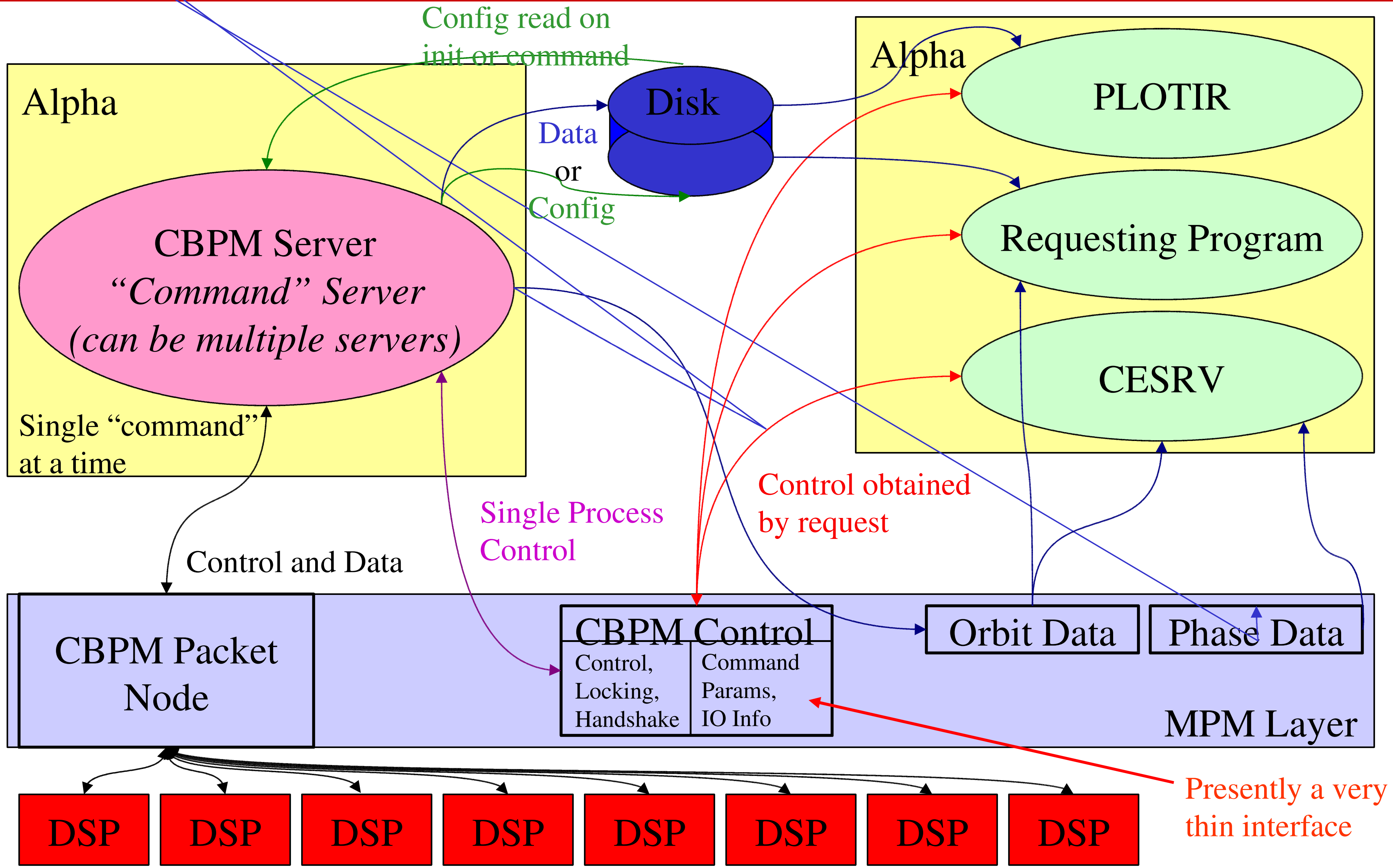
CBPM Server Operation



CORNELL UNIVERSITY LEPP
LABORATORY FOR ELEMENTARY-PARTICLE PHYSICS

Command Server

DSP Commands:

1) Get Orbit Data
2) Get Multi-Bunch Data
3) Get Betatron Phase
4) Get HEP Data
5) Cont Data (EXPERT)
6) Get Continuous Data
7) Get Injection Data
8) Get Raw Data
9) Get Processed Data
10) Beta Phase (EXPERT)
11) Idle Modules
12) Peak Find
13) Calibrate Pedestals
14) Set Gain
15) Set Delay
16) Test Raw Data
17) ALL Bunches
18) Timing Scan
19) Delay Scan
20) Synch DSPs

Control System Commands:

101) Select Menu
102) Change Module Mode
103) Update Act List
104) Display Mod List
105) Display Mod
106) Dump BPM Structs
107) Write Raw File
108) Write Buttons File
109) Read/Write Mod Config
110) Run Command File
111) CESR Interface

Utility Commands:

201) System Status
202) Show CESR Currents
203) DSP Control
204) Set Typeout Level
205) Update DB
206) Help
207) Set Attenuators
999) Exit

DSP Parameters

```
typedef struct {
        int num_samples;
        int num_turns;
        int trig_cnt;
        int spacex_turn;
        int gain_adj;
        int delay_cal;
        int gain_xcal;
        int avg_mode;
        int scale_mode;
        int update_mode;
        int gain[MX_CBPM_CHAN];
        int glob_delay;
        int chan_delay;
        int num_bunches;
        int bunch_id[MX_AUTO_SAMPLE];
        int use_data_enable;
        int trig_mask;
        int fft_plane;
        int phase_mode;
        int pedestal_mode;
        int num_user_params;
        int user_params[MX_USER_PARAMS];
        int checksum;
  } CBPM_XBUS_CMD_PARAMS;
```

- DSP Operation Specified by:
  - Command(s)
    - Possibly a sequence of commands to prepare state
  - Few 10’s of parameters

Server Expansion Issues

- Single Server vs. Multi-Server
  - Multiple detector types doing different things potentially argues for multiple servers
    - Correlated data
      - Configure multiple servers
      - Trigger operation via shared “timing system command” command path
      - Data output
        - Stitch multiple files together
        - Shared memory would allow a “data server” to package data
  - Identical control methods between modules
    - Allows for single large server
    - All data, constants, and control in one place allows for maximum flexibility in manipulating system and data
    - Some inefficiencies (sequential module access vs limited parallelism)
- Server Event Requests
  - Presently have multiple server – single request support
  - New:
    - Single request?
      - Doesn’t support all types of operations unless modules running some default operation autonomously
    - Multiple request?
      - Possibly desirable
        - Straightforward modification to server concept
      - Not clear to me whether absolutely necessary
        - Different types of event requests could be handled sequentially since not truly real-time

MPM versus SHARED Memory

- Shared Memory
  - Offers full program access
    - Data structures laid into memory with direct program access
    - All executable accessing shared memory have to be kept fully up-to-date
  - Direct access available on only one node
  - CRS has pointed out that VMS clustering also supports cluster-level shared memory across cluster nodes!!!!

- MPM Memory
  - Accessible from all control workstations
  - Indirect program access (as presently configured)
    - Mailbox memory space available through MPM call and data array transfer
    - Does not support internal program structuring of memory except by creating an interface package
    - If DAQ given direct access to a section (as opposed to access via VMPMLIB calls), then would have true multi-workstation shared memory