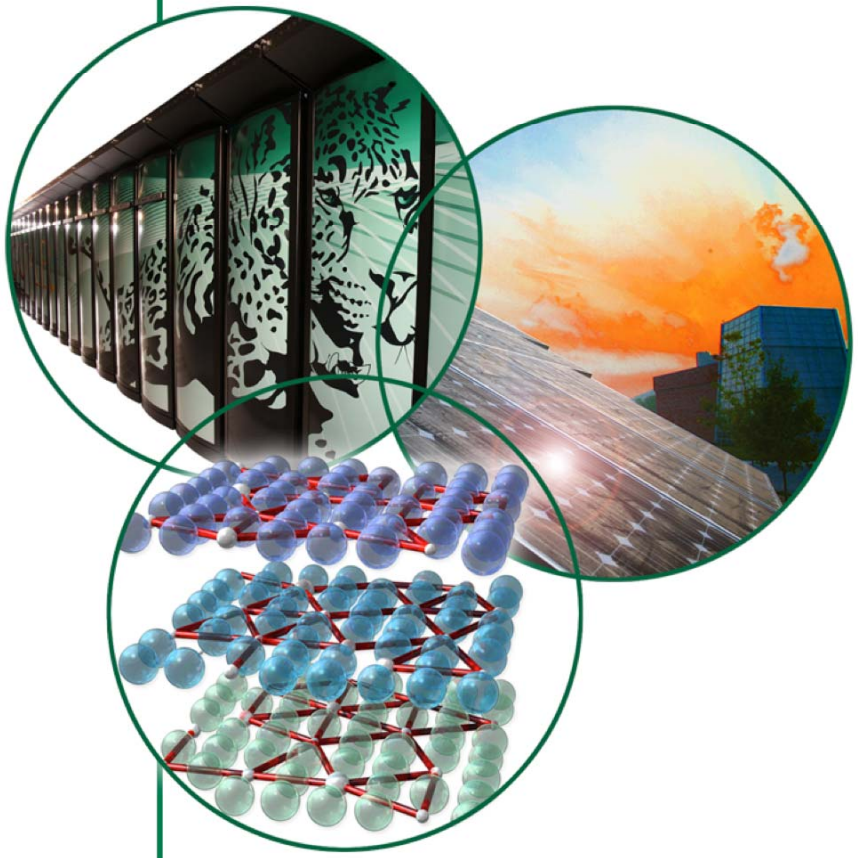
Scalable Tool Infrastructure for the Cray XT Using Tree-Based Overlay Networks

Philip C. Roth
Computer Science and Mathematics Division, Oak Ridge National Laboratory

Jeffrey S. Vetter
Computer Science and Mathematics Division, Oak Ridge National Laboratory
College of Computing, Georgia Institute of Technology



Future Technologies Group ornl

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Motivation

- Leadership class resources like Jaguar Cray XT at Oak Ridge National Laboratory (ORNL) are scarce, so allocation is valuable
- Systems growing larger and more complex
- Tools are critical for making good use of such systems
  - Debuggers
  - Performance, especially on-line automated tools
  - System administration

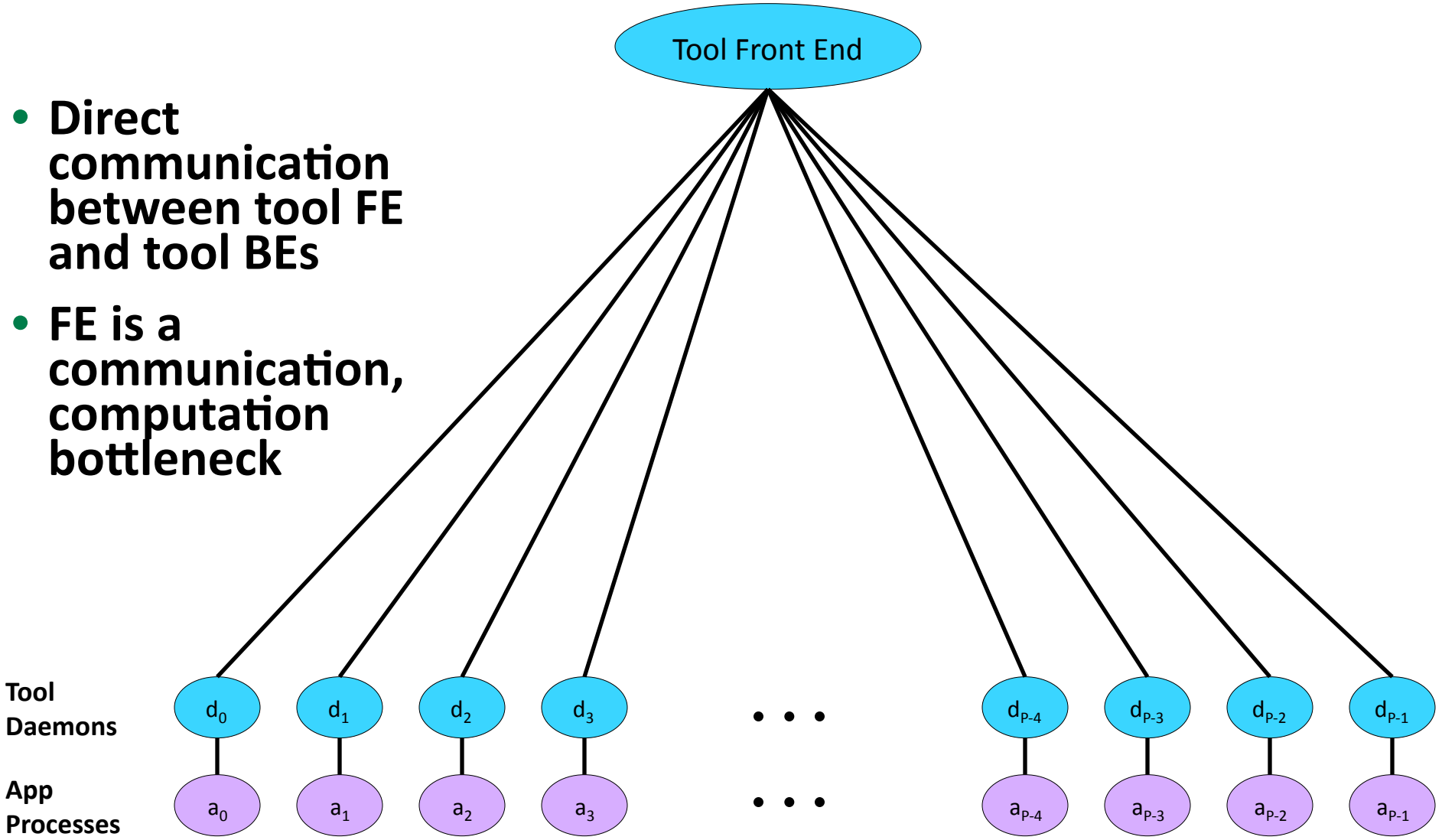
- Tools must scale at least as well as {application,system} under study

