Outline

- Introduction
- Background
- Distributed DBMS Architecture
- Distributed Database Design
- Distributed Query Processing
- Distributed Transaction Management
- Building Distributed Database Systems (RAID)
- Mobile Database Systems
- Privacy, Trust, and Authentication
- Peer to Peer Systems

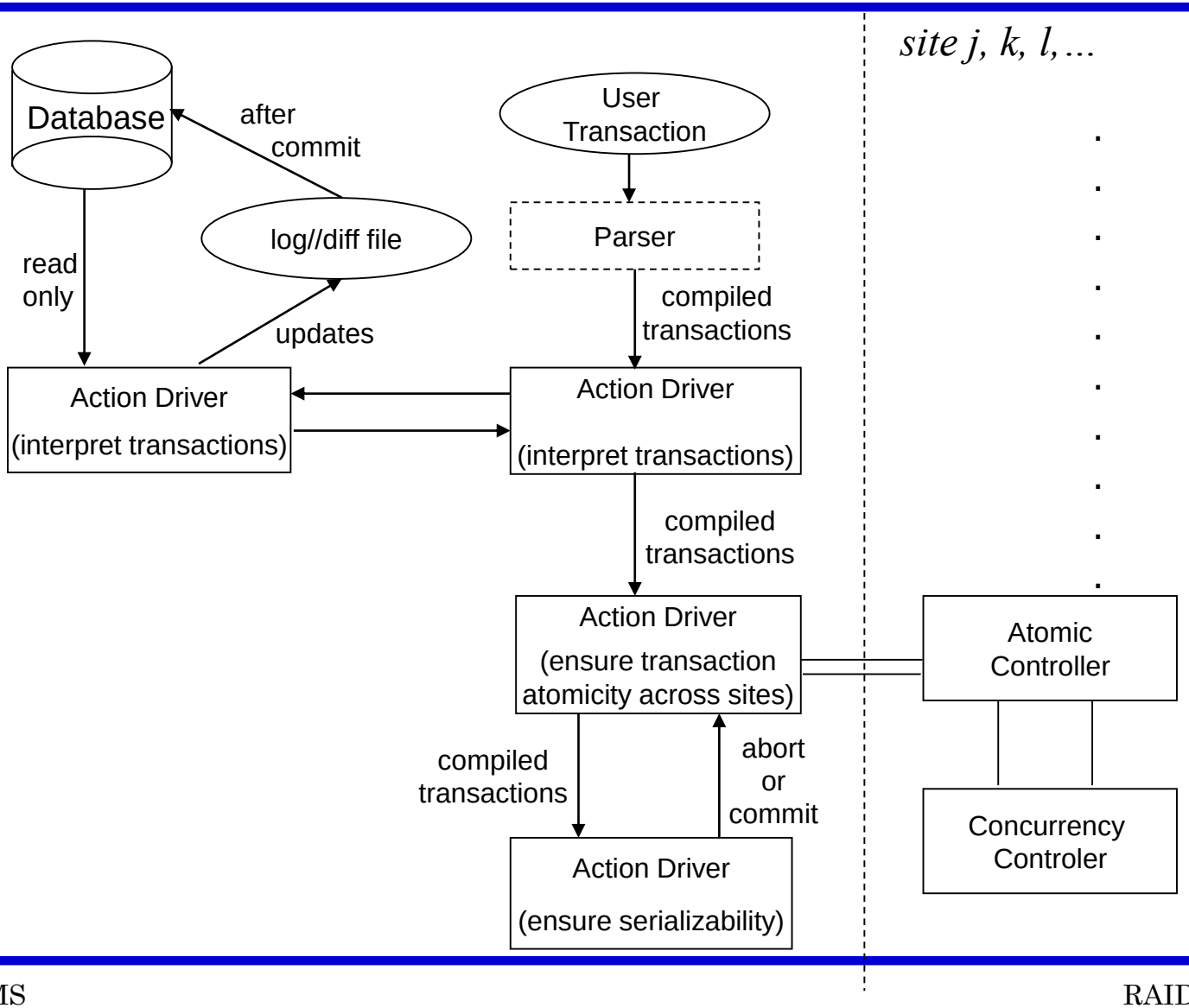
Useful References

- B. Bhargava and John Riedl, *The Raid Distributed Database System*, IEEE Trans on Software Engineering, 15(6), June 1989.
- B. Bhargava and John Riedl, *A Model for Adaptable Systems for Transaction Processing*, IEEE Transactions on Knowledge and Data Engineering, 1(4), Dec 1989.
- B. Bhargava, *Building Distributed Database Systems*.
- Y. Zhang and B. Bhargava, *WANCE: Wide area network communication emulation systems*, IEEE workshop on Parallel and Distributed Systems, 1993.
- E. Mafla, and B. Bhargava, *Communication Facilities for Distributed Transaction Processing Systems*, IEEE Computer, 24(8), 1991.
- B. Bhargava, Y. Zhang, and E. Mafla, Evolution of a communication system for distributed transaction processing in RAID, Computing Systems, 4(3), 1991.

