

MCMD Discussion - II

BOF - CCA Meeting

Silver Springs, MD

July 20, 2007

Previous BOF

- Discussed use cases documented on the wiki
- Noted differences between programming models regarding groups
- Outlined basic concepts and capabilities

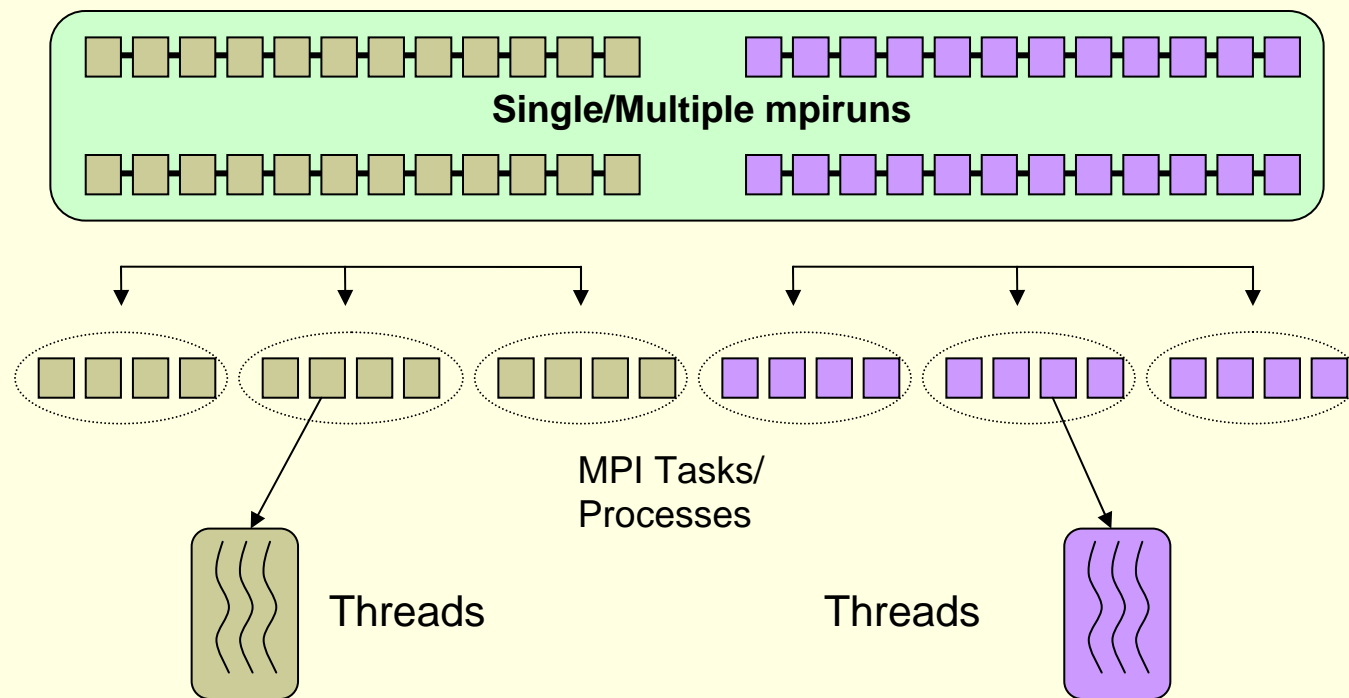
Goals for Today

- Decide on basic capabilities
- Propose architecture and interfaces for group management
 - Post on wiki page for discussion
- So that we can move to do the prototype implementation

Recap: MCMD Use Cases

- So far ..
 - Hierarchical Parallelism in Computational Chemistry
 - Coop Parallelism
 - Ab Initio Nuclear Structure Calculations
 - Coupled Climate Modeling
 - Molecular Dynamics, Multiphysics Simulations
- Fusion use-case: to be described based on yesterday's talk

