

Data Input/Output in High Performance Scientific Programs



Thomas Ludwig

Ruprecht-Karls-Universität Heidelberg

Computer Science Department

Parallel and Distributed Systems

t.ludwig@computer.org



Motivation

- Measures

 - Mega (2^{20}) – Giga (2^{30}) – Tera (2^{40}) – Peta (2^{50}) – Exa (2^{60})

 - Flop/s: floating point operations per second

 - A 2GHz Pentium processor has about 4 GFlop/s (peak)

- Supercomputers

 - Current No. 1: NEC's Earth Simulator

 - Compute power: 35 TFlop/s

 - 10 TByte memory; 700 TByte disk space

 - Main application field: climate modelling etc.

 - Future No. 1 (in 2008): IBM's Blue Gene

 - Compute power >1 PFlop/s

 - Main application field: protein folding etc.



Some Numbers

Storage performance

- A single disk makes about 50MByte/s
- To store 1 GByte takes 20 seconds
- To store 1 TByte takes 20.000 seconds (~5 hours)
 - Or 20 seconds with 1000 disks
(but where is my "file"?)

There must not be a problem with any of these disks 😊

Storage performance is a crucial factor



Outline

- Traditional vs. High Performance Input/Output (I/O)
- Parallel I/O in Programs
- Parallel I/O in the System
- Own Research Directions

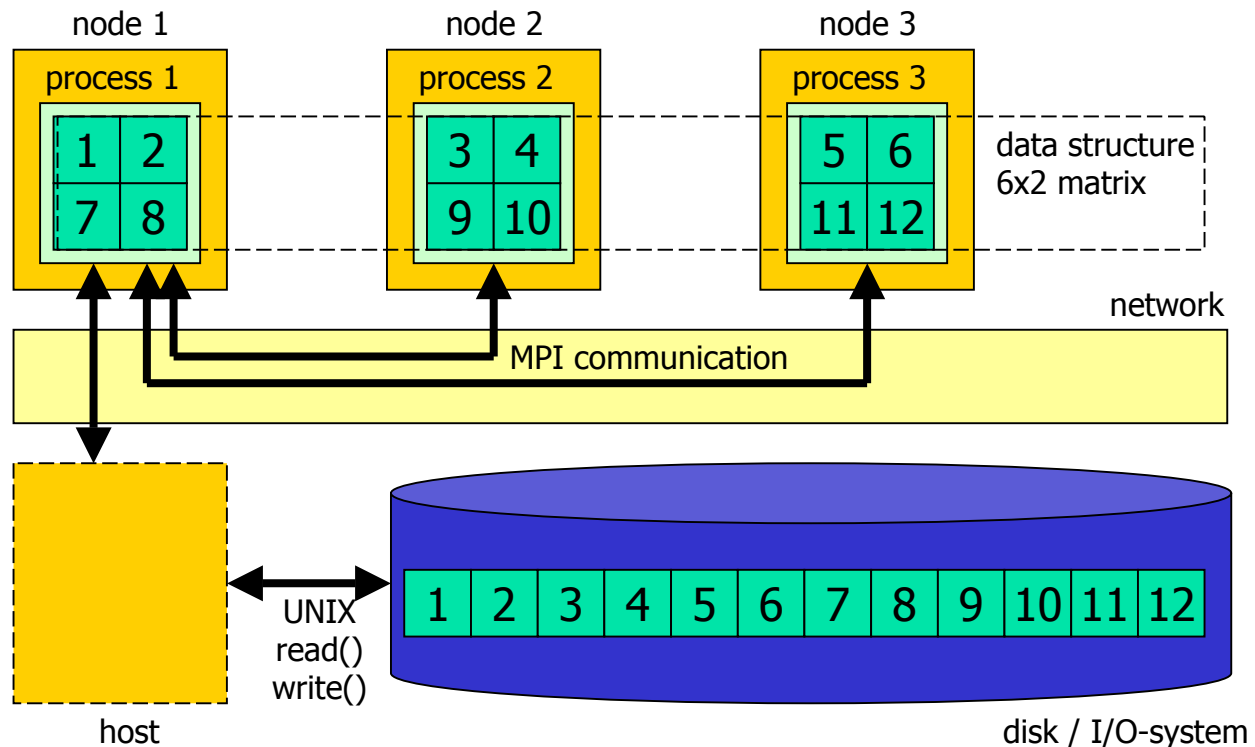


Abstraction Layers

- Program level
 - How do I access my data?
 - Single process / multiple processes
- System level
 - Where is my data?
 - Single disk / multiple disks
 - Powerful modern I/O-systems
 - RAID – Redundant Array of Inexpensive Disks
 - SAN – Storage Area Network
 - NAS – Network Attached Storage
- Concepts at program and at system level are independent of each other!

