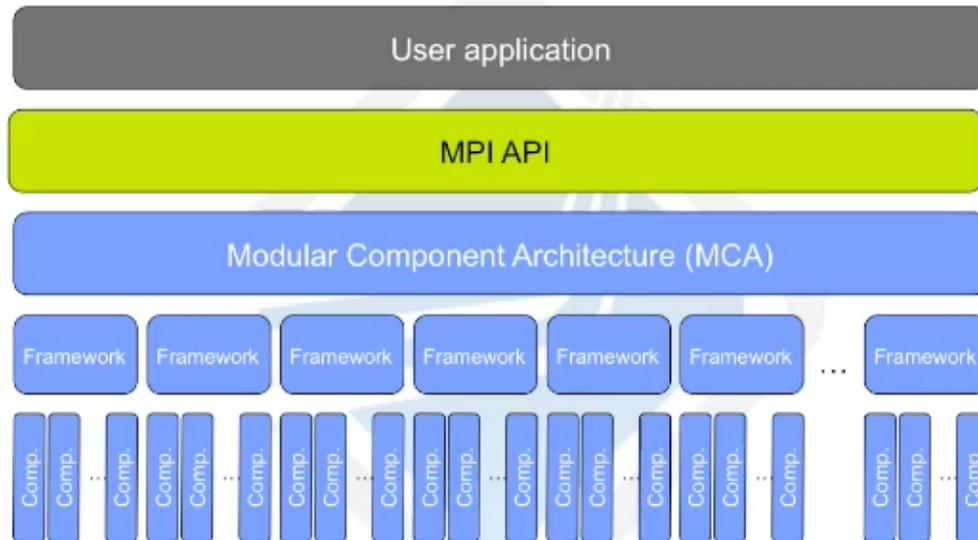
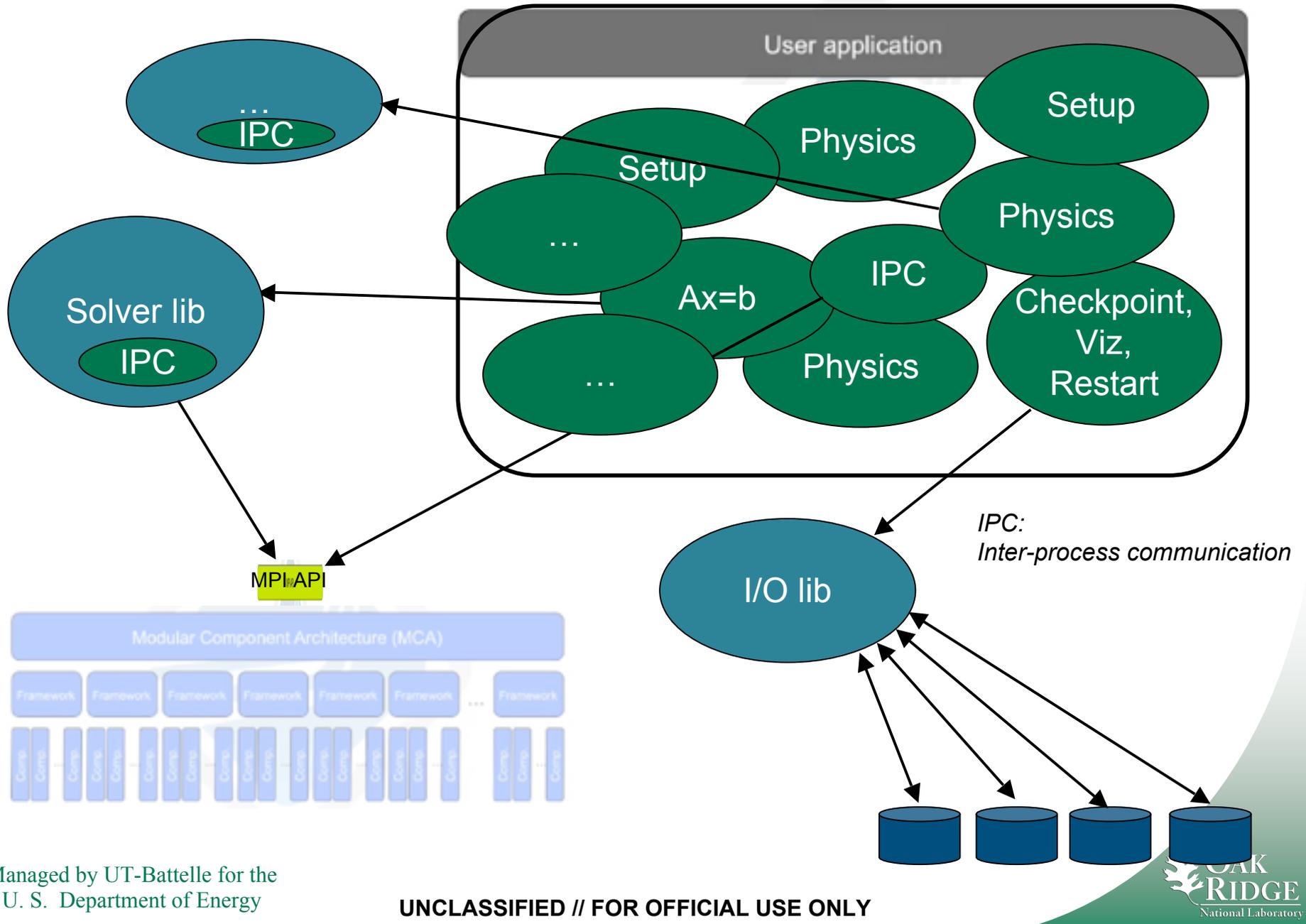


