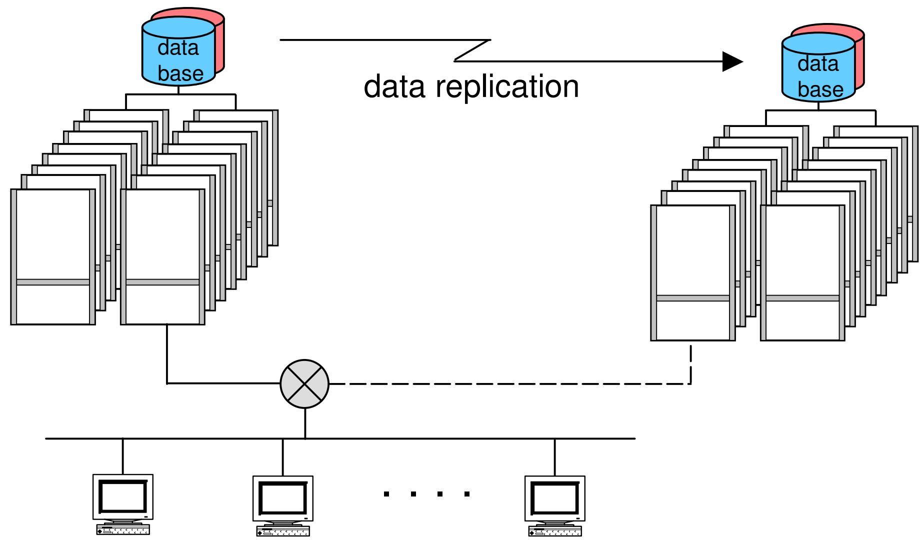
Achieving Century Availability

Dr. Bill Highleyman
Chairman
The Sombers Group, Inc.
billh@sombers.com

Disaster Tolerance Today

NonStop Active/Passive Architecture



Characteristics:

- Recovery time minutes to hours (RTO)
  - Recover transactions
  - Bring up applications
  - Switch users
- Loss of data in the replication pipeline (RPO)
- All users affected upon failover
- Capacity underutilization

The Sombers Group

Question: How can we improve on this?

- Instant recovery time (RTO = 0)
- No data loss (RPO = 0)
- 100% capacity utilization
- Higher availability
- AND all this at less cost

Answer: Active/Active Architectures

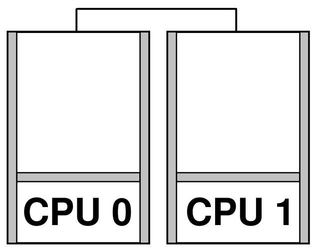
(one in which every system carries its share of the load)

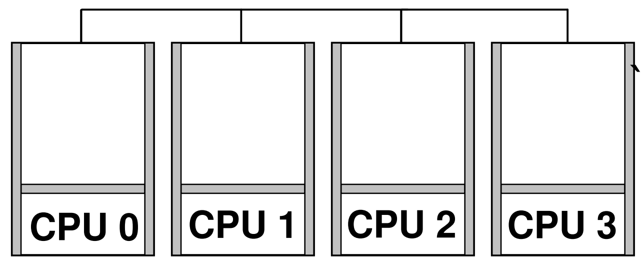
HP WORLD 2004
Solutions and Technology Conference & Expo

The Sombers Group

First, some NonStop background:

In a NonStop system, downtime occurs when two subsystems fail, taking down a critical process:





X	X

1 failure mode

Failure 1	X	X		
Failure 2	X		X	
Failure 3	X			X
Failure 4		X	X	
Failure 5		X		X
Failure 6			X	X

6 failure modes

As a system grows larger, failure modes increase, and failures occur more frequently.

HP WORLD 2004 Solutions and Technology Conference & Expo

In a NonStop multi-processor system, downtime grows faster than the square of the system size

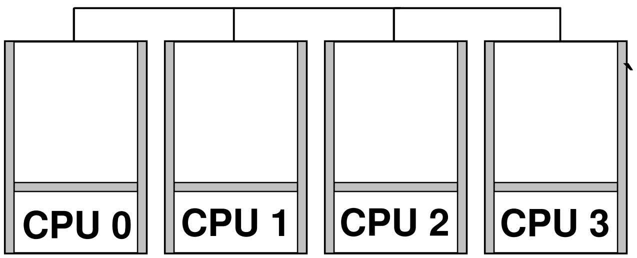
In general, if

n = number of processors in the system

then

failure modes = $\frac{n(n-1)}{2}$

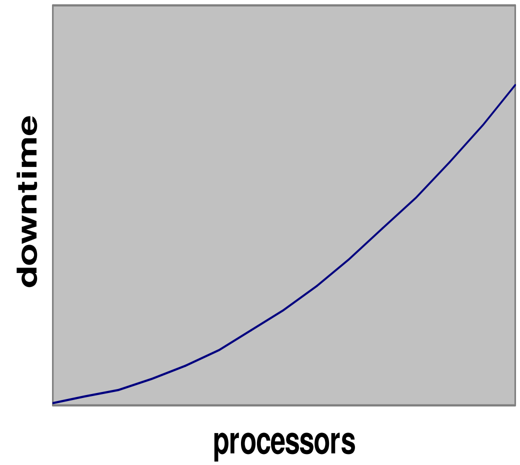
n	failure modes	downtime
2	1	x1
4	6	x6
8	28	x28
16	120	x120



X	X		
X		X	
X			X
	X	X	
	X		X
		X	X

The System Size Rule

As a NonStop system grows, downtime grows even faster.