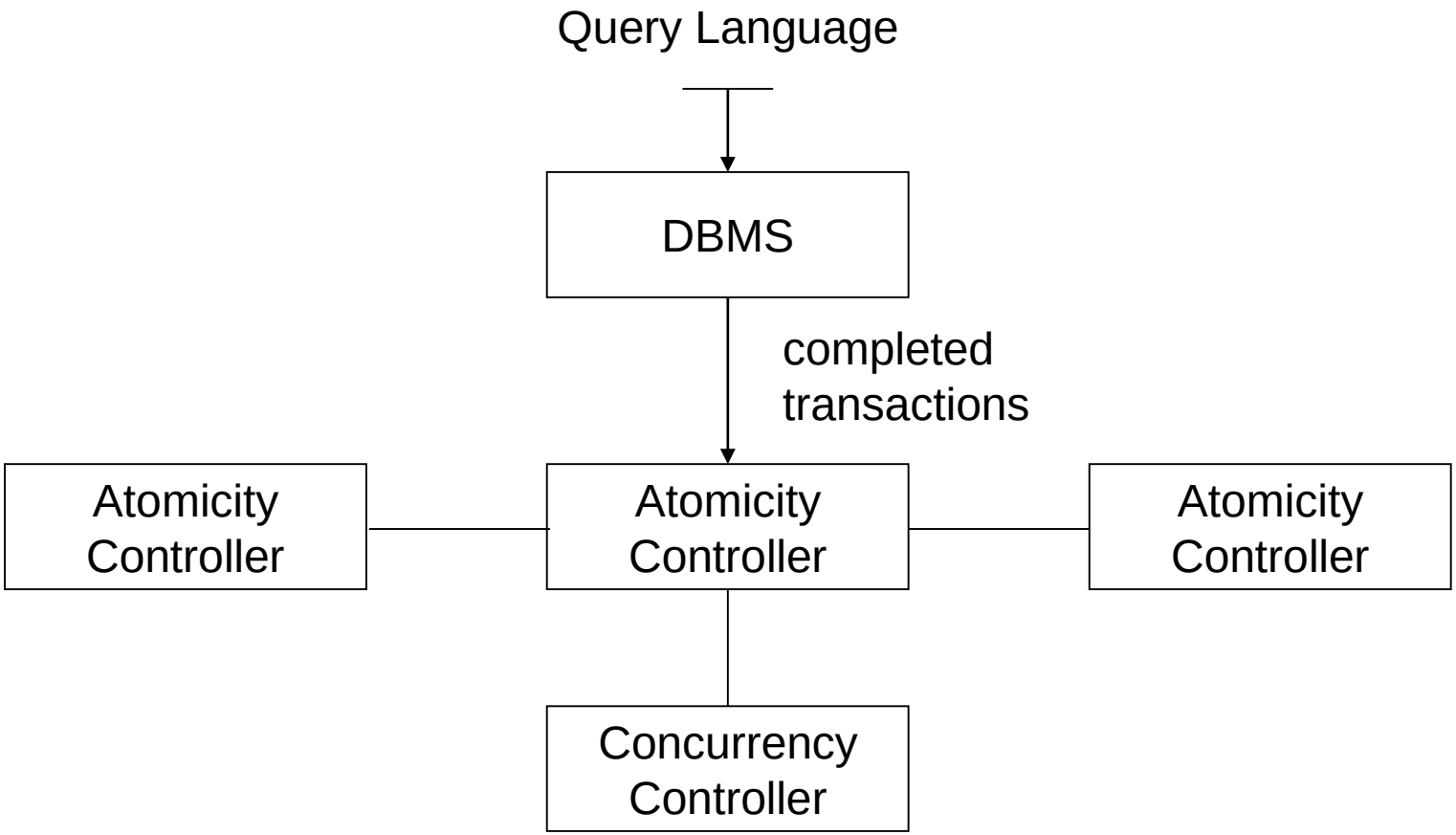
Implementations

1.	LOCUS (UCLA)	File system OS
2.	TABS (Camelot) (CMU)	Data servers OS
3.	RAID (Purdue)	Database level (server)
4.	SDD-1 (Computer Corp. of America)	Transaction manager Data manager
5.	System – R* (IBM)	Database level
6.	ARGUS (MIT)	Guardian (server)