Barriers to Tool Scalability

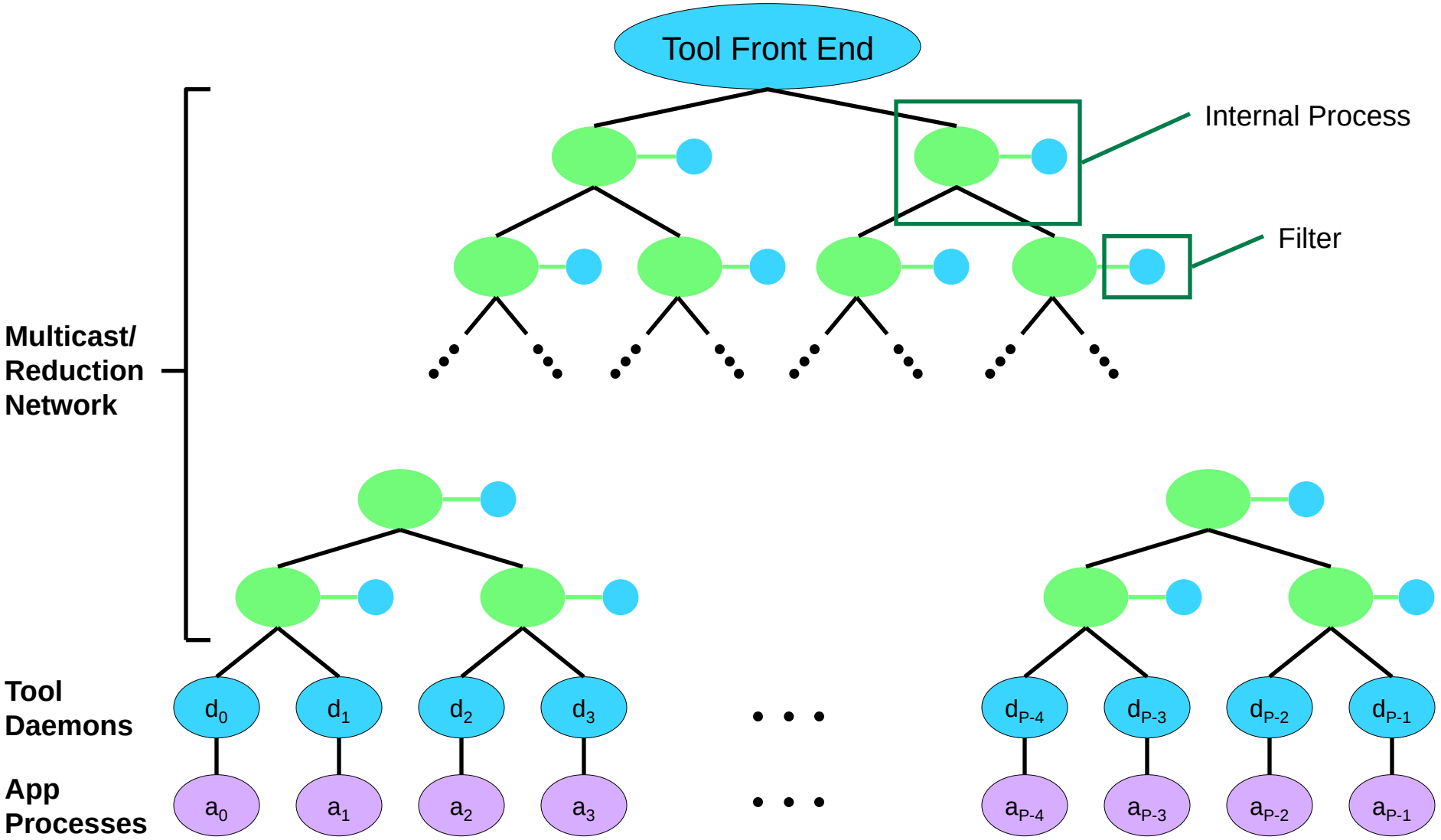
- **Managing performance data volume (collection *and* processing)**
- **Communicating efficiently between distributed tool components**
- **Making scalable presentations of performance analysis results**

Traditional Parallel Tool Organization

- **Direct communication between tool FE and tool BEs**
- **FE is a communication, computation bottleneck**



Tree-Based Overlay Networks



MRNet

- Implementation of Tree-Based Overlay Network concept
- Supports scalable multicast and data reduction operations
  - Data transferred over streams
  - Filters associated with streams manipulate data passing across network
- Integrated in Paradyn automated performance tool (University of Wisconsin-Madison)
- Used by Stack Trace Analysis Tool (STAT)
- Used as runtime for programming model for data intensive applications

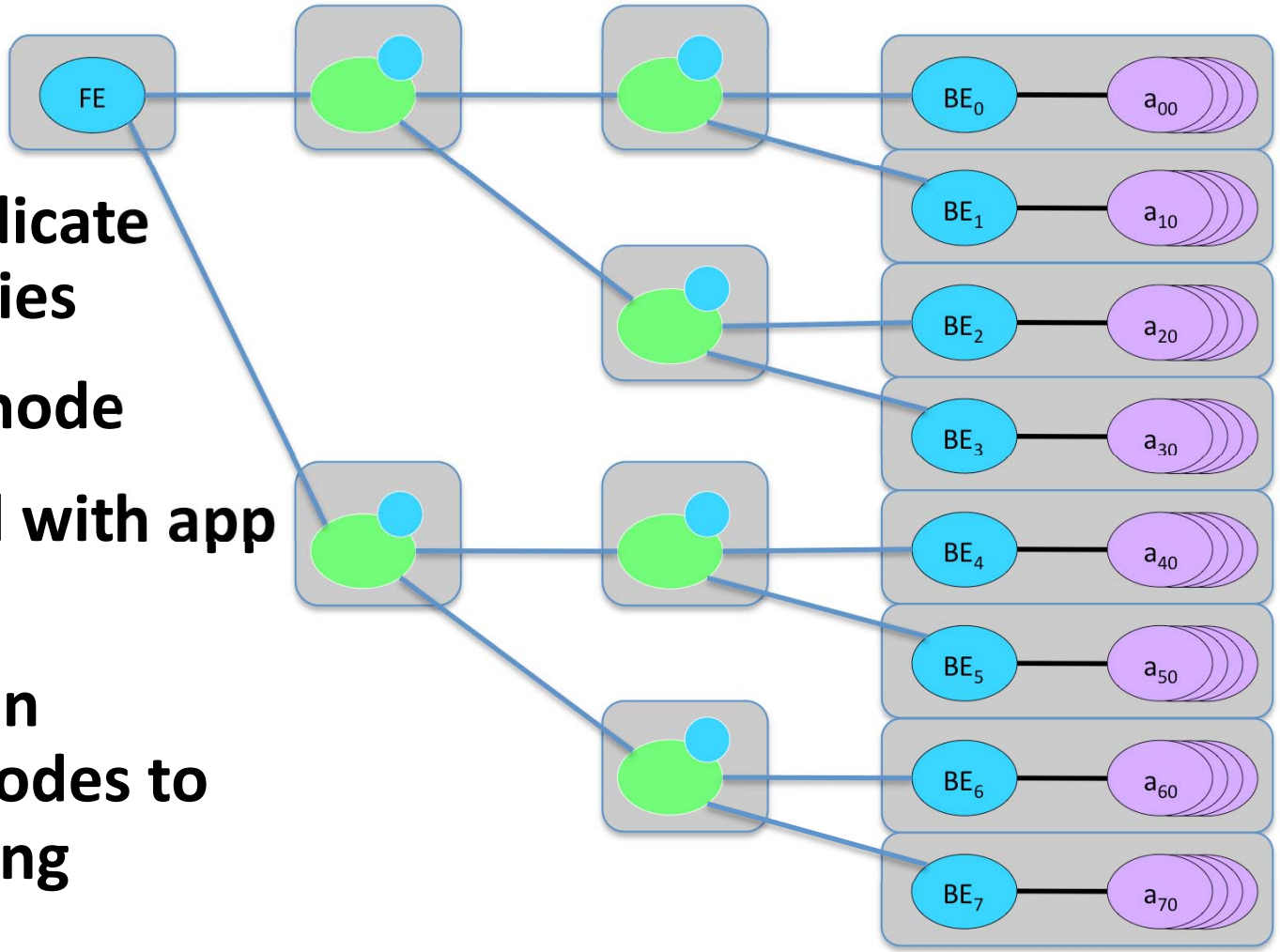
Porting MRNet to Cray XT

- Catamount
  - No server side TCP/IP sockets
  - No information about tool support library
  - Too many barriers

- Compute Node Linux/Cray Linux Environment
  - More straightforward port from Linux cluster implementation
  - Differences mainly during process network instantiation
    - Process creation
    - Process connection
  - Users requested support for new “flattened tree” process placement

