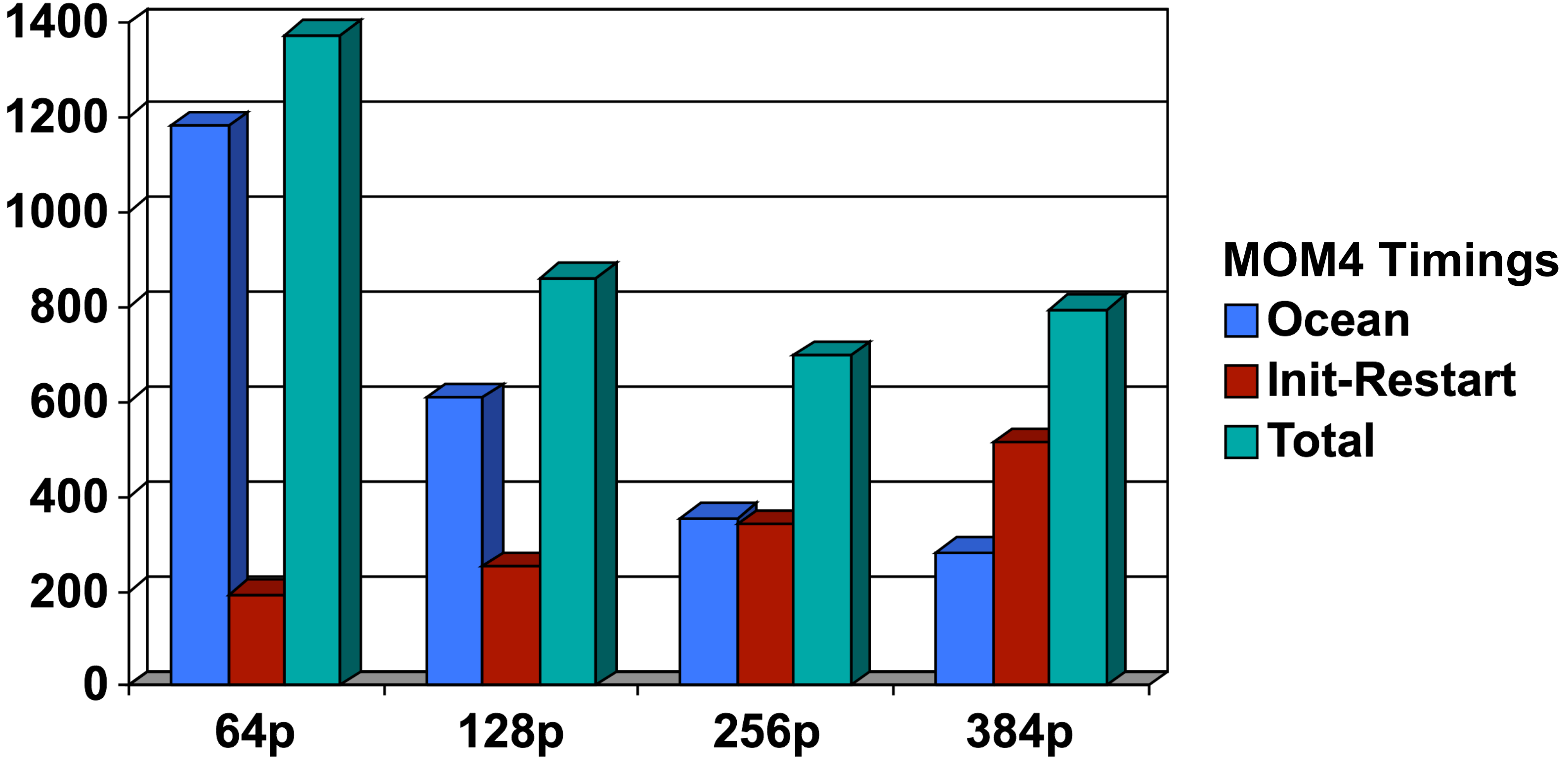
Ocean Forecast Modeling

- Ocean Forecast Australia Model (OFAM)
  - Grid is 1191x968x47
    - Global grid with E-W periodic boundaries
    - 1/10 degree horizontal resolution around Australia
    - coarser resolution outside of Australia region
  - 85 GB of main memory
  - 1 day of simulation for every 10 minutes of wall-clock time on 21 processor NEC SX-6

OFAM Benchmark



- Ocean computations scaled nearly linearly
- Init-Restart time increases with number of processors
- I/O code involve all processors or single processor causing bottlenecks

OFAM Benchmarks

IBM Power 5 Benchmark	Inclusive elapse time	OCEAN section only	Startup-shutdown phase
64 cpu	1375 sec	1182 sec	193 sec
128 cpu	862	609	253 sec
256 cpu	697	356	341 sec
384 cpu	796	283	513 sec

OFAM2 Development

- OFAM 2
  - Development of improved global grid for ocean forecasting
  - Grid size is >4x larger than OFAM
  - CPU time is >8x larger than OFAM
  - Number of CPUs is ~10x more than OFAM to maintain elapse time (64 --> 640)
  - Current MOM4/FMS I/O system is not optimal and does not scale.

