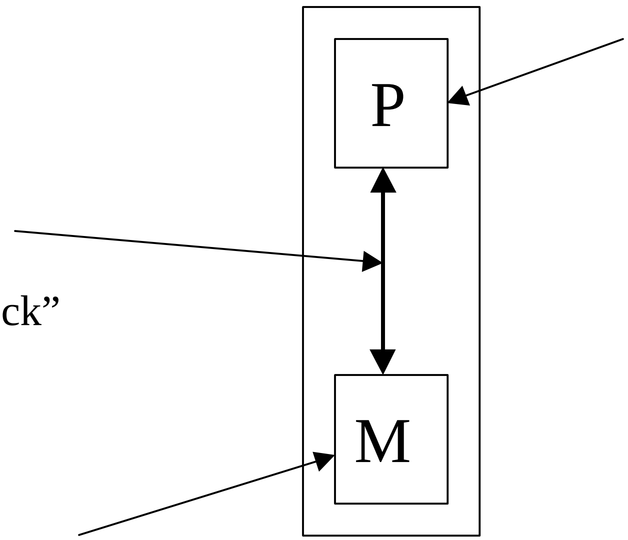
Distributed Computers

Andrew Huang

2/1/00

6.911 Architectures Anonymous

Generic Computer / PC

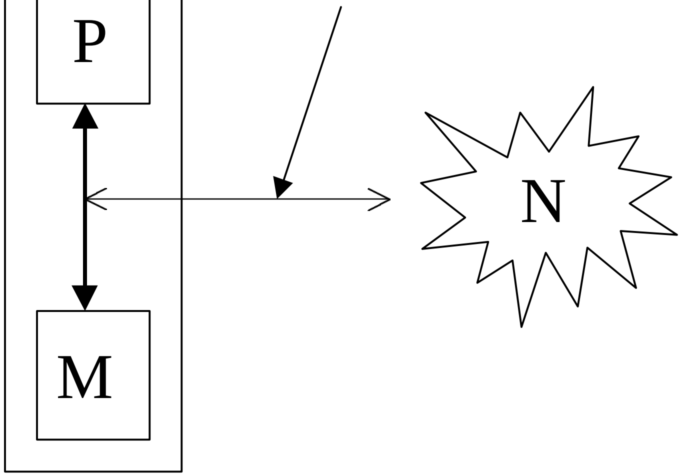


Single processor, i.e.
one program counter

BW = 1 GB/s,
lat. = 150 ns
“von neumann bottleneck”

