Parallel Tools Platform Parallel Debugger

Greg Watson

Project Leader

Design Considerations

- Parallel applications range from 2 to 128K processes
- Combined mult-process and threaded model is possible
- Some debug operations are performed on
  - All or a subset of processes
  - Individual processes
- Processes are typically dependent on each other
- Mixed language (e.g. C and Fortran) typical

Programming Models

- Message passing model
  - Distinct processes exchange data using messages
  - Explicit send/receive and collective operations
- Shared memory model
  - Data structures are shared
  - Locking or atomic operations required
- Currently targeting message passing, shared memory later

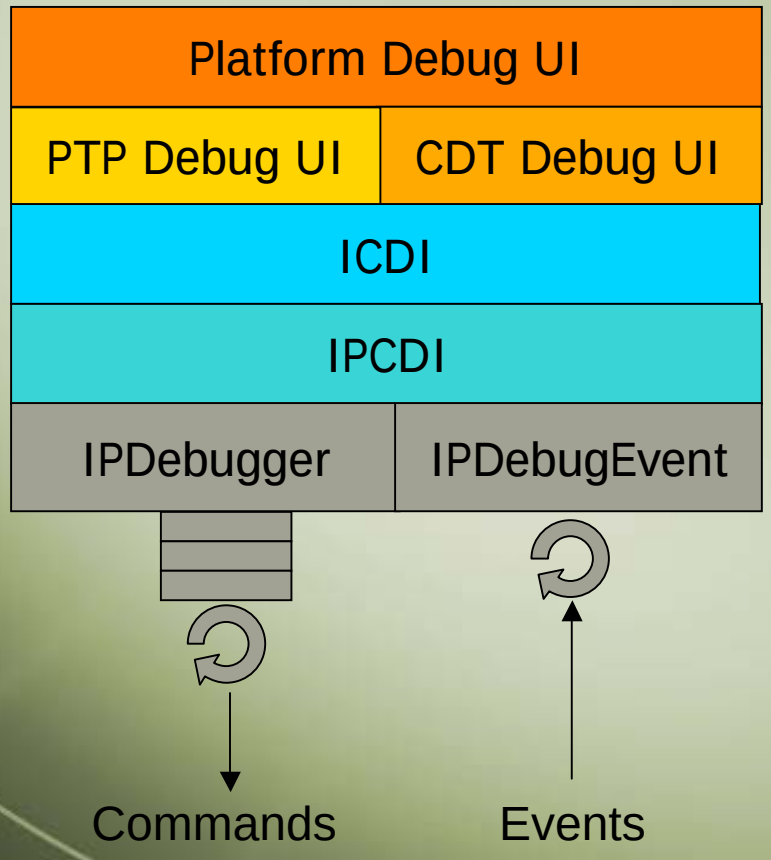
Debugging Methodology

- Breakpoint across all processes to synchronize
  - In master/worker, breakpoint only workers
- Step all processes until error
  - Stepping one process typically not possible due to dependencies
- Examine snapshot of data across all (or subset) of processes
- Compare data from one process to another

Architecture Overview

- Eclipse UI front end
  - Reuse Debug and CDI where possible
- High level parallel debug API
  - Process sets used for efficiency
  - Asynchronous command/event model
  - Extension point to allow alternate backends
- Backend (SDM)
  - Startup
  - Command broadcast
  - Event aggregation
  - MI only used for low level debug actions

UI Architecture



Debug API

```
public interface IPDebugger {
   public void stepOver(BitList set, int count)
        throws PCDIException;
   ...
}
```