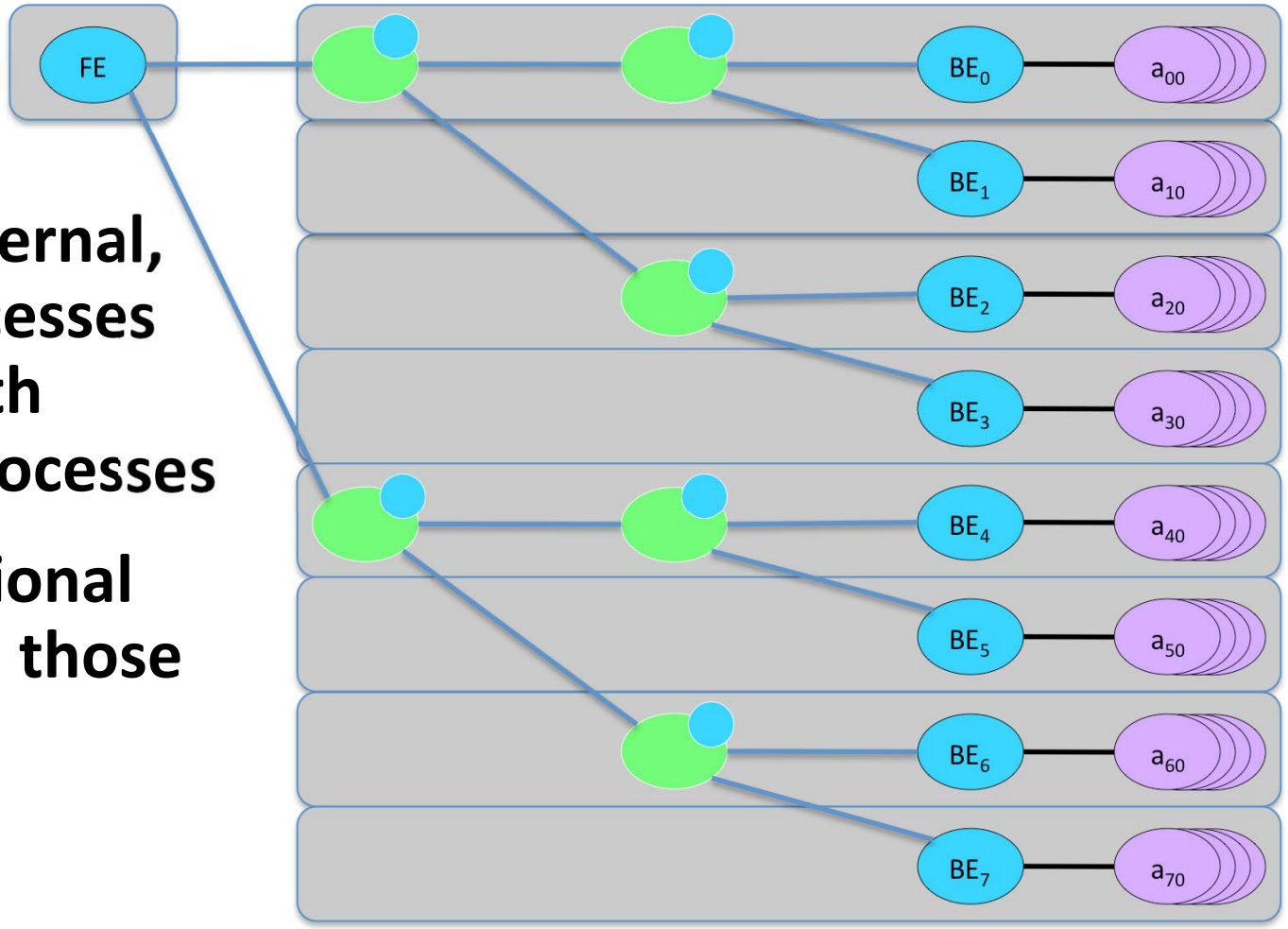
MRNet/XT Process Placement: Traditional

- Gray boxes indicate node boundaries
- FE on service node
- BEs co-located with app processes
- IN processes on "additional" nodes to avoid perturbing application



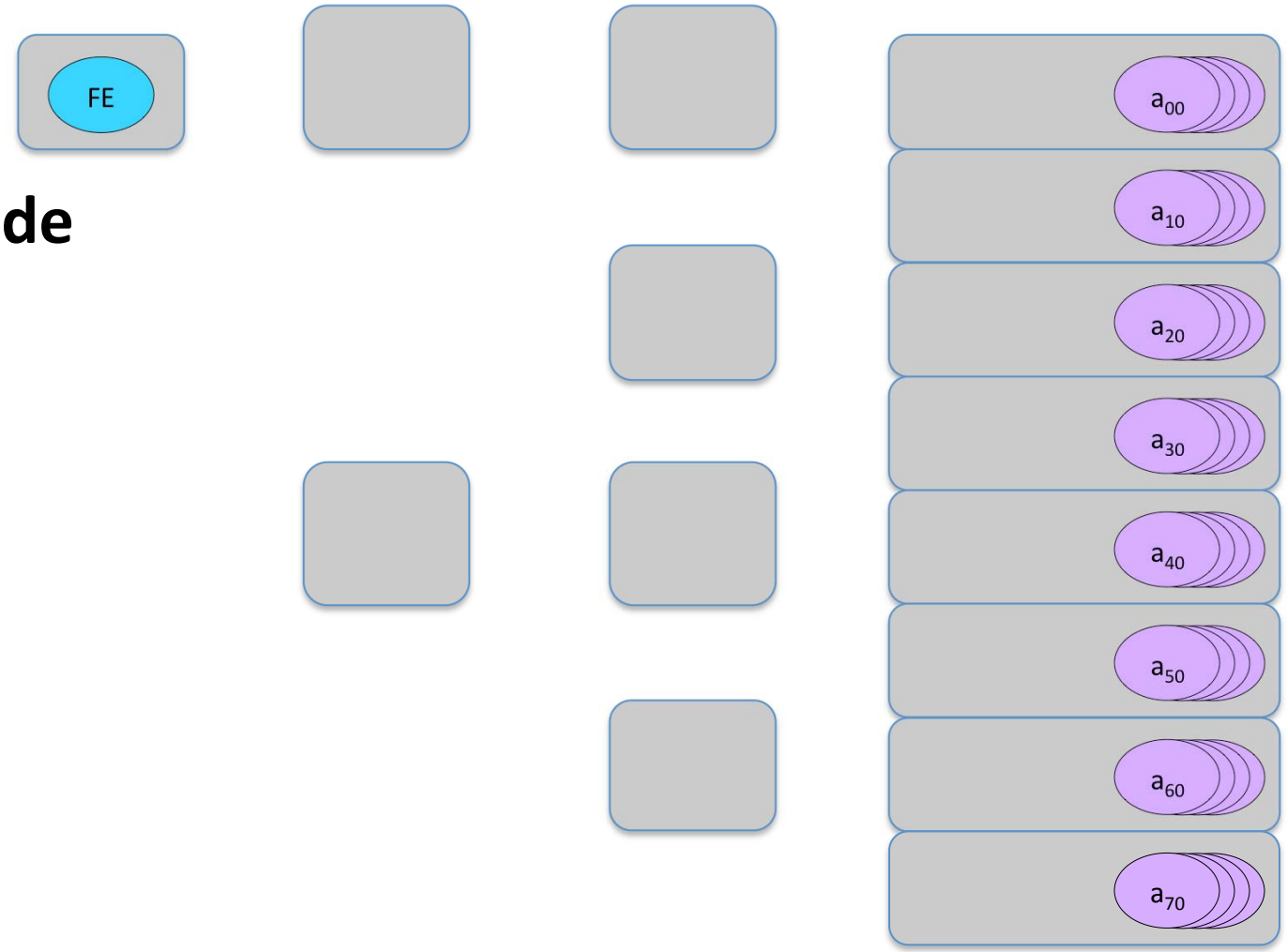
MRNet/XT Process Placement: Flattened

- MRNet/XT internal, back-end processes co-located with application processes
- Uses no additional nodes beyond those used by the application