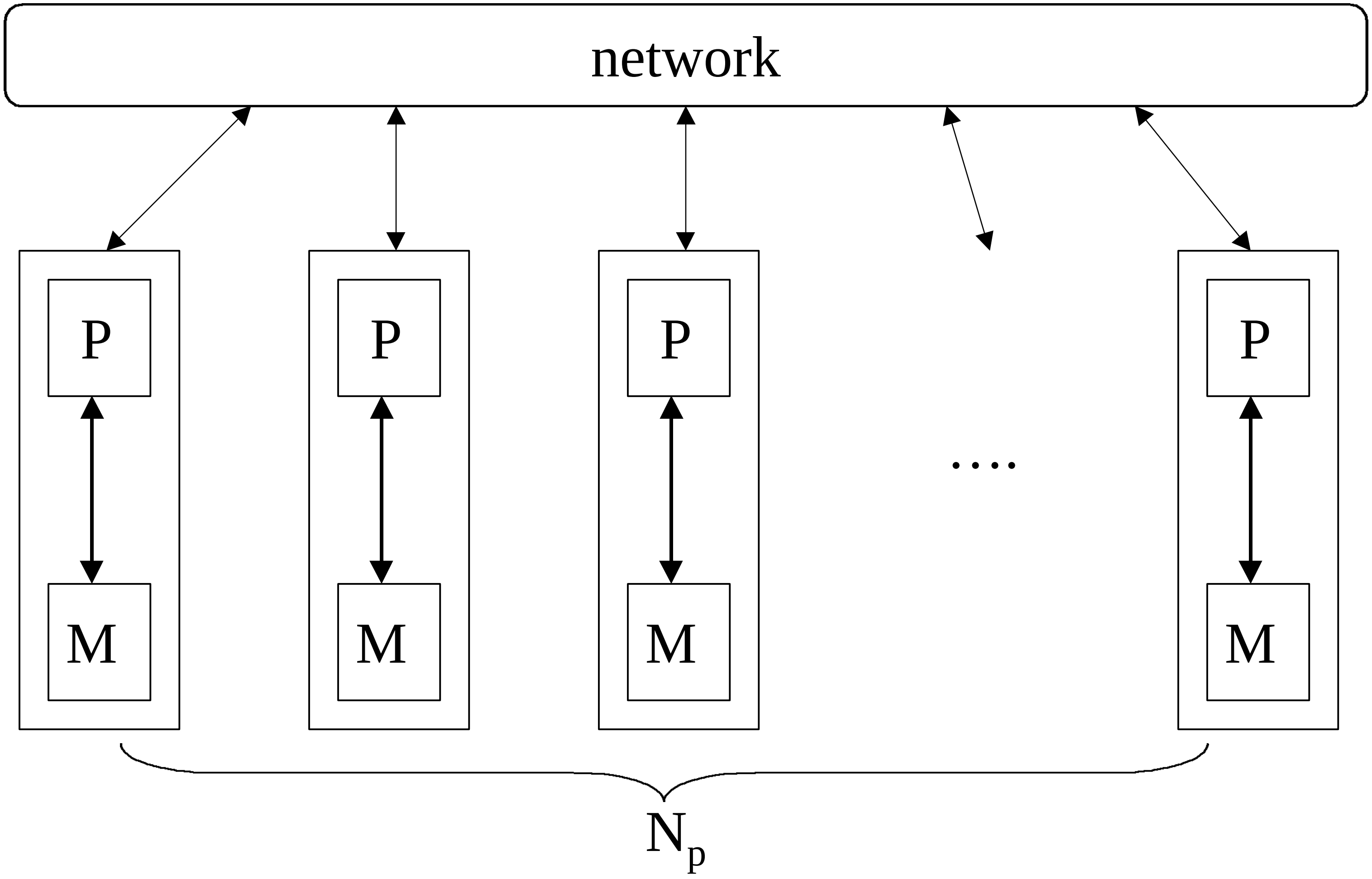
Single memory space,
large (100 MB), backed by secondary
storage (disk)

Generic Computer / PC



BW ~ 0.01 GB/s
lat. ~ 1 ms or 1,000,000 ns

Network Of Workstations (NOW)



NOW

- Examples of NOWs
  - seti@home & other screensavers
  - rendering farms

NOW Pros

- Cost
  - virtually free
- Collateral performance growth with COTS technology
  - R&D costs amortized over huge market
- Well suited for low communications bandwidth, processor intensive applications

NOW Cons

- Poor performance on many important problems
  - communications intensive, non-localized problems
  - granularity mismatch
- Restrictive programming model
- System management difficult
  - nonhomogenous networks, unreliable clients

Beowulf

- Beowulf clusters
  - incremental improvement over NOW
  - dedicated machines in dedicated network
    - typically network of 2-4 processor SMP x86-class machines, 128 MB memory, 10 GB disk
    - typically 100 Mbit or 1 Gbit ethernet
  - uniformity helps performance tweaking, system admin

Beowulf Pros

- Retains collateral technology benefits of NOW
- dedication of hardware allows for tweaking
  - highly optimized network card drivers available
  - bonded ethernet for more bandwidth
- better programming models
  - MPI, PVM, BSP, BPROC, DSM software layers available

Programming Models

- Message Passing
  - MPI (Message Passing Interface)
  - PVM (Parallel Virtual Machine)
  - BSP (Bulk Synchronous Processing)
- Shared memory
  - DSM, similar to Shasta developed at DRL
- Shared parallel filesystems

Beowulf Cons

- Limited communication bandwidth
  - fails on out-of-core computations, large databases, synchronization intensive code
- star/switched network topology
- security
- reliability
- programmer’s environment
- debugging?

Extreme Beowulf

- Dedicated, higher performance NI, richer network
  - ASCI (Accelerated Strategic Computing Initiative) Red
    - Highest performance computer today (Top500)
      - 4536 nodes @ 2 PPro processers/node
      - 0.5 TB DRAM overall @ 0.5 MB/s BW to a processor
      - 1 TB disk @ 1 GB/s RAID BW per subsystem
      - 800 MB/s network interfaces, 51.6 GB/s bisection BW, mesh network
      - message passing programming model
      - no published latency numbers

It’s the Wires...