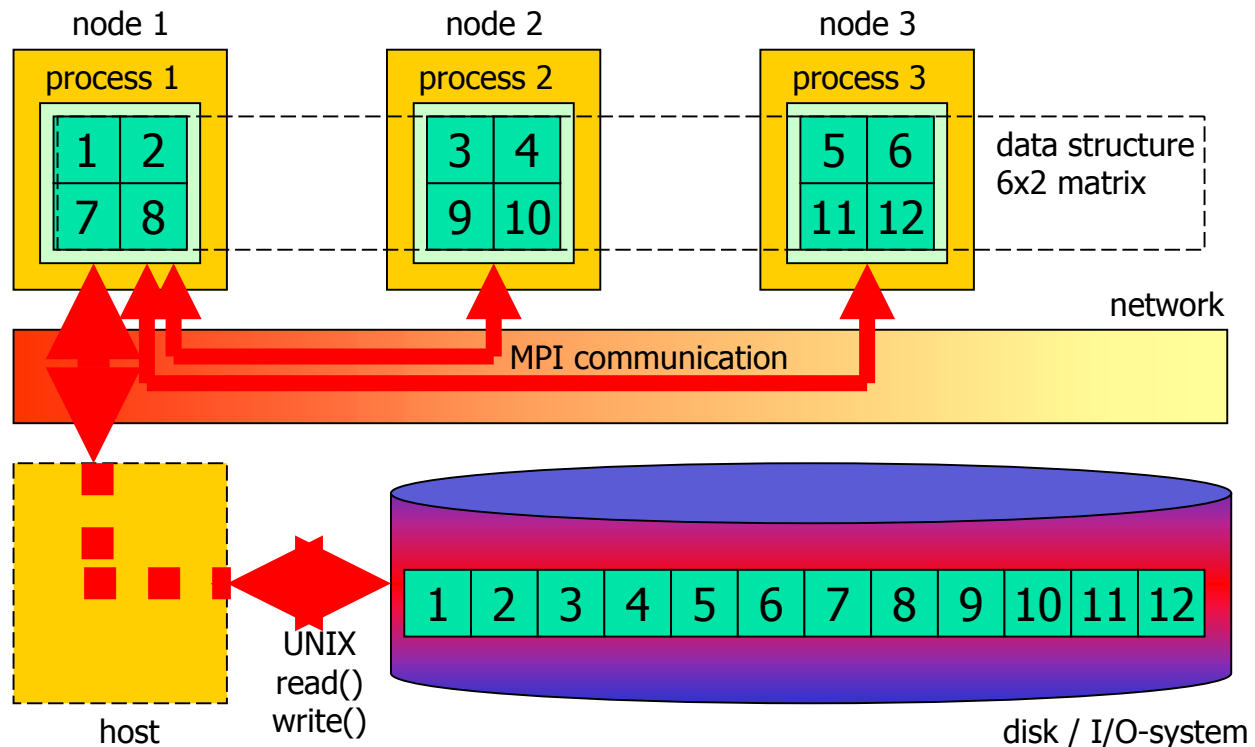
Traditional Approach

Parallel program with MPI communication



Traditional Approach: Problems

Parallel program with MPI communication





Modern I/O-Concepts

Parallelization of I/O

- At program level (Parallel file I/O)
 - Each process can directly make I/O calls
 - I/O calls from different processes may address identical files (even identical bytes)
- At system level (Parallel file system)
 - We use more than one disk
 - A file may be spread over many disks
 - The size of a file may exceed the size of a disk