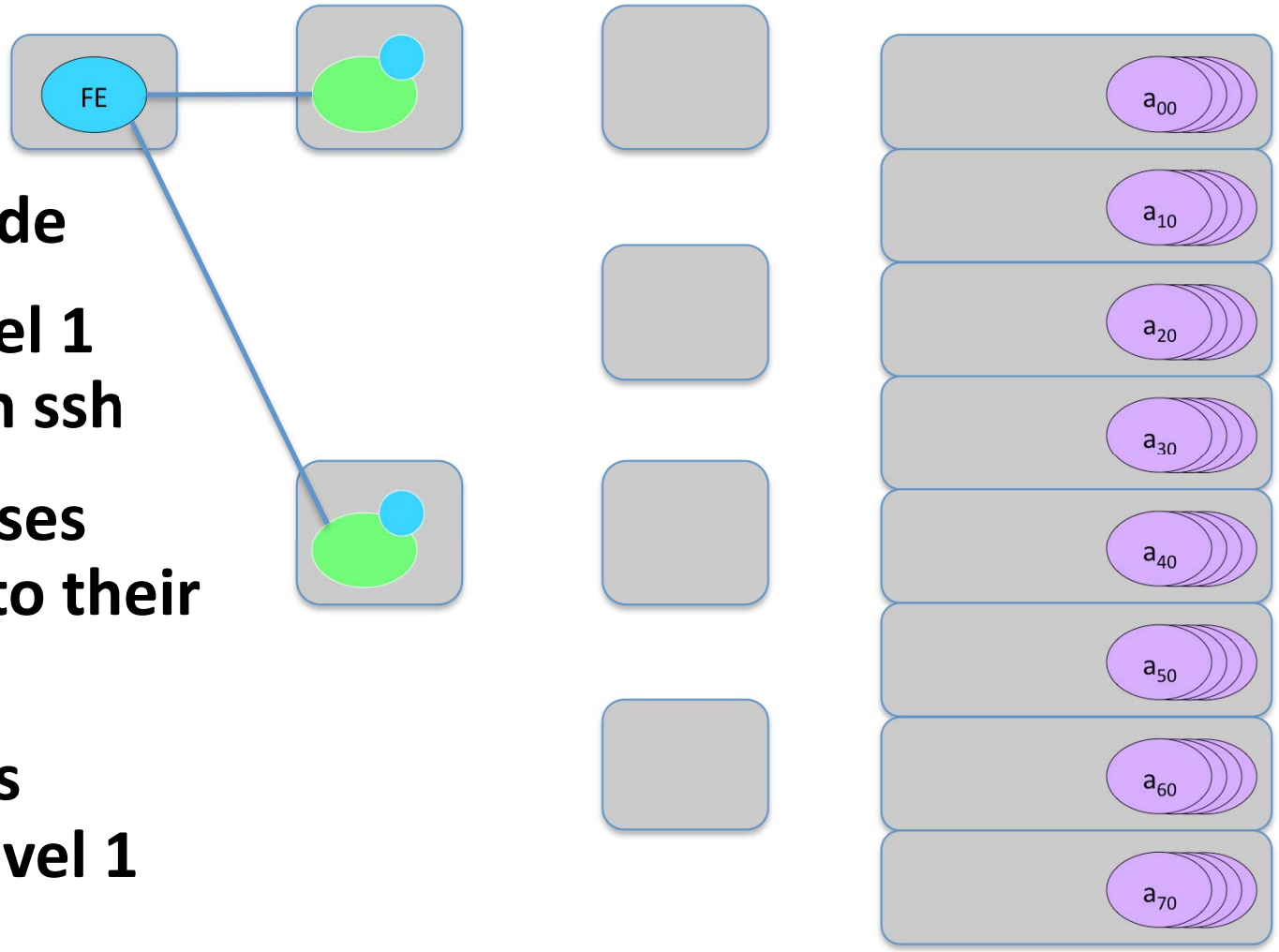
Traditional MRNet Instantiation

- **FE on login node**



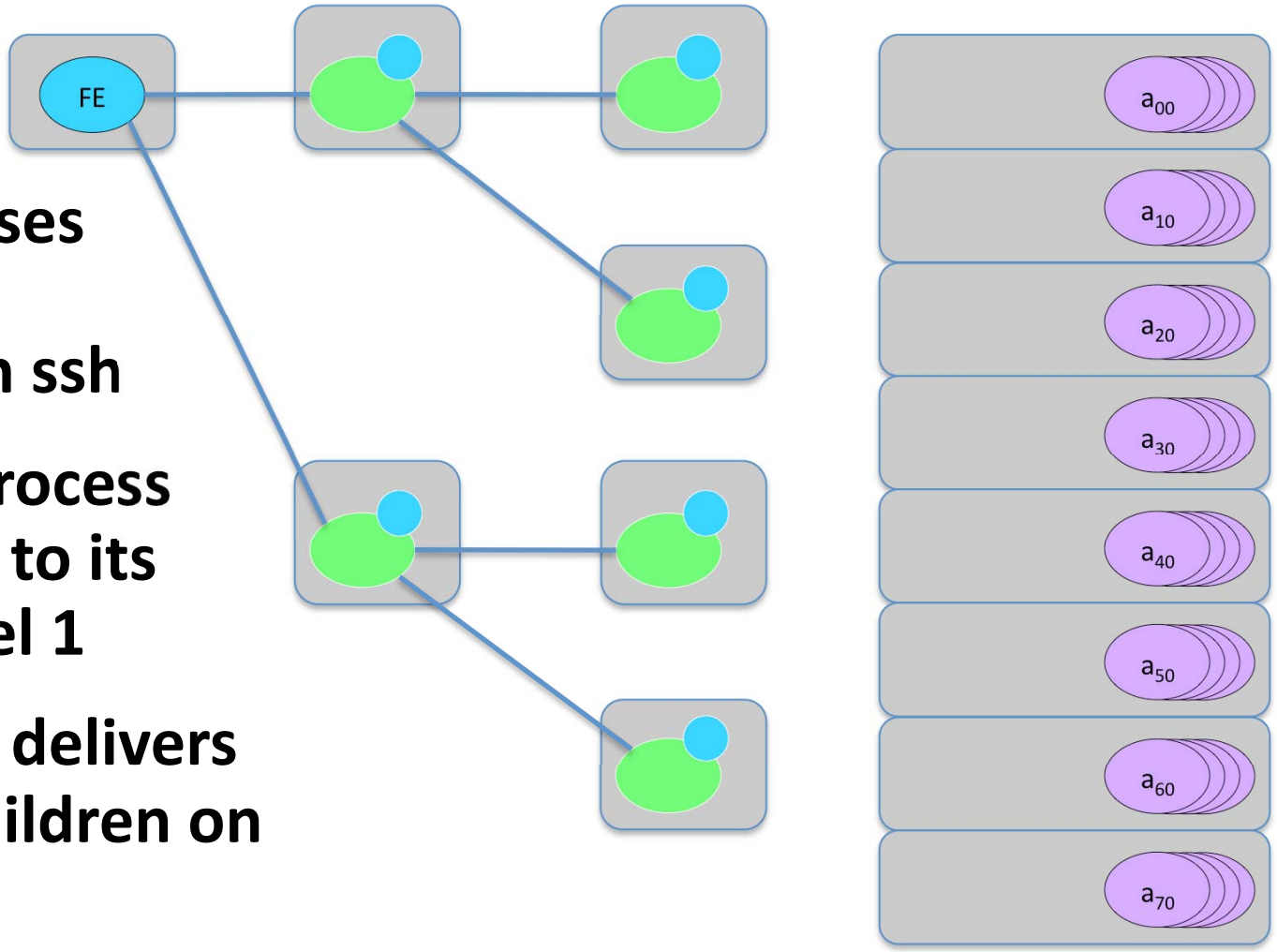
Traditional MRNet Instantiation

- FE on login node
- FE creates Level 1 processes with ssh
- Level 1 processes connect back to their parent
- Parent delivers topology to Level 1 processes