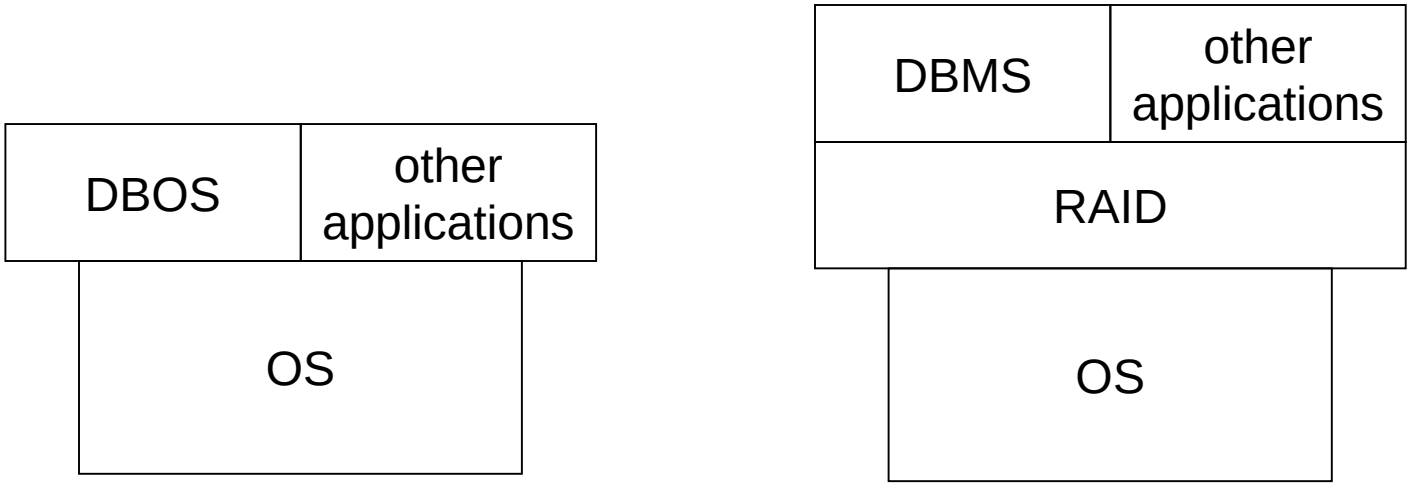
Architecture of RAID System



RAID Transactions



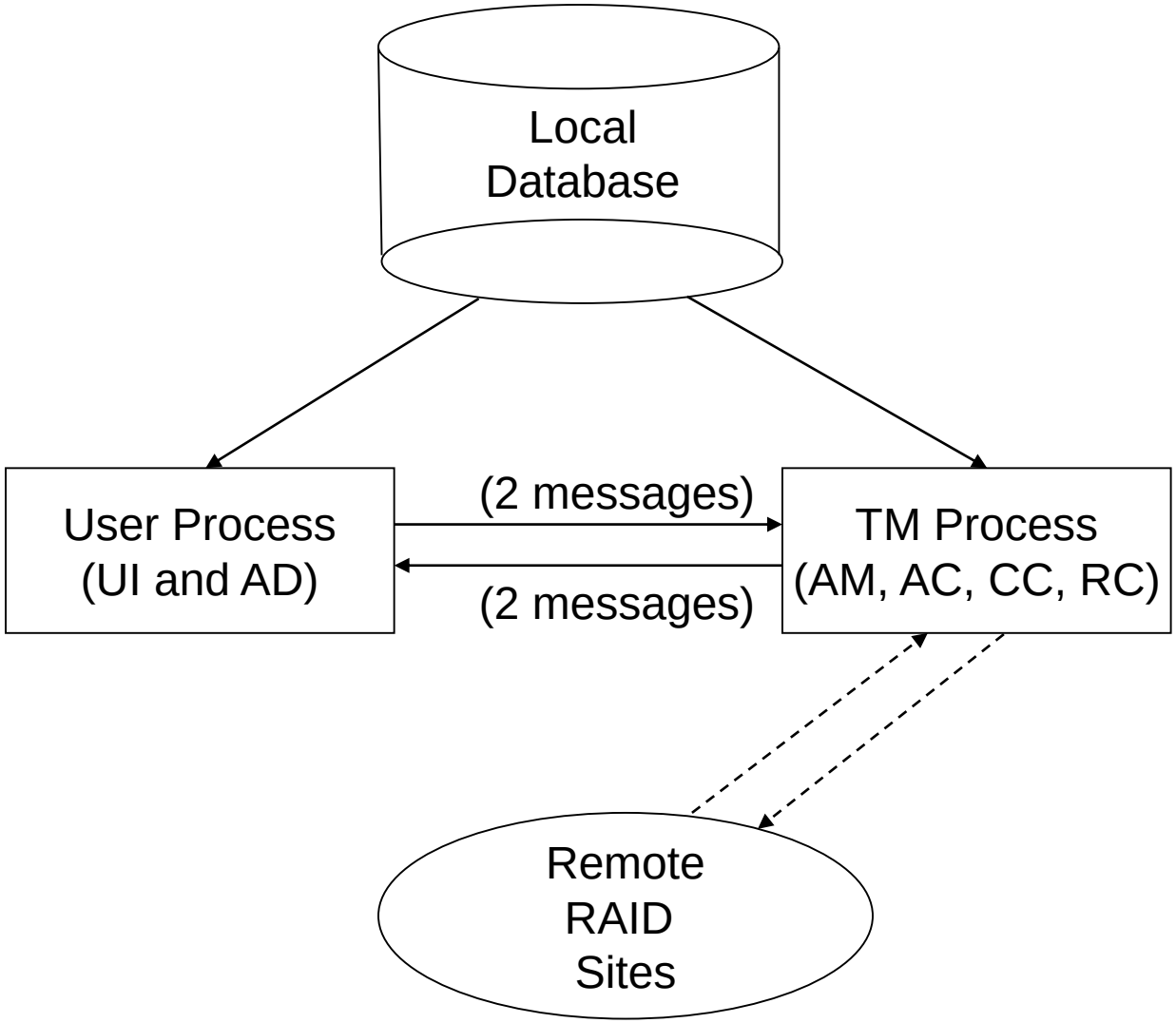
RAID Distributed System



RAID supports reliability

- transactions
- stable storage
- buffer pool management

Transaction Management in one Server



CPU time used by RAID servers in executing transactions

Server CPU Time (second)				
Server	AC		CC	
Transaction	user	system	user	system
Select one tuple	0.04	0.14	0.04	0.06
select eleven tuples	0.04	0.08	0.02	0.02
Insert twenty tuples	0.20	0.16	0.12	0.13
Update one tuple	0.04	0.10	0.02	0.02