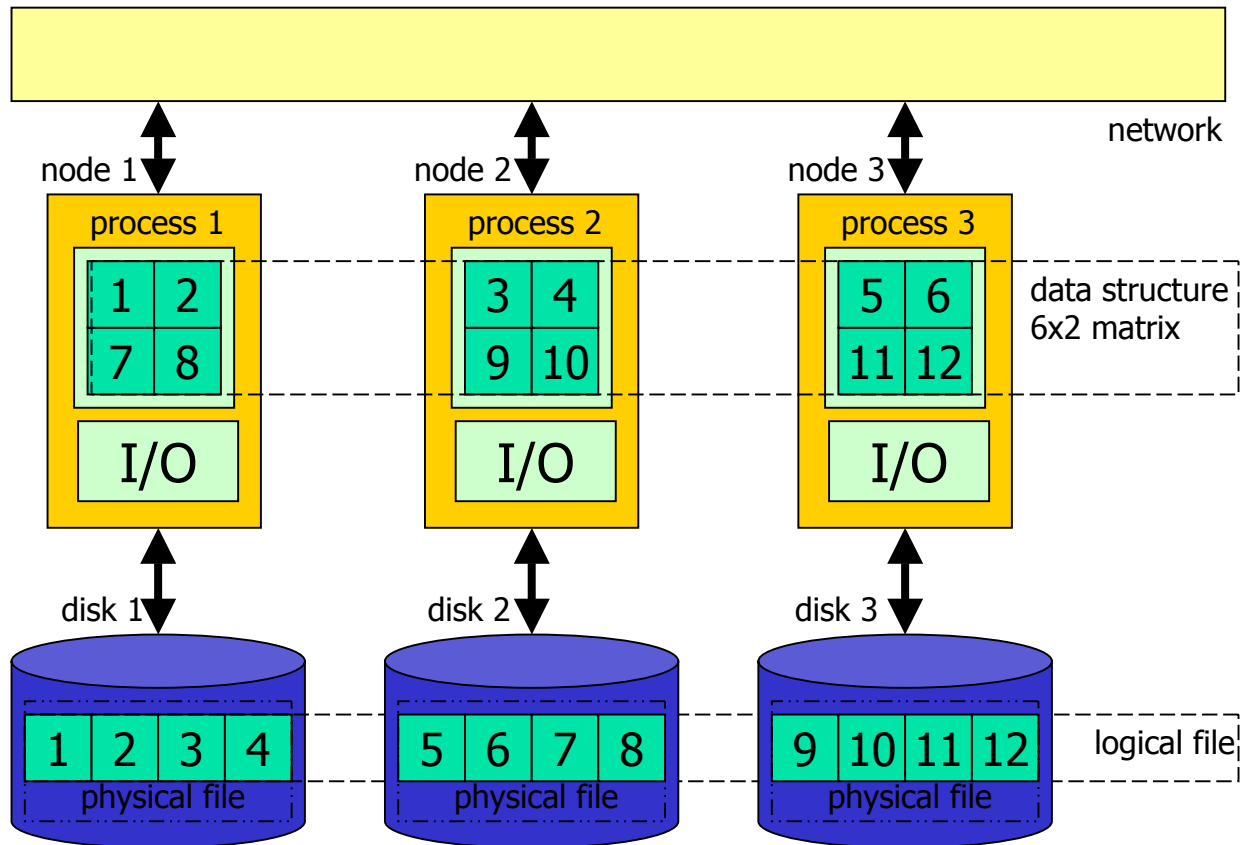
Both together is called "Parallel I/O"