Michael Hannah

Compare/Contrast

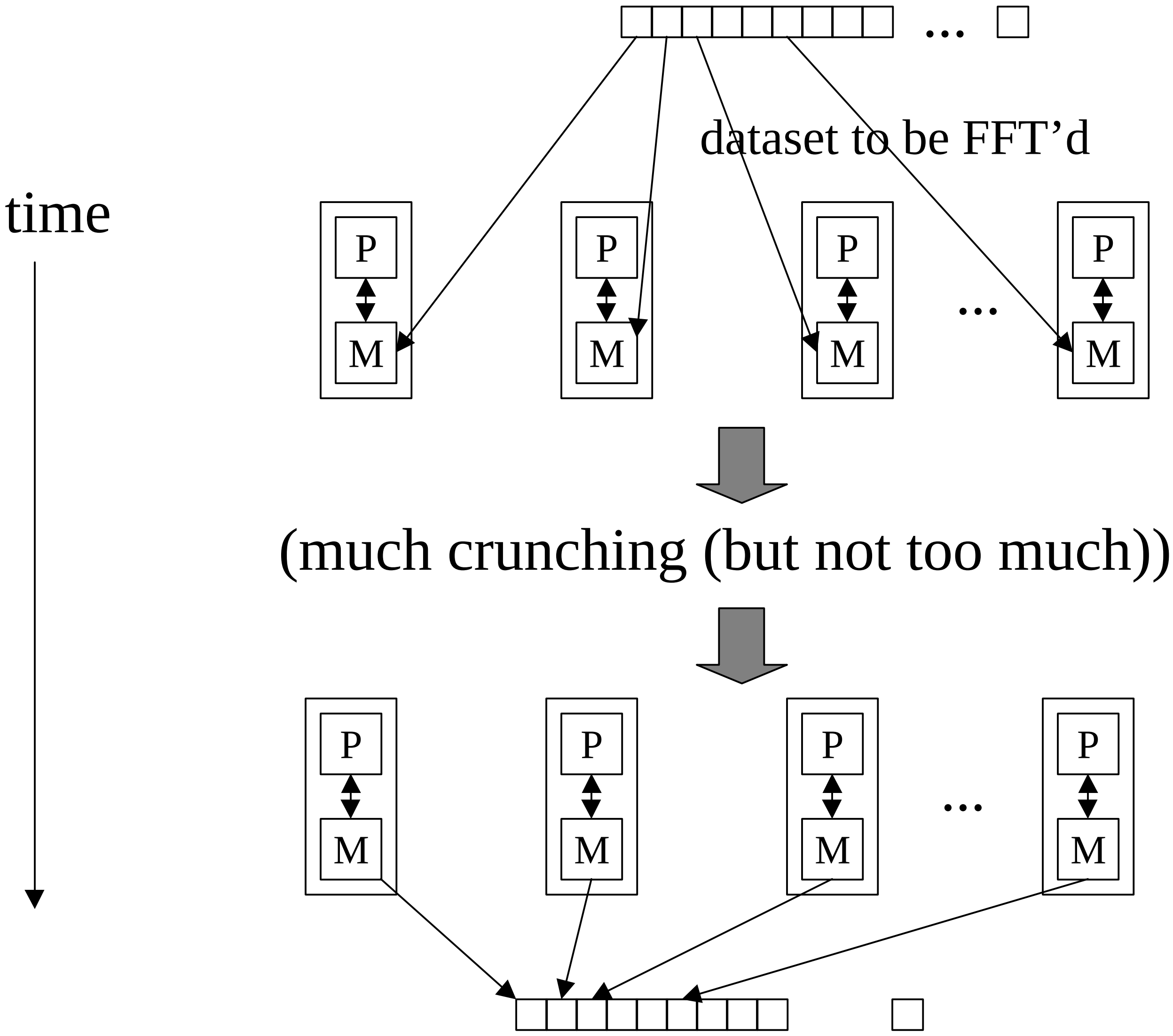
- SGI Origin 2K
  - 2 GB/s per-link network BW, 371 ns latency in largest systems, hiearchical fat hypercube
  - scalable to 512 nodes, ccNUMA/shared memory model
  - cost is 5x to 10x that of COTS distributed machine

Applications on Dist. Machs.