MOM4-FMS issues

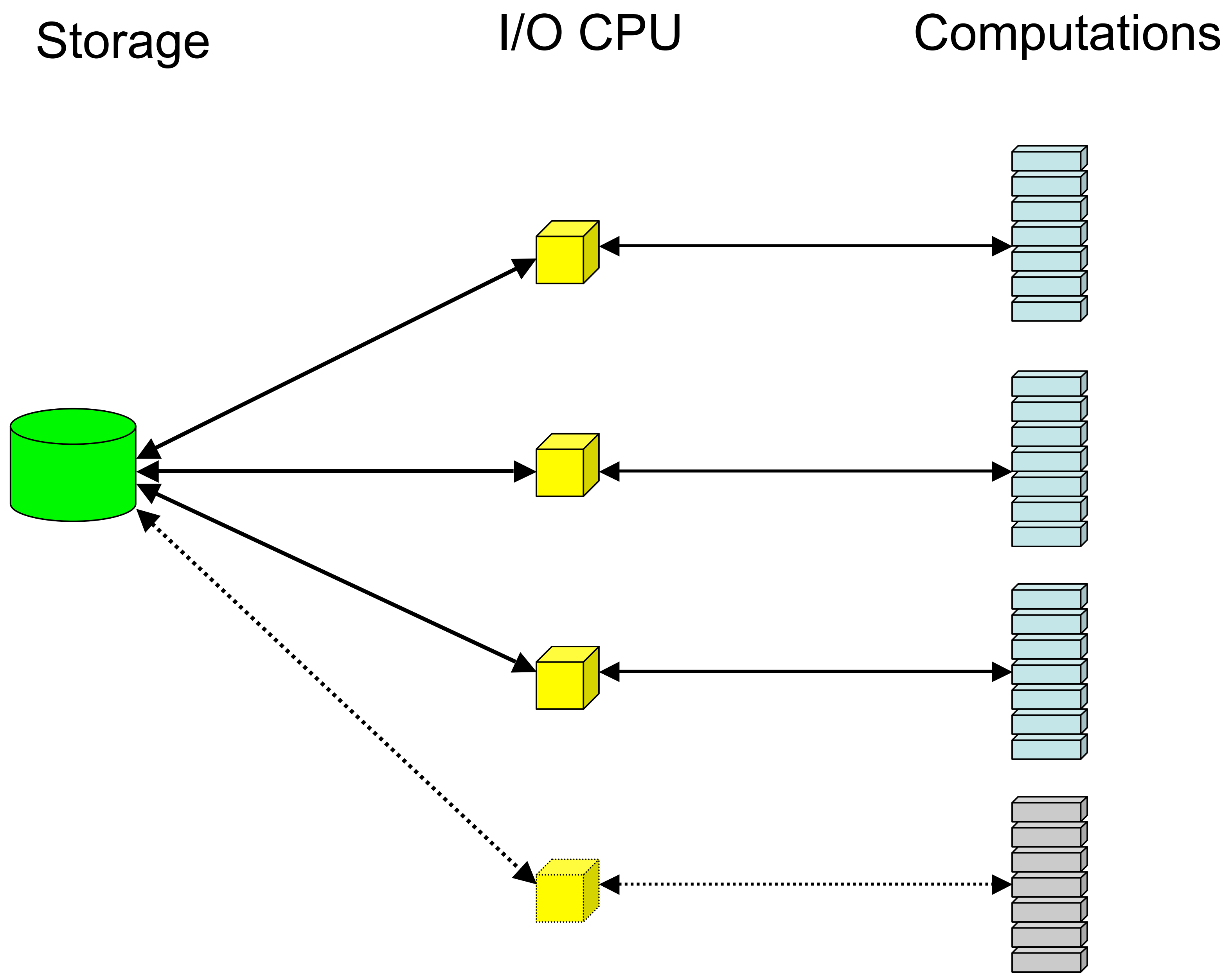
- MOM4 startup-shutdown issue
  - Decreasing read/write performance with increasing # cpus
  - Startup is reading restart file and initial conditions (FMS issue)
  - Shutdown writing restart files (FMS issue)
- FMS I/O architecture
  - Uses single cpu or all cpus for I/O tasks
  - I/O tasks are performed by computational processors
  - Global arrays held in master processor creating high memory requirements on computing architectures.