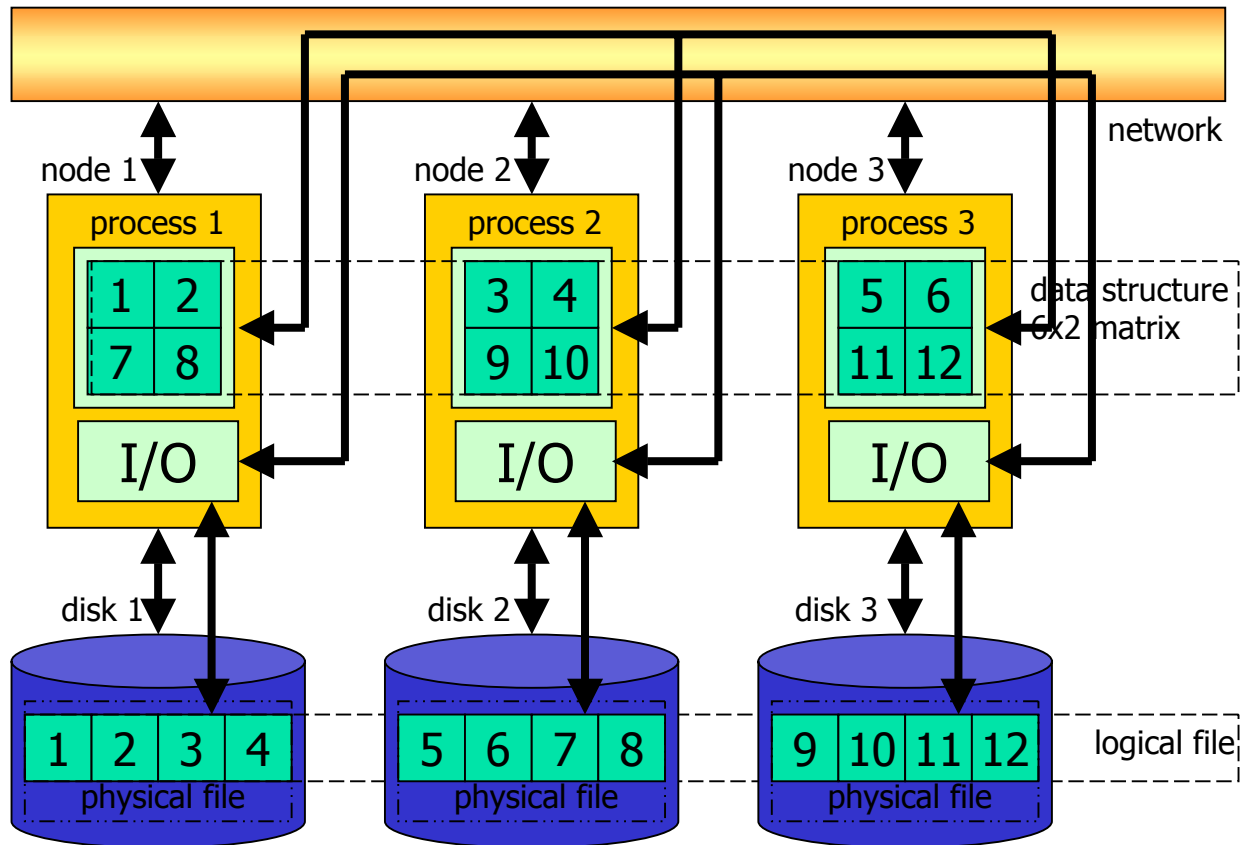
Application Classes

- Applications with parallel I/O
 - Massively parallel programs
 - Numerical simulations (biology, physics, ...)
 - Regular data structures (vectors, arrays, ...)
 - High performance computing and networking
 - Usually MPI-based (message passing interface)
- Applications with other high performance I/O concepts
 - Database applications (biology, search engines)
 - Media data applications (video, audio, ...)

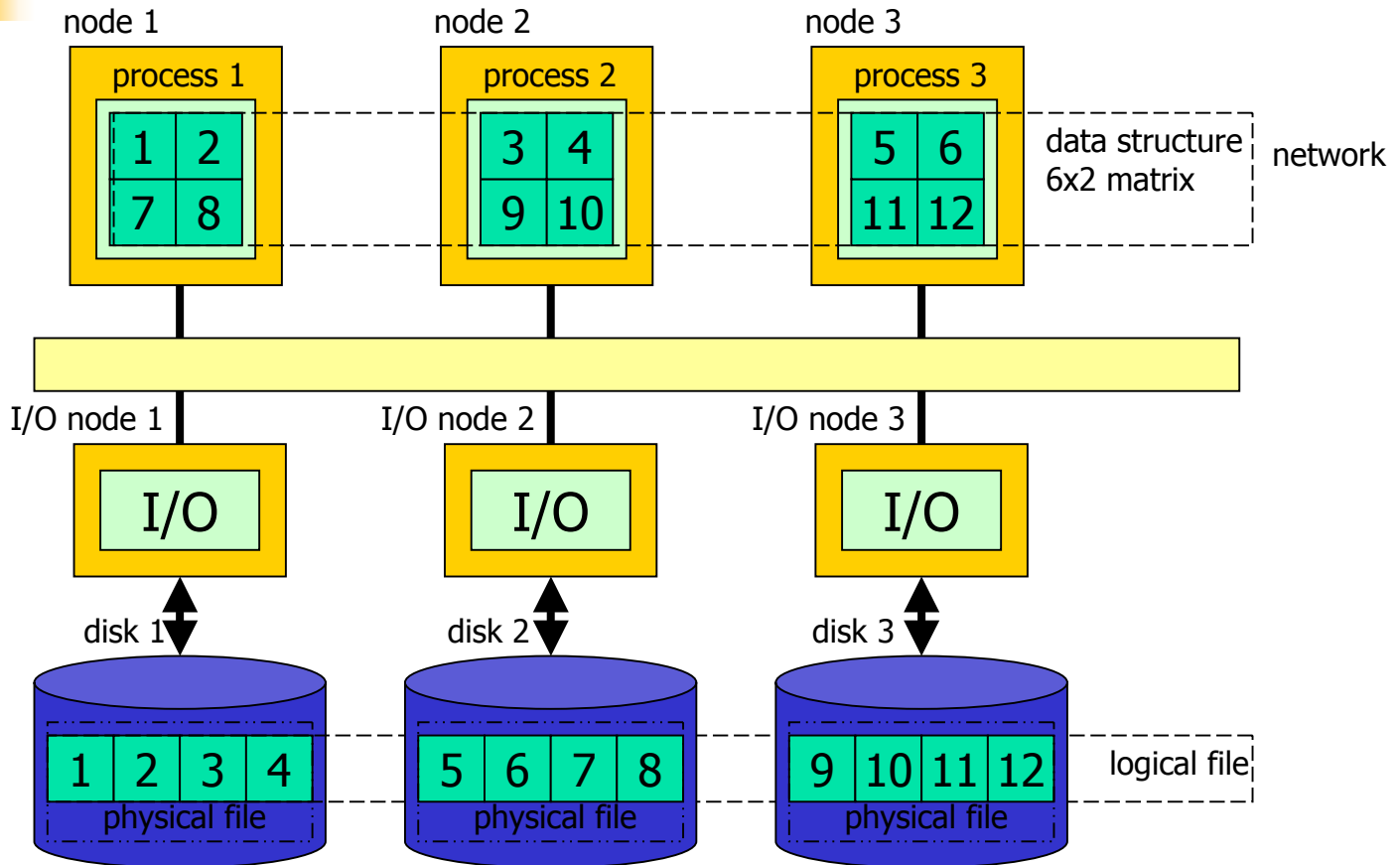
The Concept of Parallel I/O



The Concept of Parallel I/O...