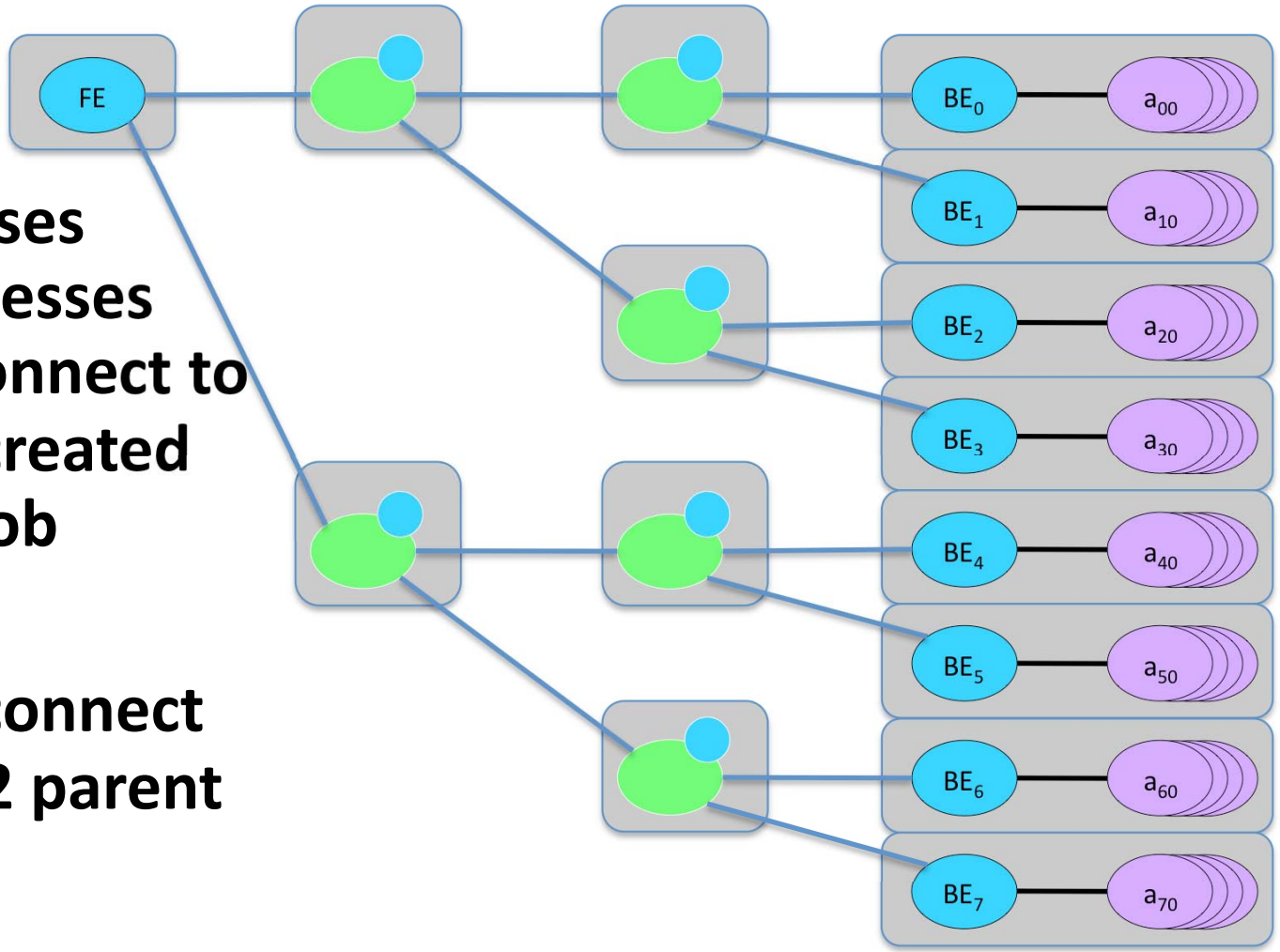
Traditional MRNet Instantiation

- Level 1 processes create Level 2 processes with ssh
- Each Level 2 process connects back to its parent on Level 1
- Level 1 parent delivers topology to children on Level 2



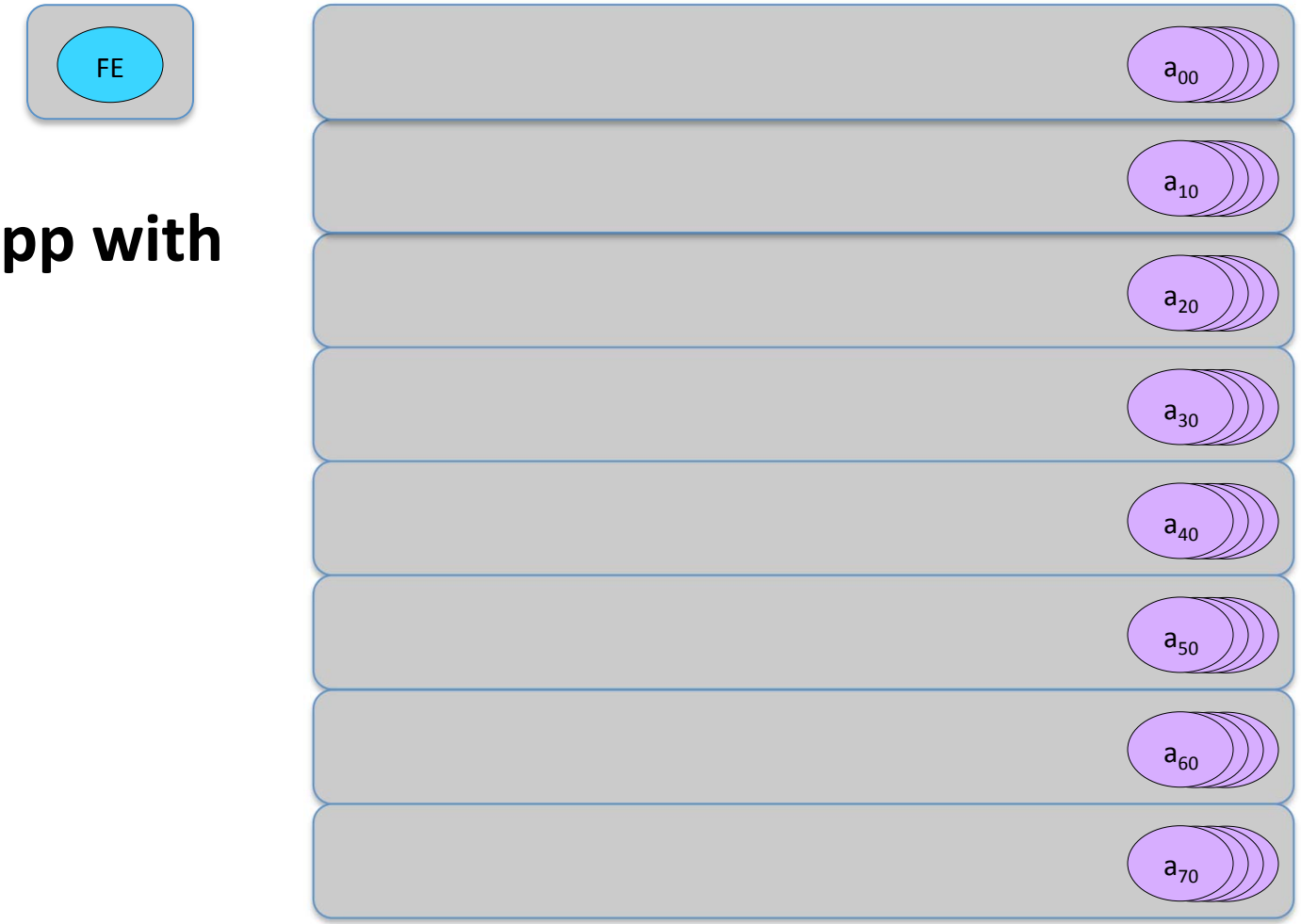
Traditional MRNet Instantiation

- Level 2 processes create BE processes with ssh (or connect to BE processes created with parallel job launcher)
- BE processes connect back to Level 2 parent