FMS I/O performance issues

- Issues with FMS I/O routines
  - FMS write/read modes are not adequate for OFAM2
  - FMS is used in MOM4, OASIS, and others
  - FMS is owned by NOAA GFDL with Balaji as project leader
- Investigation of FMS I/O changes to solve issues
  - Investigate I/O fabric architecture
    - Designate I/O processor per group of processors
    - Use I/O nodes and MPI to gather/scatter file data
    - I/O nodes can be allocated as needed
  - Implement I/O solution for FMS
    - Return code changes to GFDL
    - Implement in MOM4 and OASIS
    - Maintain backwards compatibility

MOM4/FMS Upgrade

- FMS I/O Rules to apply
  - No single processor holds global arrays
  - Don’t use all processors in I/O to disk
  - Interconnects are faster than Storage I/O bandwidth
- FMS I/O Design Changes
  - Designate I/O processors/nodes
  - Assign groups of computational processors to I/O processors
  - Use parallel NetCDF among I/O processors
  - Designate MPI I/O communication groups

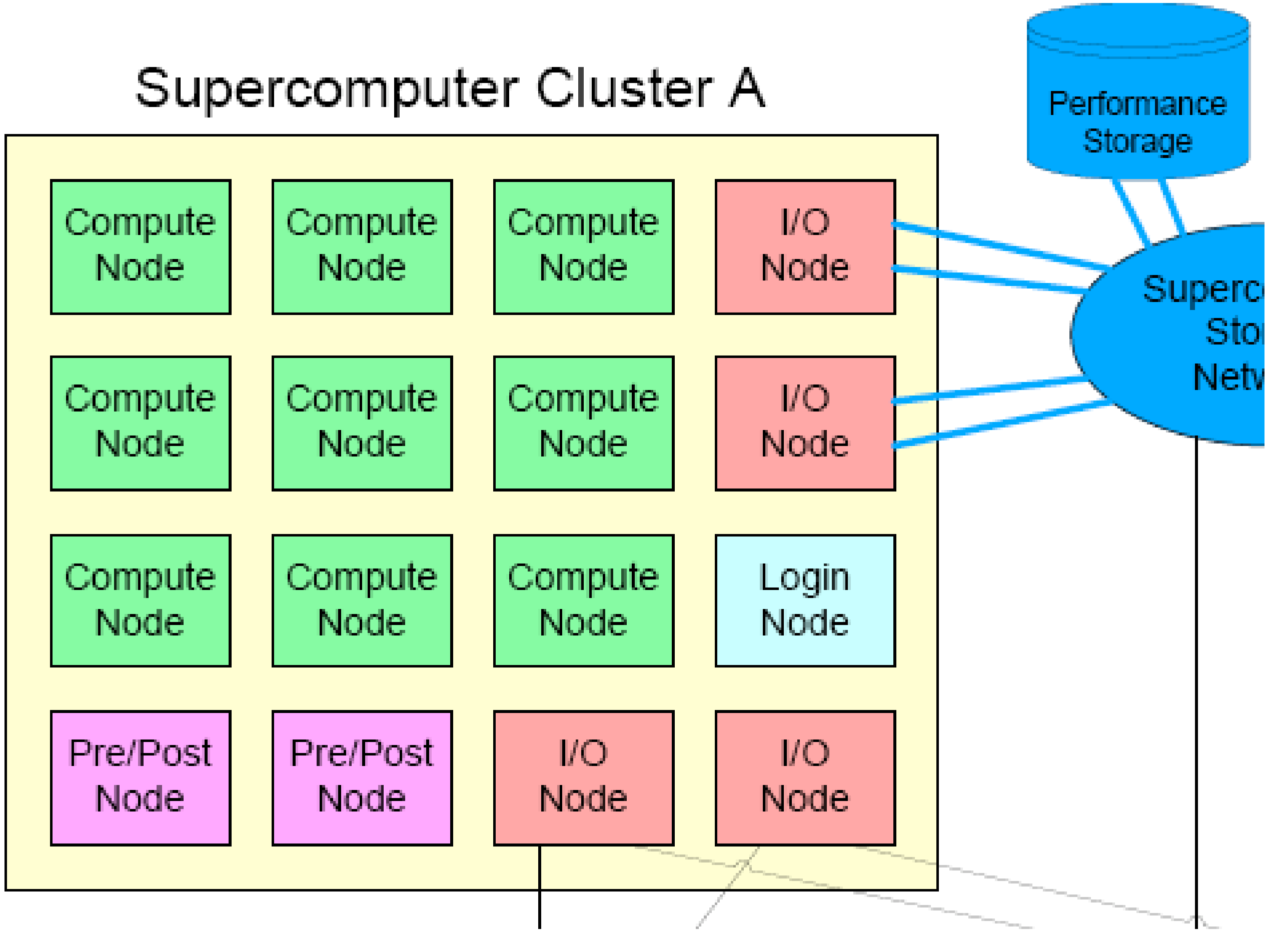


Storage

I/O CPU

Computations

Example architecture with specific service nodes



FMS Namelist Sample

- old FMS namelist

```
&fms_io_nml
    threading_read='multi'
    threading_write='single’
    fileset_write='single' /
```

- new FMS namelist

```
&fms_io_nml
    threading_read=’fabric'
    threading_write=’fabric’
    fileset_write=’fabric'
    io_threads = 3 /
```