The Concept of Parallel I/O...





Parallel I/O and Clusters

- Why are clusters relevant?
 - Clusters have replicated resources for computing *and* for I/O
 - Physical disk space is available
- Crucial technical issues
 - Disk reliability (fault tolerance required)
 - Metadata performance (no final solution yet)
 - File info like e.g. size must be stored in a central component: performance bottleneck



Parallel I/O in Programs

- Important categories

- Input data / Result data
- Intermediate data

Max. amount of data = size of main memory

- Other categories

- Temporary data: no parallel access needed
- Checkpoint data: no persistency needed
- Out-of-core-execution



Parallel I/O Libraries

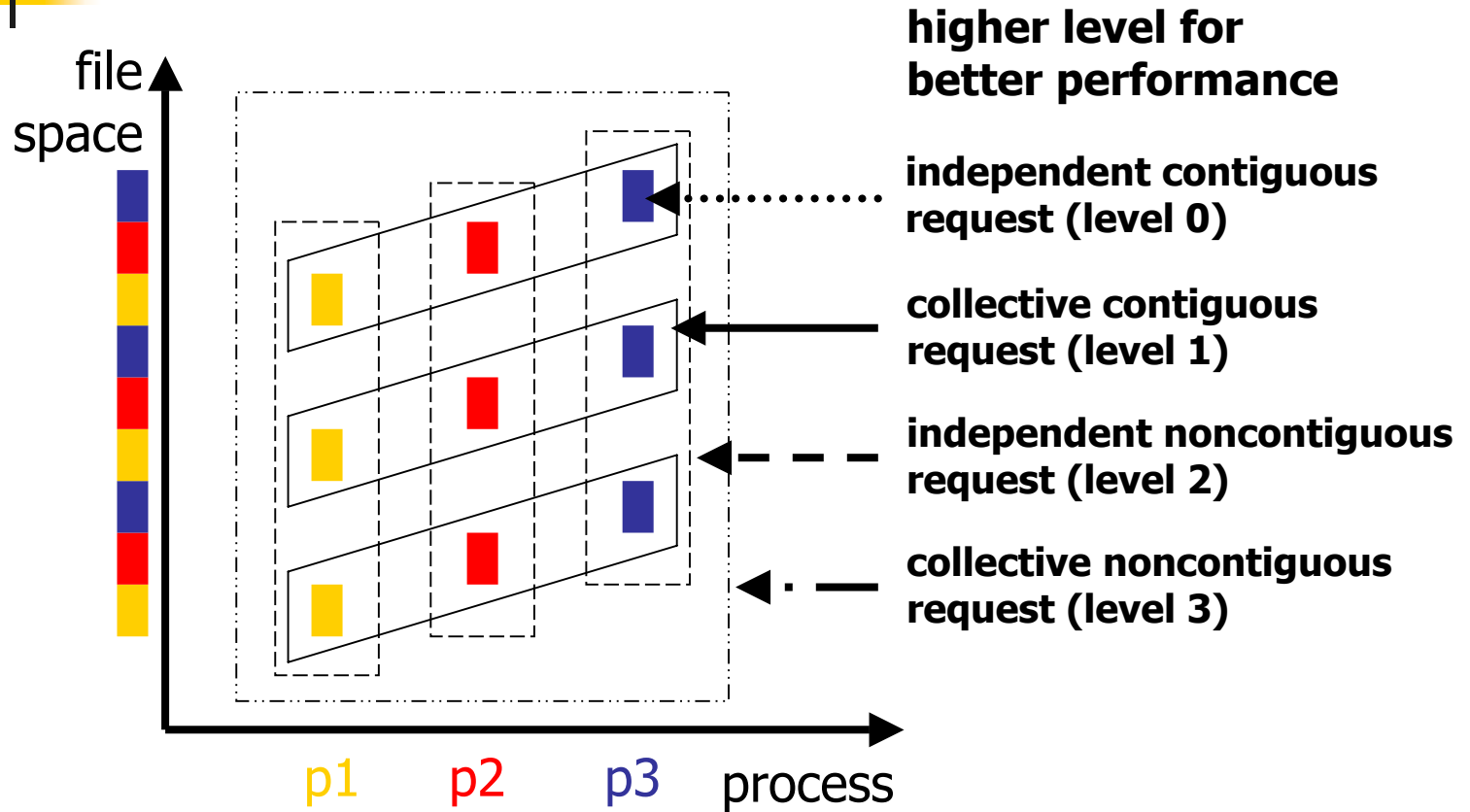
- Parallel programming with message passing on large supercomputers/clusters
 - Use send()- and receive()-routines
 - MPI is the current standard for this
- Parallel I/O: defined by MPI-IO
 - Defines read()-operation similar to receive()
 - Defines write()-operation similar to send()
 - Keeps most other semantic details of MPI message passing