Recap: Target Execution Model and Global Ids



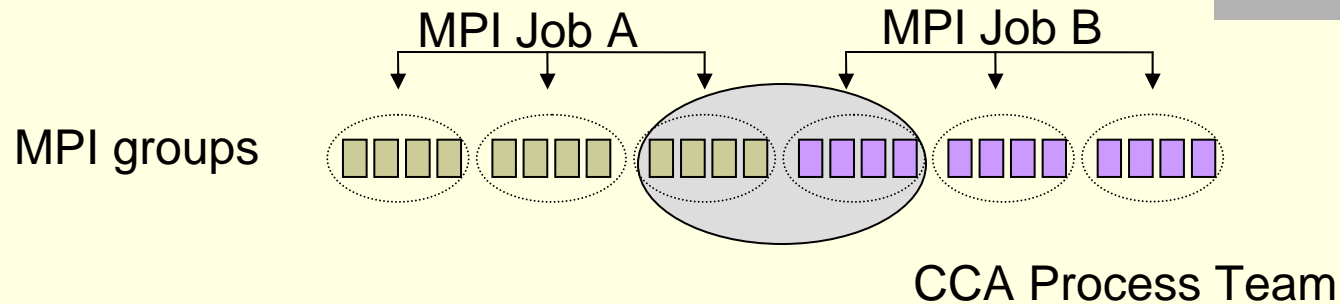
■ Global id specification

■ global id = <machine id> + <job id> +
<task/process rank> + <thread id>

Recap: Processor Group Management

- Various execution models
 - E.g. coop parallelism vs. single mpirun
- Programming Models
 - MPI, Threads, GAS models including GA, ..
- Global process and group ids
- Should be MPI-Friendly (minimum)
- Group translators

Teams or Groups?

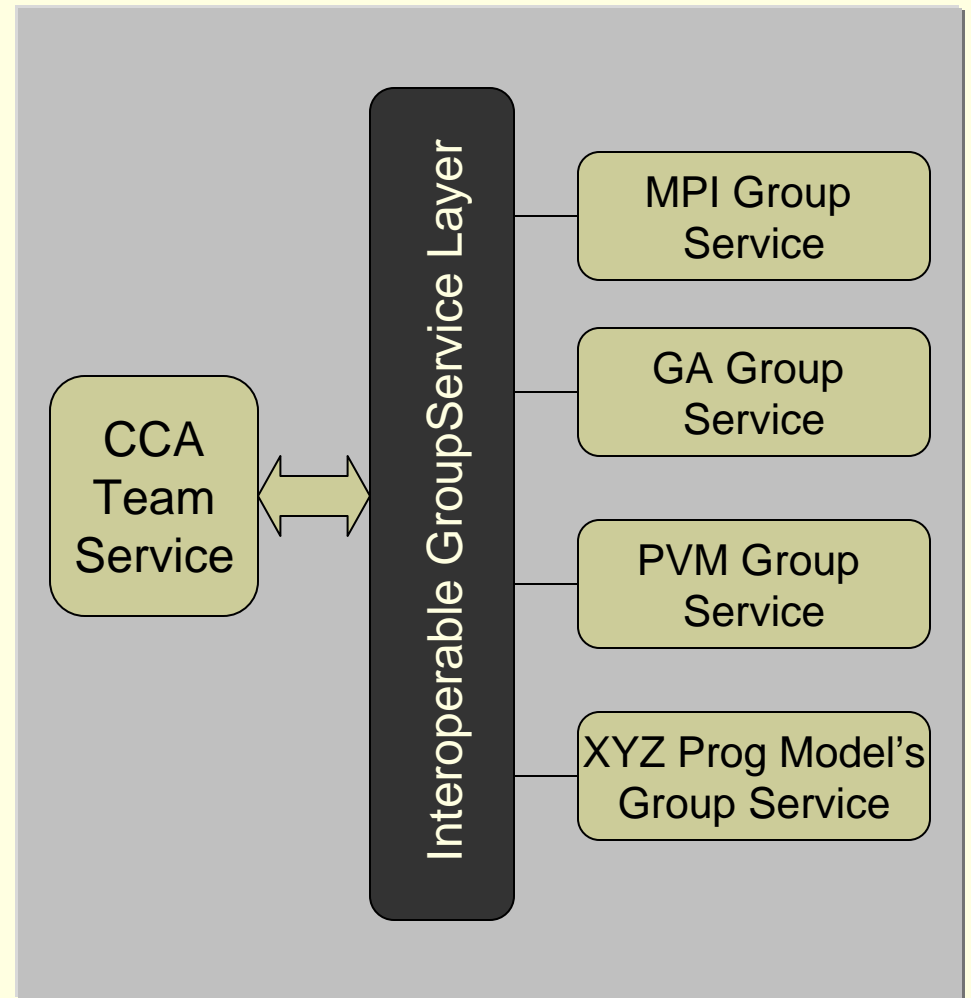
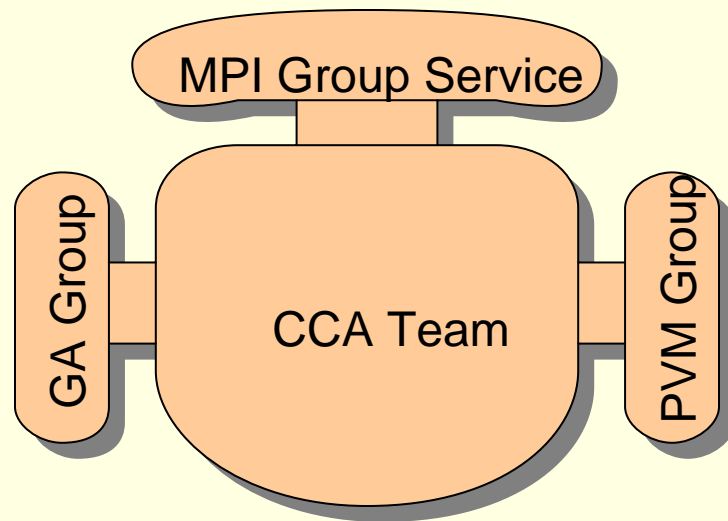