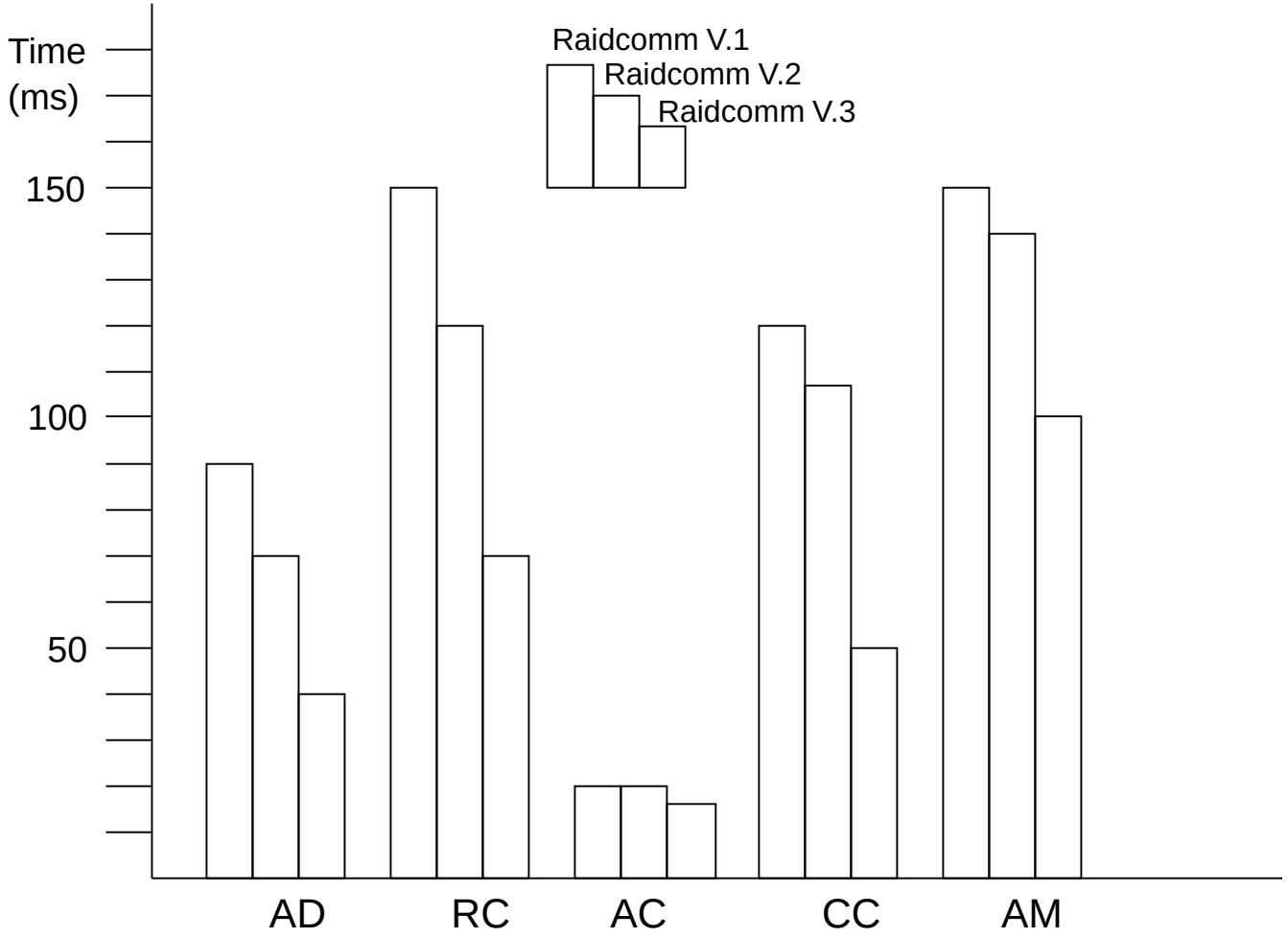
Server	AD		AM	
Transaction	user	system	user	system
Select one tuple	0.34	0.90	0.00	0.00
select eleven tuples	0.54	1.48	0.00	0.00
Insert twenty tuples	1.23	3.10	0.14	0.71
Update one tuple	0.34	0.76	0.04	0.58