MPI-IO Semantic Details

- Collective I/O
 - All processes of a given set access a file at the same time (at different positions)
 - Allows for optimization by the library
- Noncontiguous access
 - E.g. matrix is mapped to memory row by row and process reads first column
 - Library optimizes physical access

a11	a12	a13	a21	a22	a23	a13	a22	a33
-----	-----	-----	-----	-----	-----	-----	-----	-----

MPI-IO Parallel I/O Support





MPI-IO Implementation

- MPI implementation: e.g. MPICH
- MPI-IO implementation: ROMIO

- ROMIO (same group as MPICH)
 - Implements MPI-IO semantics
 - Sits on top of single disks, parallel file systems or selected high performance I/O- systems
 - Transforms MPI-IO calls to disk access requests
 - Open source / public domain



Parallel File Systems

- Parallel file system
 - Distributes a single file over several disks
- Not too many systems available
- A selection
 - GPFS (IBM): older proprietary approach
 - PVFS: current popular open source system
 - Lustre: future open source system



PVFS2 – Parallel Virtual File System Version 2

- PVFS2 by Argonne National Labs and Clemson University
- Flexible configuration of compute nodes and I/O nodes
- Many technically advanced features
- Does not yet solve the metadata server bottleneck problem

- Open source software
- Used for parallel I/O research



Own Research

- Deployment and optimization of parallel I/O in cluster environments
 - Performance measurement
 - Load balancing
 - Deployment in production environments
 - Access pattern classification and detection