Image from OpenMPI web pages.

Code developer view of the Universe



Application MPI use

<i>Code</i>	<i>Logical SLOC</i>	<i>MPI</i>		
		<i>pt2pt</i>	<i>coll</i>	<i>All</i>
AORSA	20,671	2	0	3
GYRO	44k	2	4	17
HYCOM	18,533	3	3	18
MCNP	>100k	4	5	21
Moldt	767	0	0	0
NPB 3.3	20,671	4	5	33
POP (1.4.3)	16,770	4	3	23
POP (2.0.1)	21,304	4	4	17
ROMS	110,984	4	6	17
S3D	19,483	3	7	22
SAGE	109,000	5	6	37
sPPM	3,752	2	1	10
Sweep3d	1,081	2	3	9
<i>Totals</i>		8	9	80

3rd Party MPI use

Code	Logical SLOC	MPI		
		pt2pt	coll	All
BLACS	7k	6	4	46
HPL	2k	6	0	15
Hypre	70k	7	10	46
MADNESS	40k	2	0	12
ParMetis	7k	4	10	21
PETSc	120k	12	9	55
ScaLAPACK	115k	0	0	0
Tril:epetra	15k	4	8	16
Tril:AzOO	15k	4	0	9
Tril:ML	45k	3	6	25
VisIt	1.8M	6	11	76
Zoltan	30k	8	8	40

MPI functionality used by apps

<i>MPI routine</i>	<i>Apps</i>
MPI_Irecv	10
MPI_Isend	11
MPI_Issend	1
MPI_Recv	10
MPI_Recv_init	1
MPI_Send	9
MPI_Send_init	1
MPI_Sendrecv	1
<i>Support routines</i>	
MPI_Attach_buffer	1
MPI_Cancel	1
MPI_Detach_buffer	1
MPI_Get_count	1
MPI_Iprobe	3
MPI_Probe	1
MPI_Request_Free	2
MPI_Start_all	1
MPI_Test	3
MPI_Wait	9
MPI_Waitall	5

<i>MPI routine</i>	<i>Apps</i>
<i>Collective comm</i>	
MPI_Allgather	3
MPI_Allreduce	11
MPI_Alltoall	4
MPI_Barrier	10
MPI_Bcast	10
MPI_Gather	3
MPI_Gatherv	1
MPI_Reduce	5
MPI_Scatter	1