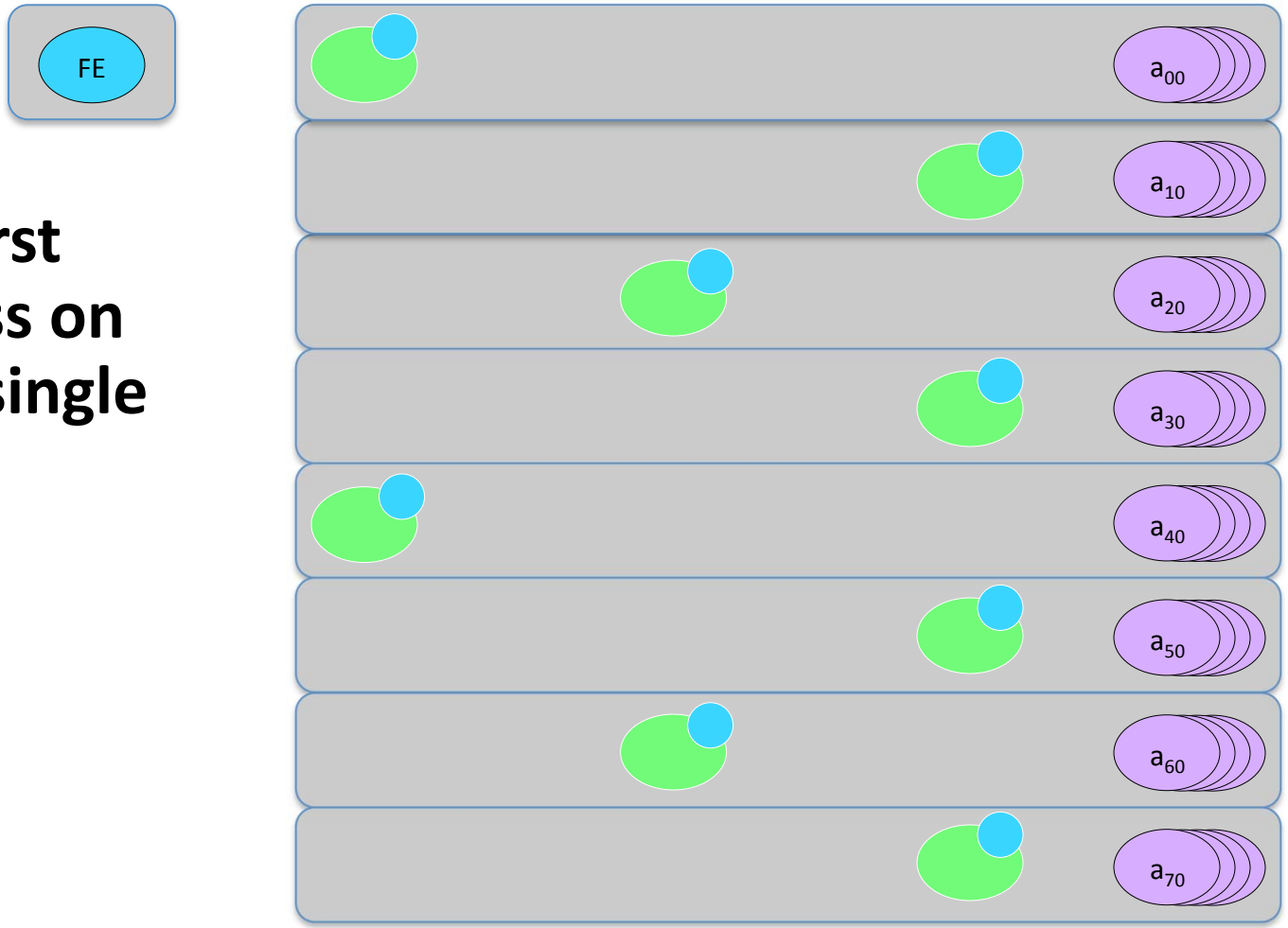
MRNet/XT Instantiation (flattened)

- **FE launches app with aprun**



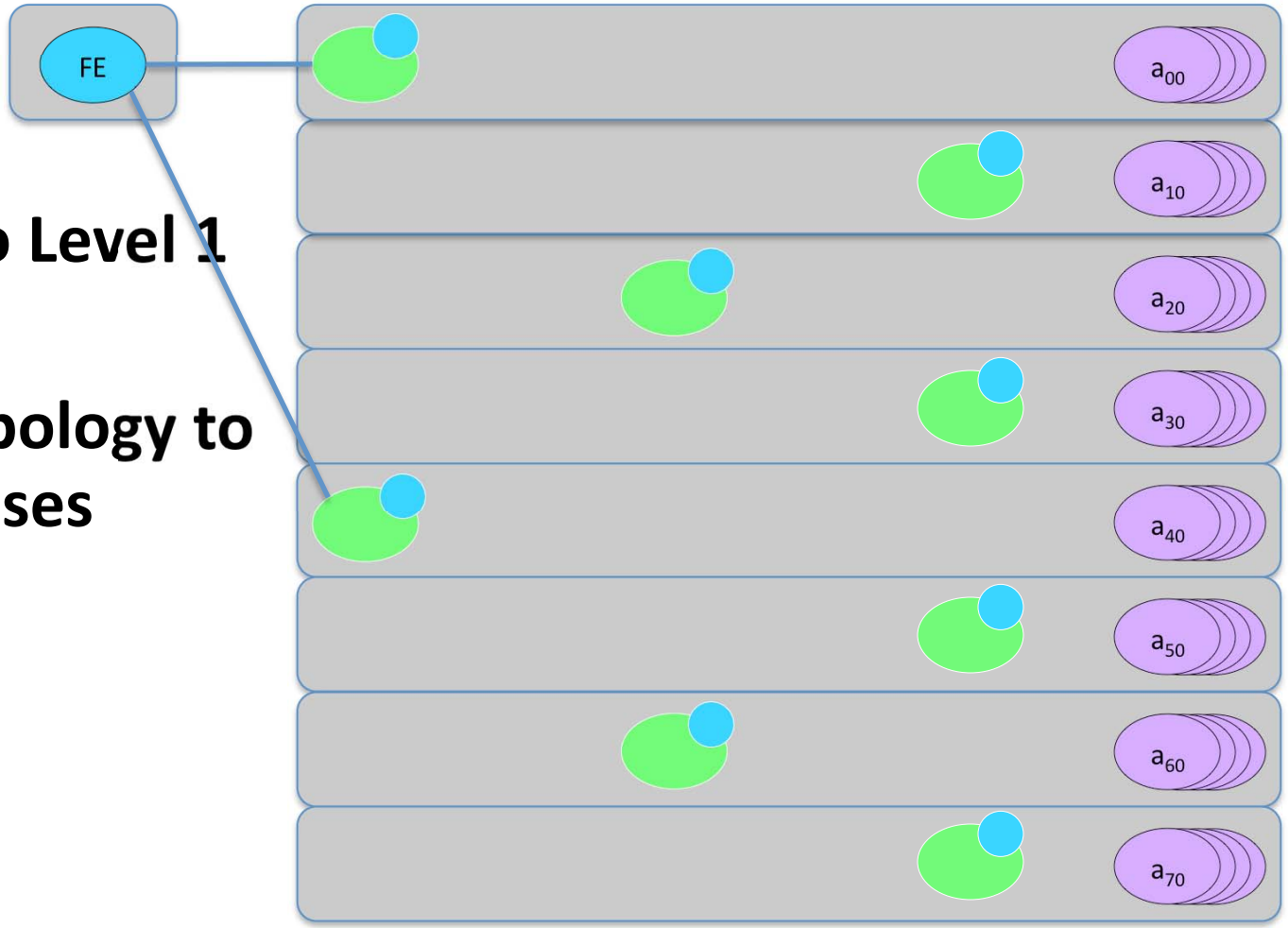
MRNet/XT Instantiation (flattened)