RAID Elapsed Time for Transactions in seconds

Transaction	1 site	2 sites	3 sites	4 sites
Select one tuple	0.3	0.3	0.4	0.4
Select eleven tuples	0.4	0.4	0.4	0.4
Insert twenty tuples	0.6	0.6	0.8	0.8
Update one tuple	0.4	0.4	0.4	0.4

Performance Comparison of the Communication Libraries

Message († multicast dest = 5)	Length Bytes	*Raidcomm V.1* *μs*	*Raidcomm V.2* *μs*	*Raidcomm V.3* *μs*
SendNull	44	2462	1113	683
MultiNull †	44	12180	1120	782
Send Timestamp	48	2510	1157	668
Send Relation Descriptor	76	2652	1407	752
Send Relation Descriptor †	72	12330	1410	849
Send Relation	156	3864	2665	919
Send Write Relation	160	3930	2718	1102


Time
(ms)
150
100
50
Raidcomm V.1
Raidcomm V.2
Raidcomm V.3
AD
RC
AC
CC
AM

Experiences with RAID Distributed Database

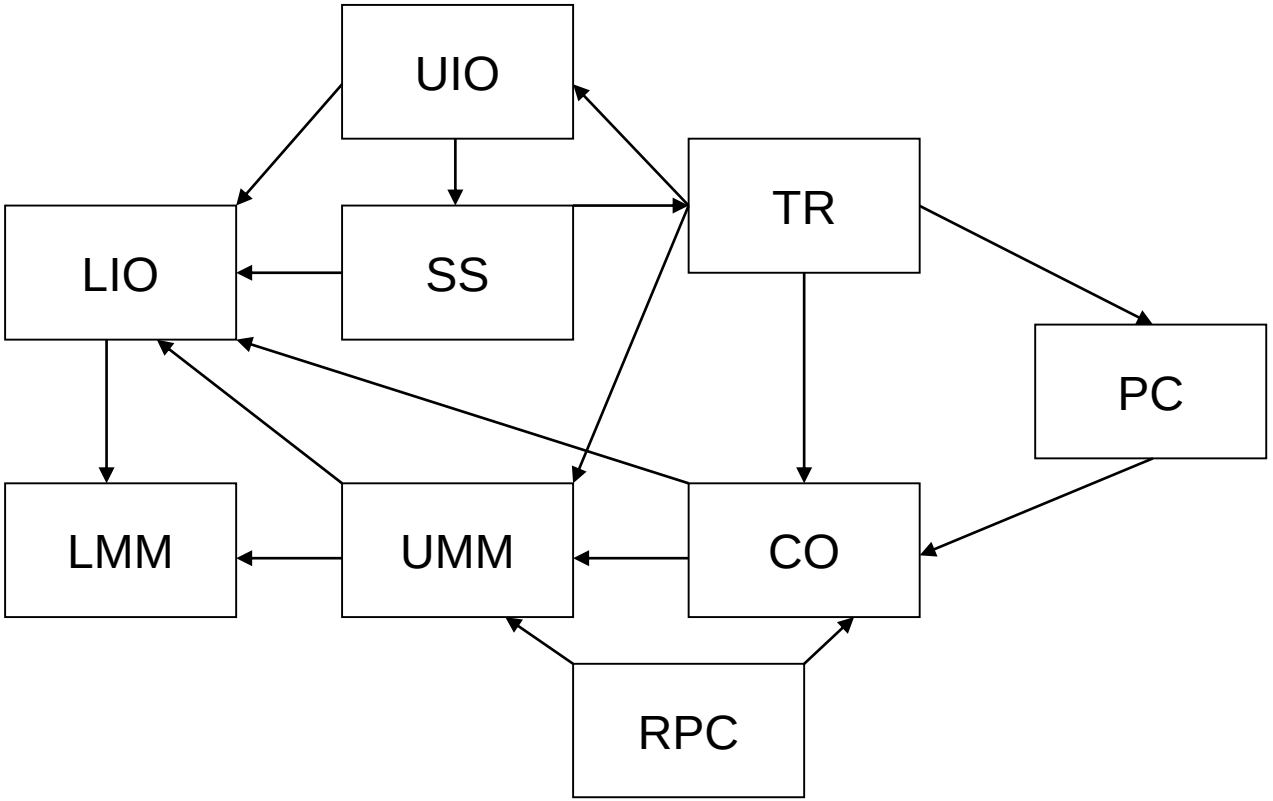
- Unix influences must be factored out.
- Communications software costs dominate everything else.
- Server based systems can provide modularity and efficiency.
- Concurrent execution in several server types is hard to achieve.
- Need very tuned system to conduct experiments.
- Data is not available from others for validation.
- Expensive research direction, but is respected and rewarded.