Application IPC

Parallel Ocean Program (POP)

<i>File</i>	<i>Functionality</i>	<i>SLOC</i>	<i>pt2pt</i>		<i>MPI coll</i>		<i>All</i>	
version 1.4.3								
boundary	Halo exchange	386	3	60	0	0	4	76
communicate	Proc mgmt	119	0	0	3	9	12	39
global_reductions	Reductions	489	2	12	3	44	7	69
stencil	Diff. stencils	796	3	88	0	0	4	118
Total		1,790	3	217	3	53	17	302

version 2.0.1								
boundary	Halo exch	1,078	3	24	0	0	3	24
broadcast	Broadcast	165	0	0	2	34	2	34
communicate	Proc mgmt	99	0	0	1	1	10	24
gather_scatter	Gather/scatter	592	5	28	0	0	5	28
global_reductions	Reductions	880	0	0	2	28	2	28
Total		2,814	5	52	4	63	13	138

POP IPC *cont'd*

Subroutine*	SLOC	Called*	MPI					
			pt2pt	coll	All			
<i>Version 1.4.3</i>								
boundary_2d	18	7	3	10	0	0	3	10
ninept_9	21	6	3	10	0	0	3	10
scatter_global	20	18	3	4	1	2	4	6
global_sum	14	53	0	0	1	1	1	1
<i>Version 2.0.1</i>								
update_ghost_cells	191	39	3	8	0	0	3	8
scatter_global	114	41	3	3	0	0	3	3
global_sum	59	83	1	1	1	1	1	1

Table 11: Select POP communication abstractions

**Double precision versions only from the Fortran module; however, the "Called" value is independent of the datatype.*

Application IPC *cont'd*

SAGE

<i>Subroutine</i>	<i>SLOC</i>	<i>Called</i>
<i>Gather/scatter</i>		
token_put	433	25
token_get	433	8
<i>Global</i>		
token_allgather	61	31
token_allreduce	416	78
token_alltoall	14	2
token_bcast	102	88
token_reduce	172	7
<i>Miscellaneous</i>		
token_build_local	246	2
token_move	180	5
<i>Totals</i>	2,193	246

**h
y
c
o
m**

<i>Subroutine</i>	<i>Function</i>	<i>SLOC</i>	<i>Called</i>	<i>MPI</i>					
				<i>pt2pt</i>		<i>coll</i>		<i>All</i>	
xctilr(1)	Halo exchange	173	169	6	19	0	0	6	19
xctilr (2)	Halo exchange	354	169	6	23	0	0	6	23
xcmaxr.1	Reduction	35	17	0	0	1	1	1	1
xcaget	Convert tiling	77	20	2	4	1	1	3	5
<i>Total</i>	<i>All comm</i>	1,925	196	9	76	3	13	18	93

The problem with abstractions :

Performance

Are we avoiding MPI complexity required for performance?

(Perhaps, but we need *predictable portable* performance.)

But even so, MPI isolates inter-process data sharing from computation.

Logical blocking of tasks, limiting compiler view of “intent” of computation. (Expressiveness issue.)

It is crucial to have a basic awareness/understanding of *all* application developer and usage issues.

“The expected outcome of the workshop is a clear set of directions for scalable application development for the coming decade and beyond”

Any new way must provide a *compelling* reason to switch.

Acknowledgments

- *MPI in Scientific Computation: An Application Survey*, R.F. Barrett, S. Ahern, M.R. Fahey, R. Hartman-Baker, J.K. Horner, S.W. Poole, and R. Sankaran, ORNL Tech Report in progress.
- Contributions from Larry Cox@LANL and Mike Heroux@Sandia.
- Codes from various sources, funded by a variety of programs, including DOE, NSF, DoD, and SAIC.
- Line counts from CodeCount™ v1.0 and SLOCCount© v2.26.