- **FE launches first MRNet process on each node in single operation**



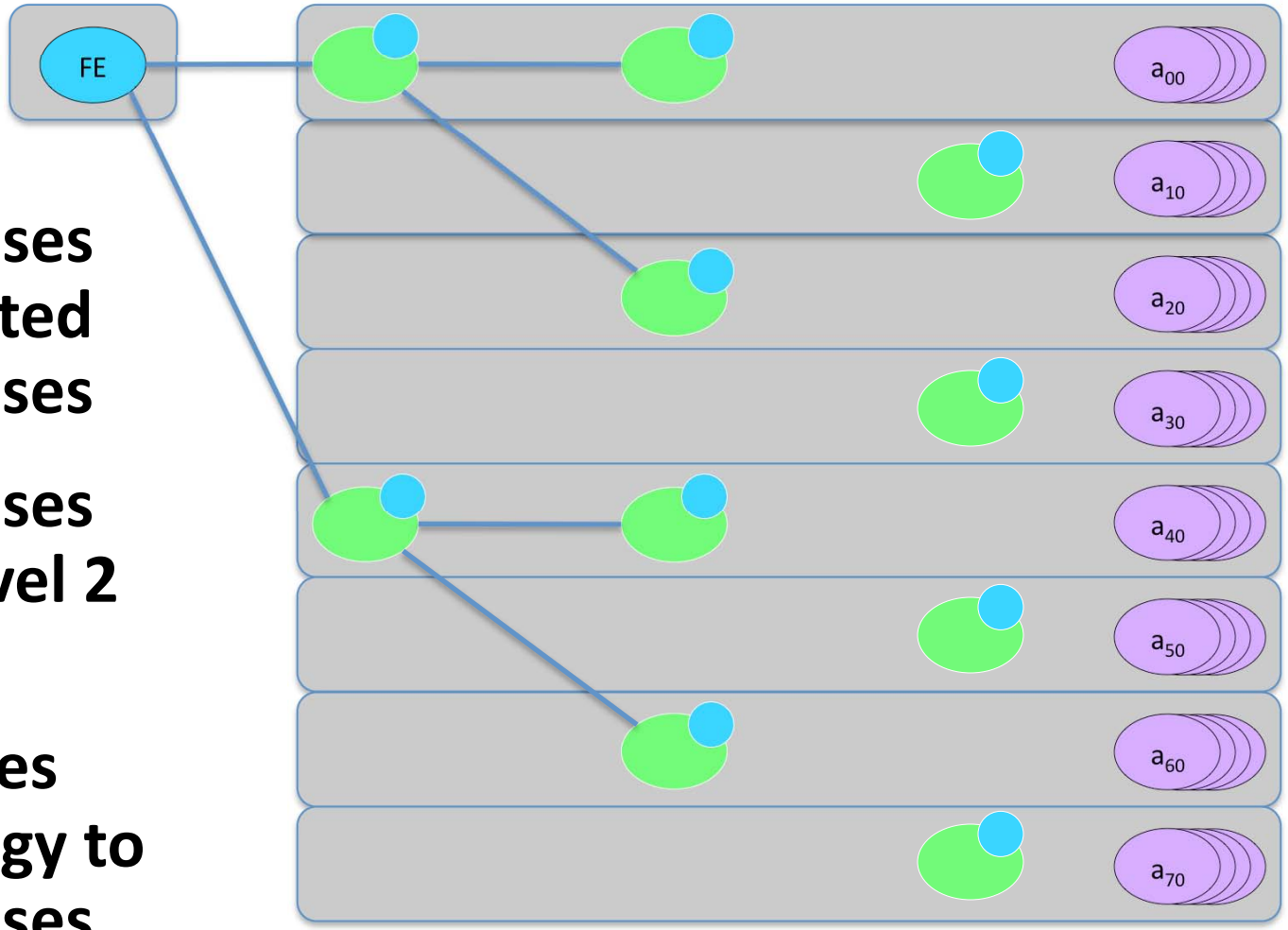
MRNet/XT Instantiation (flattened)

- FE connects to Level 1 children
- FE delivers topology to Level 1 processes



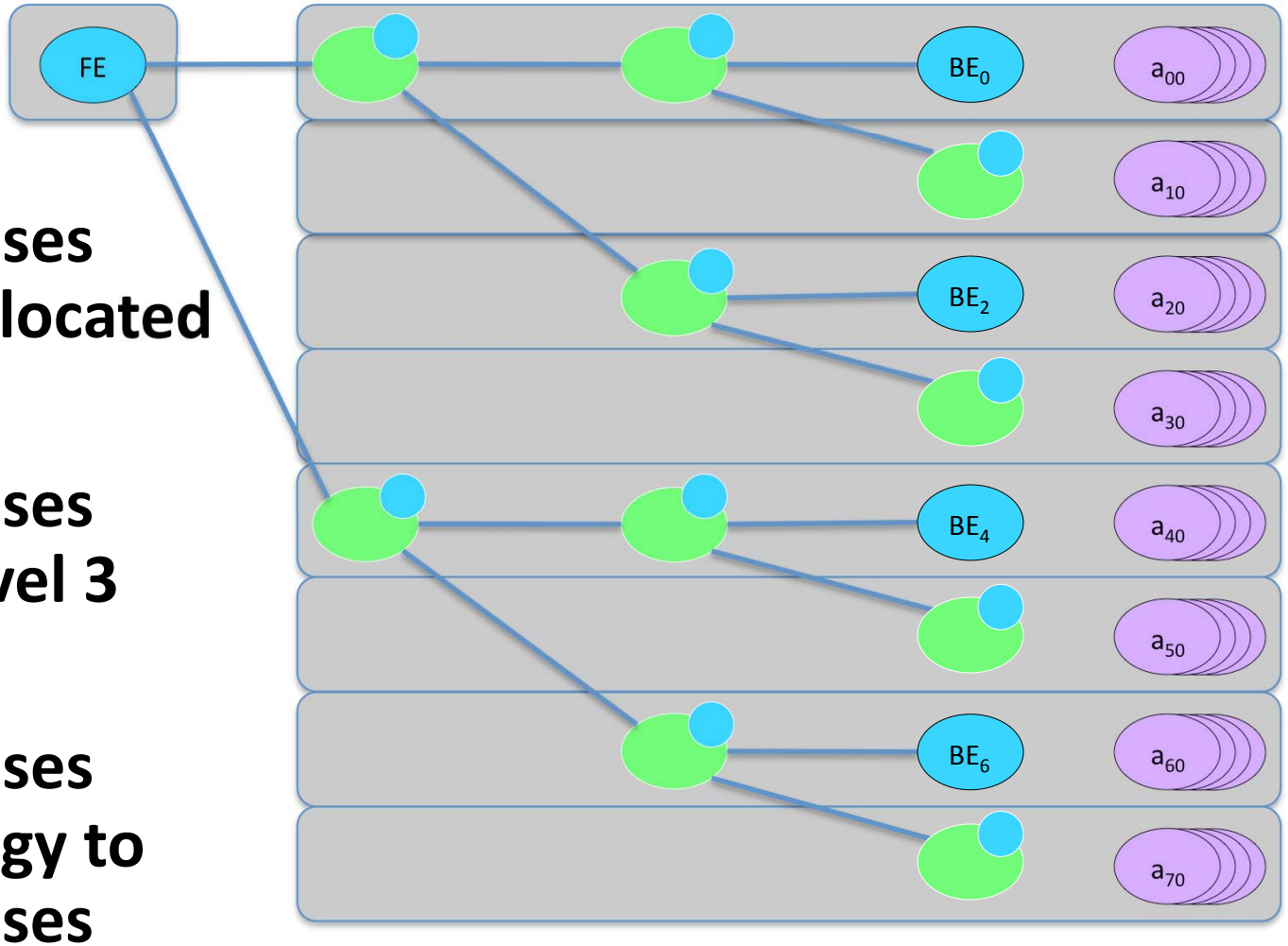
MRNet/XT Instantiation (flattened)