Layers

- Transaction
  - Transparent to errors
  - RPC – executed within transaction
- Communication
  - Datagram as basic unit
  - RPC – user level IPC (provides atomicity/reliability)
  - Broadcast
- Input/Output System
  - Lower level I/O (reading and writing raw data)
  - Stable storage (log, UNDO, REDO)
  - Upper level I/O (reliable read/write)

Description of Symbols

SS	Stable storage: special I/O for log, backup info.
LMM	Low level memory management: simple physical memory services.
UMM	Upper level memory management: virtual memory, if desired; buffer pools.
CO	Communication: semi-reliable datagrams (at most once?) ; broadcast.
LIO	Low-level I/O: interface to hardware I/O.
UIO	Upper-level I/O: read/write protocol, permitting replication and providing location independence; includes logging.
TR	Transactions management: BeginTrans, EndTrans, and Abort verbs; provides concurrency control; must have ‘hooks’ in read/write system calls.
RPC	Remote procedure call support: reliable RPC.
PC	Partition control: provides virtual fail-proof network to upper layers (also handles site failure/recovery).



Dependency graph for proposed layered, distributed system

The THOUSAND Relation

unique	two	ten	twenty	hundred	thousand
5	0	9	2	64	615
6	1	3	15	36	923
7	0	3	17	68	746
8	1	5	8	80	424
9	0	9	3	59	707
10	0	3	19	32	455
11	1	6	16	20	832
12	1	1	6	79	719
13	1	9	3	19	639
14	1	0	4	41	872
15	1	2	4	84	931

Some example tuples from the thousand relation

Single-Site RAID

Transaction	Interpret	Commit	Write
Select one tuple	2.1	0.4	0.0
Select eleven tuples	3.4	0.4	0.0
Insert twenty tuples	3.8	1.4	2.8
Update one tuple	1.9	0.5	1.5

Execution times in 1-site RAID (in seconds)

Two-Site RAID

Transaction	Interpret	Commit	Write
Select one tuple	1.9	0.4	0.0
Select eleven tuples	3.6	0.5	0.0
Insert twenty tuples	3.5	1.4	3.4
Update one tuple	1.8	0.5	1.6

Execution times in 2-site RAID (in seconds)

Three-Site RAID

Transaction	Interpret	Commit	Write
Select one tuple	2.0	0.4	0.0
Select eleven tuples	3.4	0.4	0.0
Insert twenty tuples	4.1	1.4	5.1
Update one tuple	1.7	0.4	2.2

Execution times in 3-site RAID (in seconds)

Four-Site RAID

Transaction	Interpret	Commit	Write
Select one tuple	2.2	0.4	0.0
Select eleven tuples	3.7	0.4	0.0
Insert twenty tuples	3.9	1.5	6.9
Update one tuple	1.8	0.4	2.7

Execution times in 4-site RAID (in seconds)