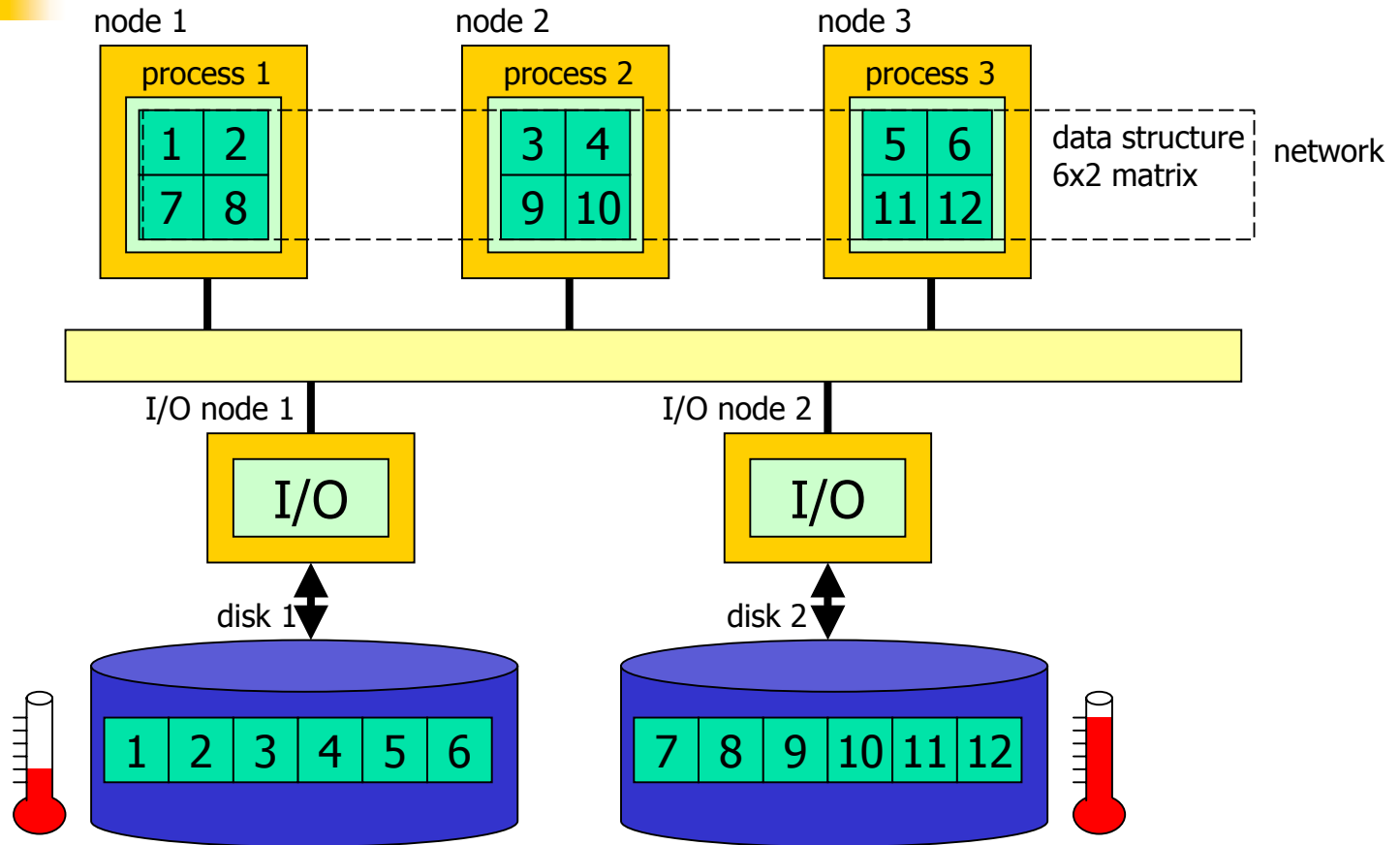
- Users are welcome 😊



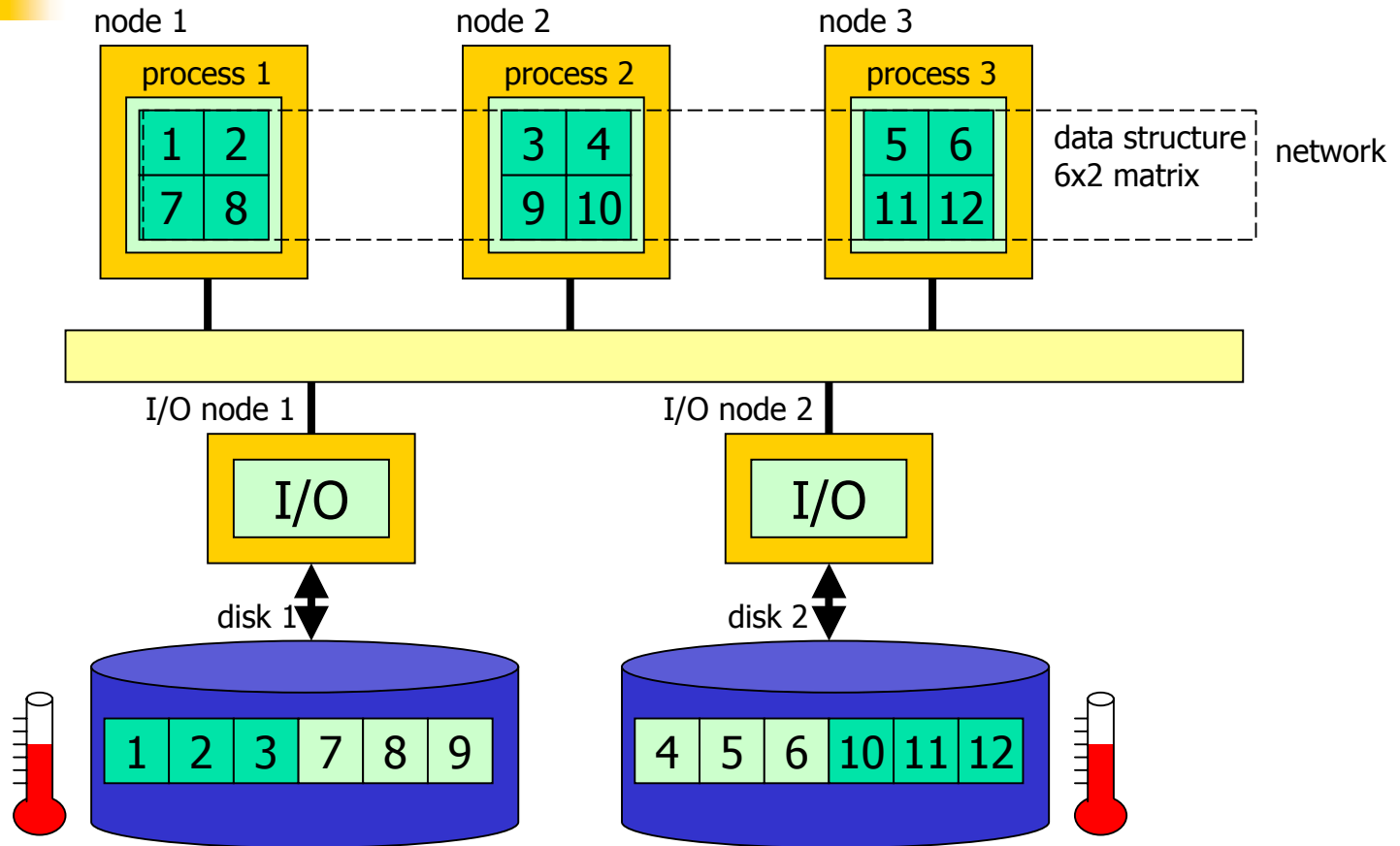
Performance Measurement

- What we can already measure
 - read/write-calls in user programs
 - disk access rates
- What's missing
 - Show corresponding values
 - I.e. which user program collective noncontiguous call results in which disk access pattern?
- Why? Needed for ...
 - ... program tuning (manual)
 - ... library tuning (manual)
 - ... load balancing (automatic)