- Level 1 processes create co-located Level 2 processes
- Level 1 processes connect to Level 2 processes
- Level processes deliver topology to Level 2 processes



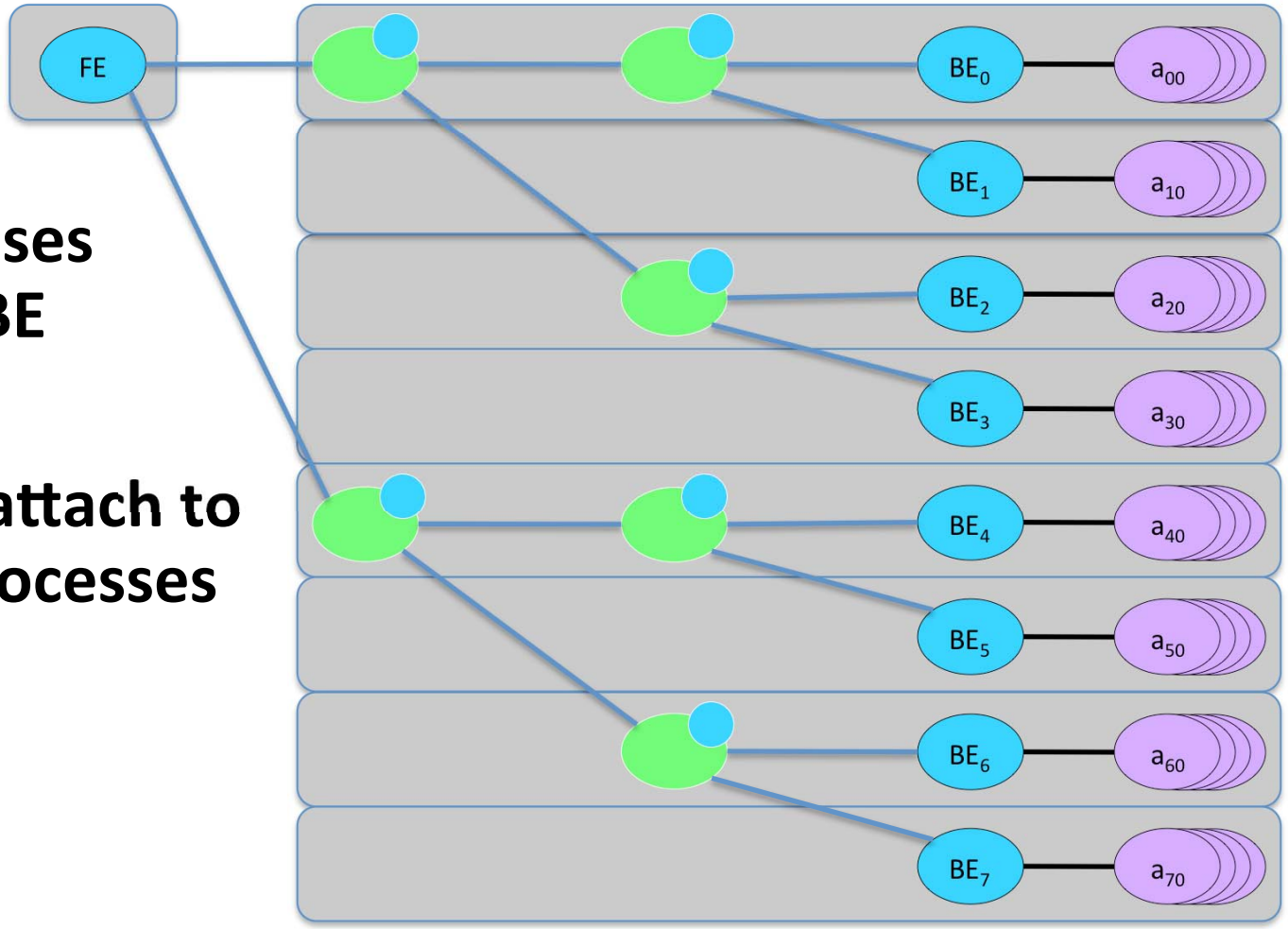
MRNet/XT Instantiation (flattened)

- Level 2 processes create any co-located BE processes
- Level 2 processes connect to Level 3 processes
- Level 2 processes deliver topology to Level 3 processes



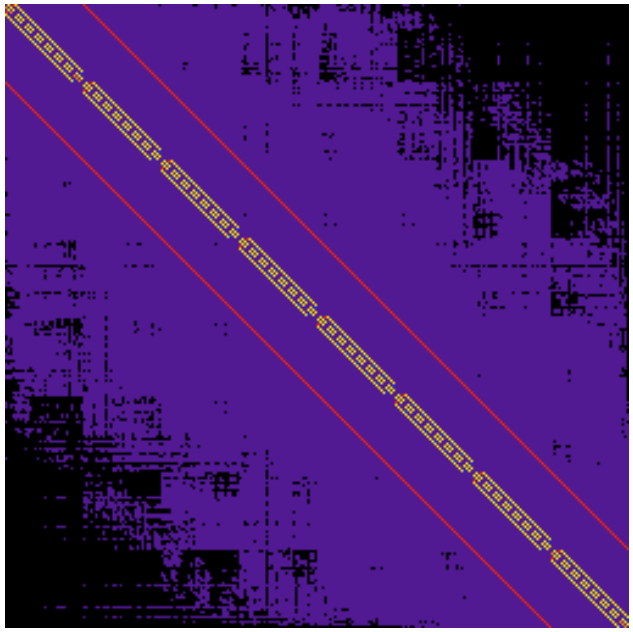
MRNet/XT Instantiation (flattened)

- **Level 3 processes become tool BE processes**
- **BE processes attach to application processes**



Example MRNet/XT tool: mpiP

- Lightweight profiling library for MPI programs
- Collects statistics about each MPI call site, e.g.:
  - Maximum message size
  - Average operation latency
- Collects data with instrumented functions at PMPI interface
- Now collects point-to-point communication topology
- Aggregates statistics and communication topology when generating reports



Communication topology matrix visualization
AMG2000 from ASC Sequoia Benchmark Suite
256 processes on a Cray XT4 at ORNL

MRNet/XT mpiP

- Traditional mpiP uses MPI point-to-point operations to aggregate data
- Investigating implementation of mpiP aggregation using MRNet/XT
  - Filters in MRNet process tree implement aggregation
  - Inductively, tool front-end receives aggregated statistics for whole program
  - Concatenation for more efficient messaging of data that cannot be aggregated (e.g., communication topology)
- Enables xP: variant of mpiP for statistical profiling of programs using any programming model or API

Summary

- Tree-based overlay networks, and MRNet in particular, are effective scalable tool infrastructure
- We have ported MRNet to the Cray XT
- We added support for "flattened tree" MRNet topologies
- We are integrating MRNet/XT into scalable tools for Cray XT such as mpiP and variants

- Thanks to Mike Brim and Barton Miller (University of Wisconsin-Madison) and Bob Moench (Cray)
- For more information:
  - rothpc@ornl.gov
  - http://ft.ornl.gov
  - http://www.paradyn.org/mrnet (general MRNet information)

- This research is sponsored by the Office of Advanced Scientific Computing Research; U.S. Department of Energy. The work was performed at Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725. This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.