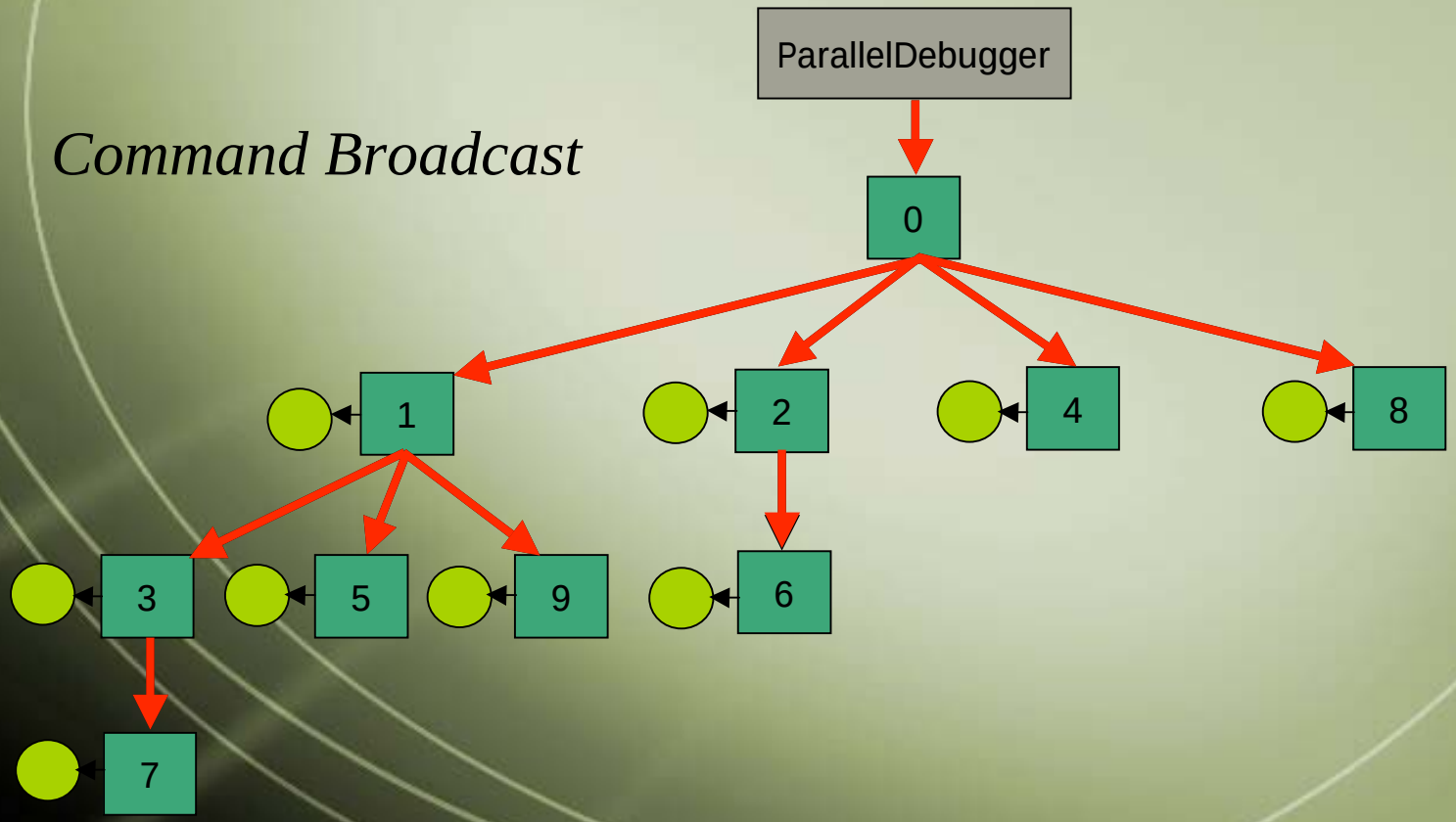
101011010111111101011111

000000000111111101011111

```
public interface IPDebugEvent {
   public BitList getSet();
   ...
}
```