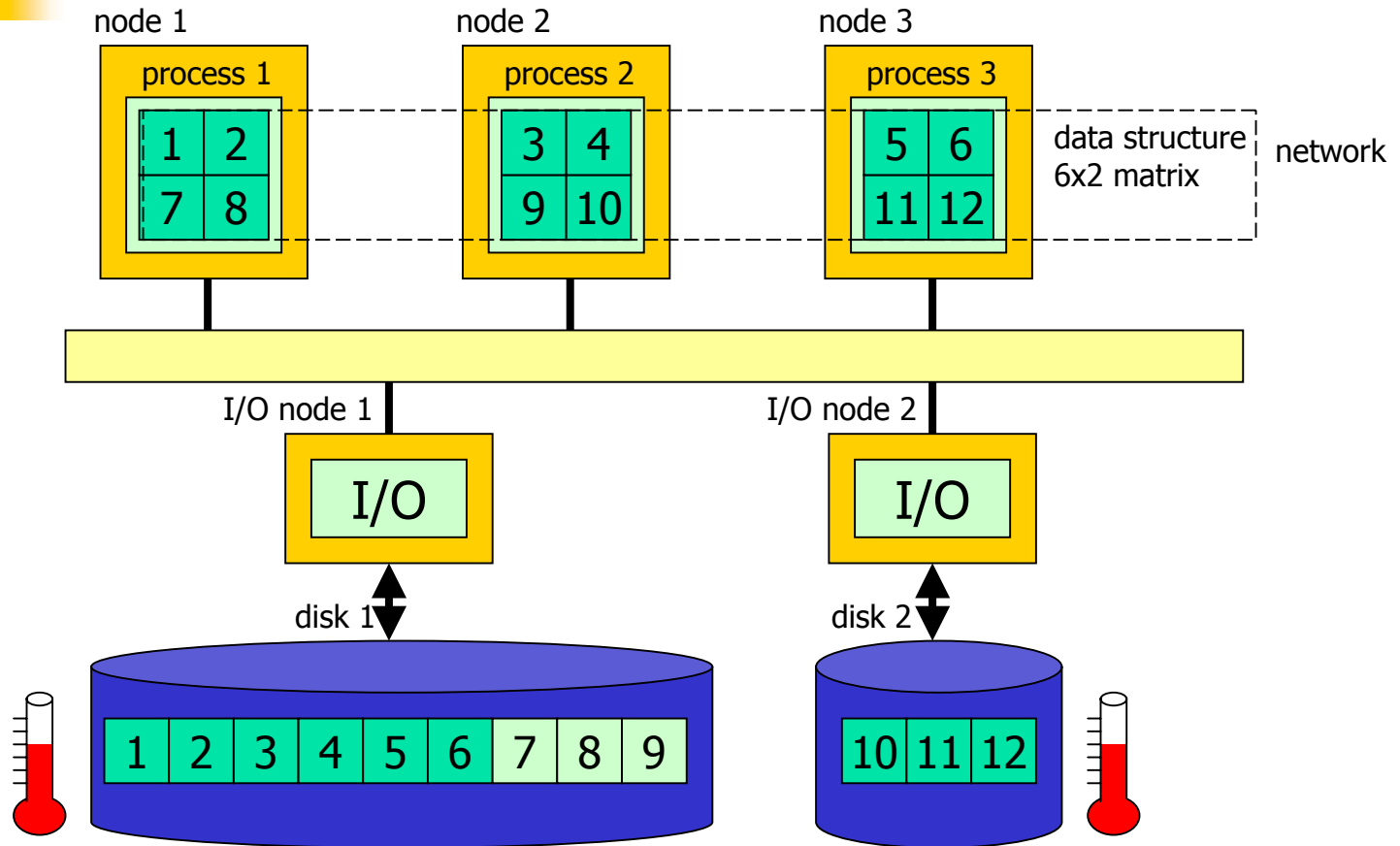
Load Balancing



Load Balancing by Redistribution



Load Balancing by Migration





The Future

- Cooperation with PVFS-group
 - Improvement of critical parallel file system issues
- Cooperation with users in Heidelberg
 - Deployment of parallel I/O at the Helics cluster
 - Research cooperations with users



Links

- www.research.ibm.com/bluegene
 - sc-2002.org/paperpdfs/pap.pap207.pdf
- www.top500.org
- www.pfvs.org
- www.lustre.org
- www.mpi-forum.org
- www-unix.mcs.anl.gov/mpi
- www-unix.mcs.anl.gov/mpi/mpich