- We propose to use a slightly different term of process(or) **teams** rather than **groups**
 - Avoid confusion with existing terminology and interfaces in programming models
 - Some use cases call for something more general than MPI groups e.g., COOP with multiple mpiruns
 - For example, CCA team can encompass a collection of processes in two different MPI jobs. We cannot construct a single MPI group corresponding to that.
 - Operations on CCA teams might not have direct mapping to group operations in programming models that support groups
 - We will also need CCA service that handles groups

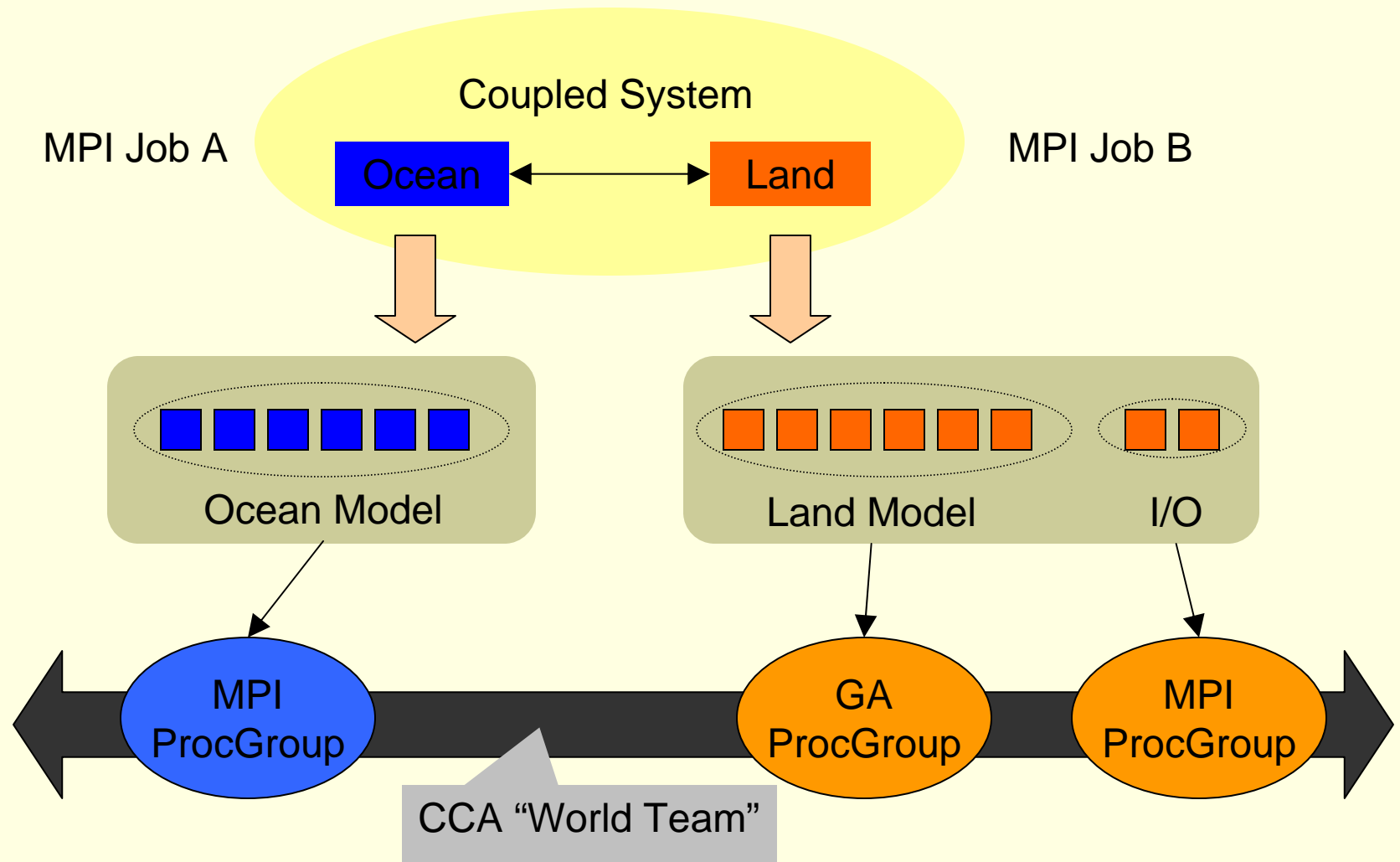
CCA Team Service

- Provides the following
 - Create, destroy, compare, split teams
 - More capabilities can be added as required