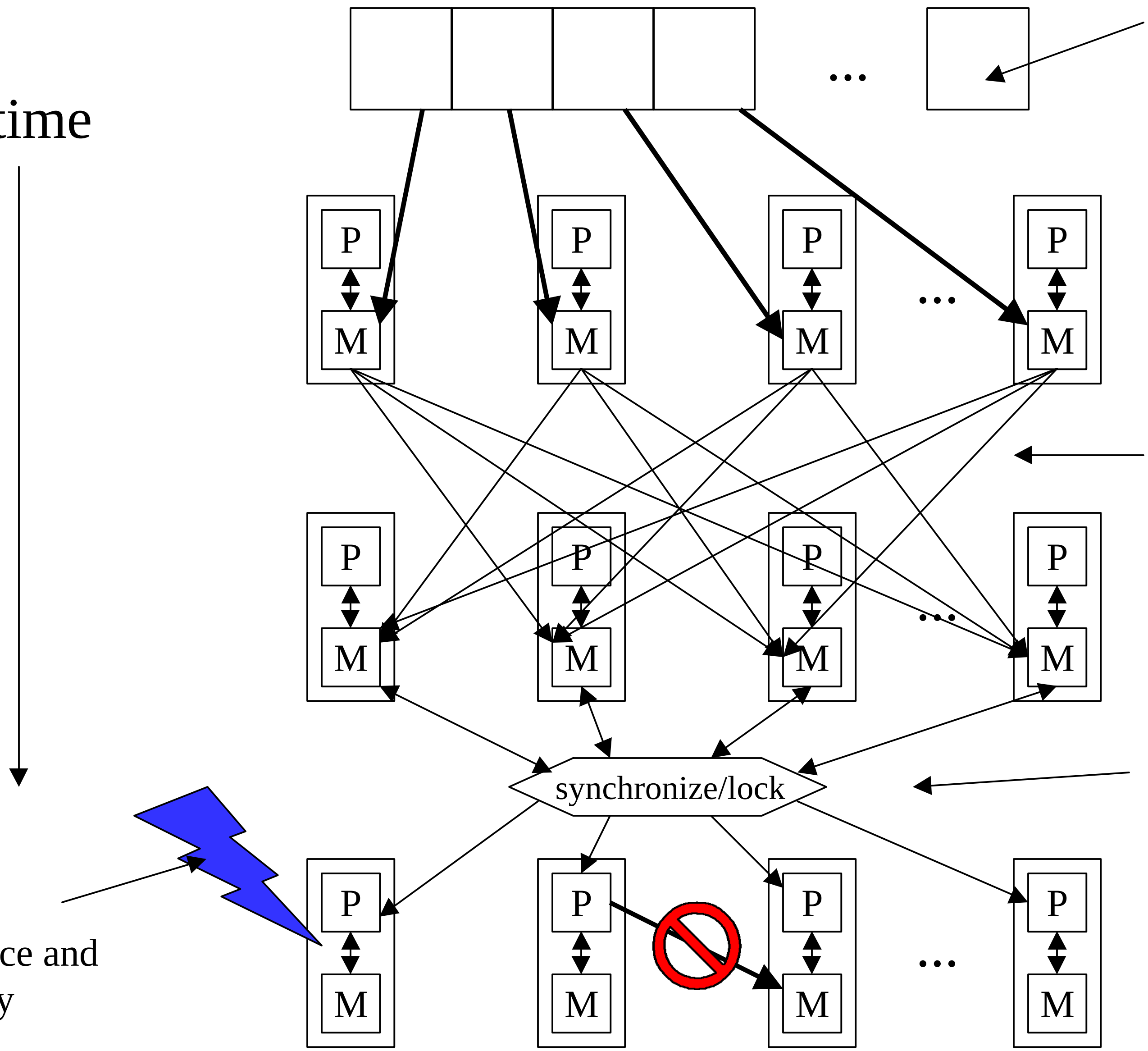
- $T_{proc} >> Latency_{net}$, but T_{proc} still manageable in real-time
- Dataset size < node local storage size
- few dependencies, synchronizations
- $BW_{proc-proc} < BW_{net}$

In-core solvers with few dependencies, i.e., crypto, off-line movie rendering; also, algorithms that can be coarsely partitioned, i.e., N-body problems, fluid flow

Applications on Dist. Machs.



Breaking Dist. Machs.



time

Large, out of core datasets

Data shuffling

synchronization

Fault tolerance and security

Summary

- Distributed computers are cheap and great for a limited number of applications
  - collateral technology scaling with mainstream computer technology
- There are some things you just can’t do with a distributed computer...
  - There is a better way...
  - To be continued!