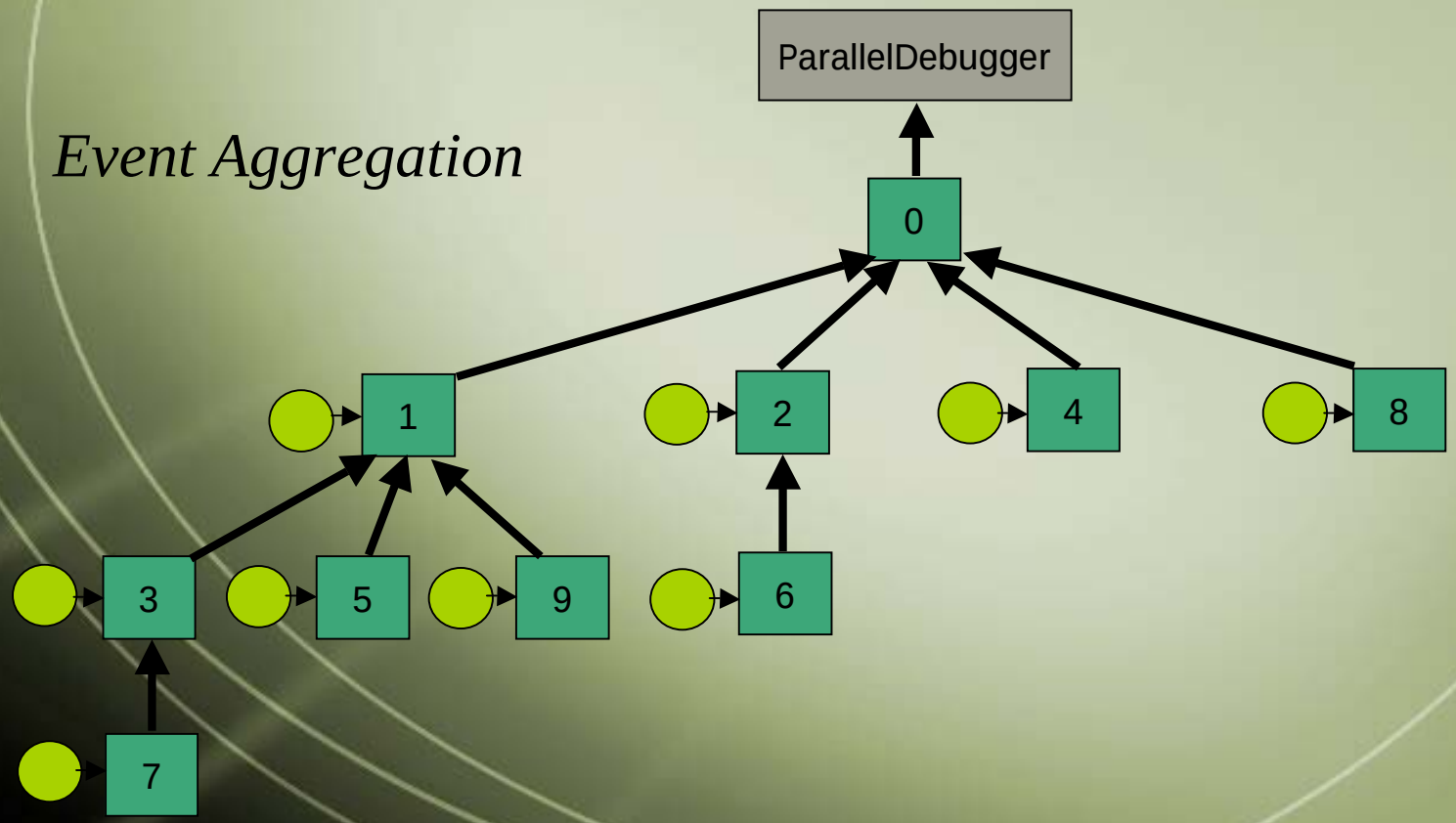
SDM Architecture

Command Broadcast



SDM Architecture

Event Aggregation



AIF

- Architecture Independent Format for data
- Independent of endianness, word size, character size
- Fully represents data type and value
- Can be used to represent complex data structures, such as linked lists
- Supports all C, C++ and Fortran types
- C and Java (partial) implementations
- Library operations:
  - Conversion to/from native format
  - Logical and arithmetic
  - Formatting and display

User Interface

- Parallel Debug View
  - Global view of processes and process sets
  - Tooltips for fast variable access
  - Regiser/unregister process to display in Debug View
- Parallel Breakpoint
  - Global - applies to all processes regardless of job or job size
  - Set-based - applies to a set of processes for a particular job
  - Color used to distinguish current set
- Current line markers
  - Multiple markers allowed
  - Different markers for registered/unregistered processes
  - Text highlighting colors

User Interface (cont…)

- Variable View
  - Supports AIF data
  - Only fetches complex data types when variables are expanded
- Array View
  - Prototype using custom widget
  - Allows 2-D view (slice) of multi-dimensional arrays
  - Likely to move to memory view

Demo