 - Assigns global ids to tasks from one or more jobs running on one or more machines
 - Global id = <machines id> + <job id> + < task id>
 - Also, <thread id> if we were to support threads at component level.
 - Locality Information
 - Gets the job id, machines id, task id of the given task
 - Plugins for programming models
 - Translation between Teams and Groups

Plugins

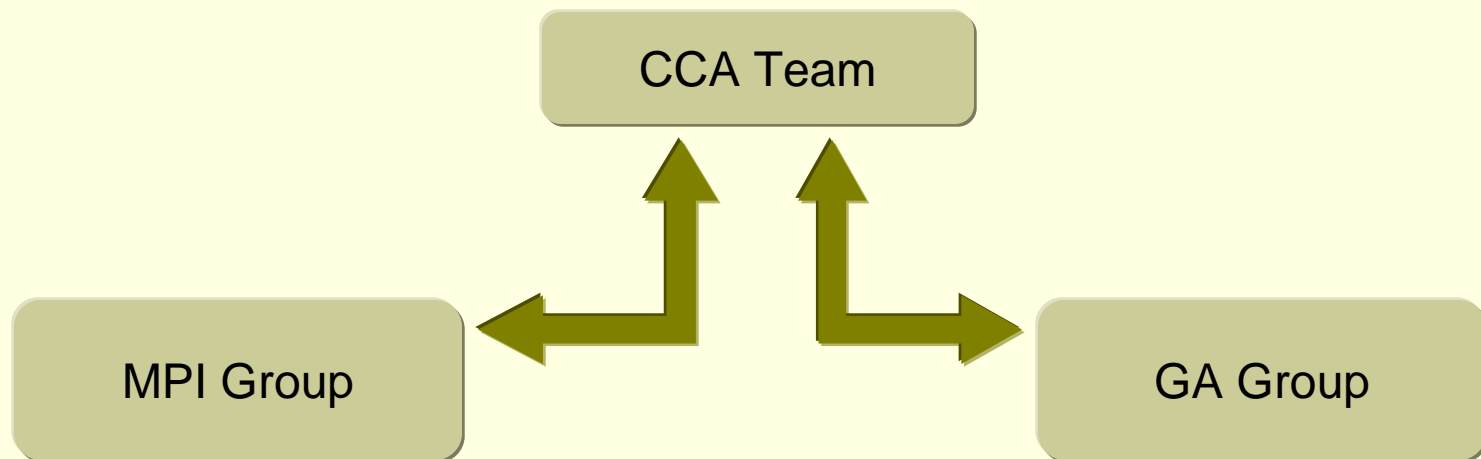


Example



Translators

- Group translation across multiple programming models



Present SIDL Interfaces

Open Questions

- Mixing threads and processes at the component level
 - No use-cases!
 - Do we need explicit thread support in CCA?
- Dynamic or static configuration
 - MPI Dynamic Process Management
 - Launching jobs