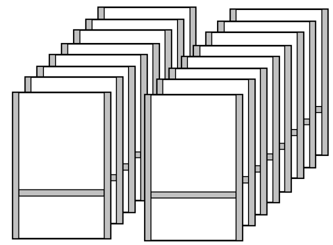


System Splitting

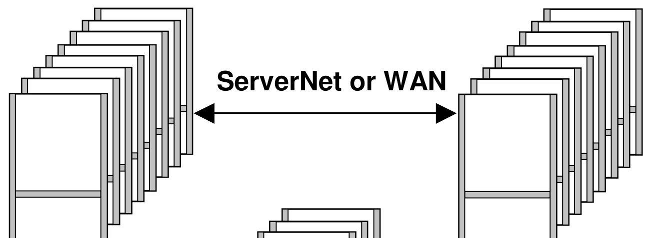
So – Can we gain anything by splitting a big NonStop system into smaller nodes?

	failure modes	downtime reduction
one 16-processor system	120	1
two 8-processor nodes	2x28=56	2.1
four 4-processor nodes	4x6=24	5

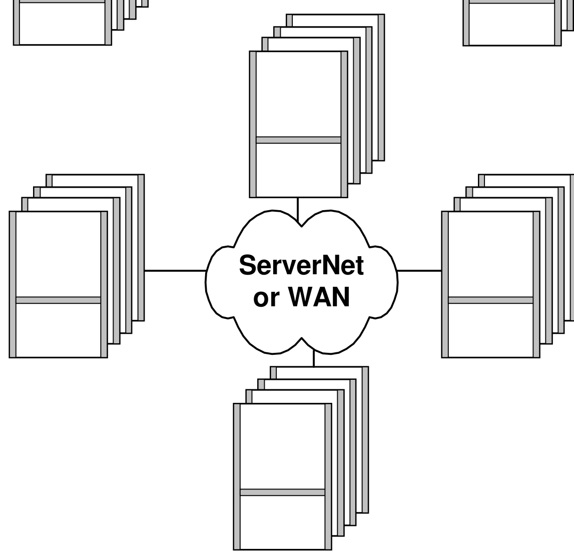
one 16-processor system



two 8-processor nodes



four 4-processor nodes

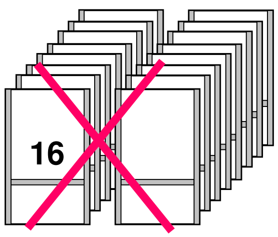


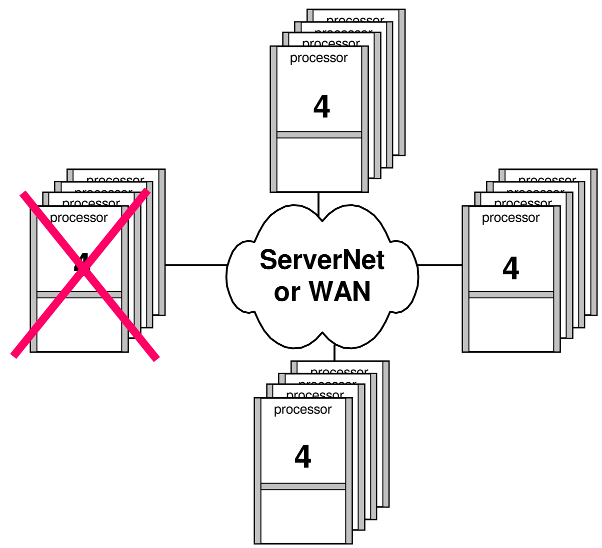
Yup!

HP WORLD 2004 Solutions and Technology Conference & Expo

System Splitting

We can improve a single system





Lose 100%
every 5 years

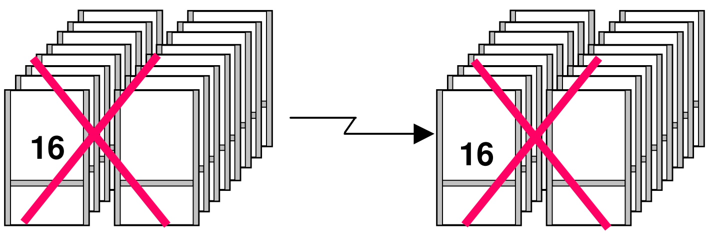
Lose 25%
every 25 years

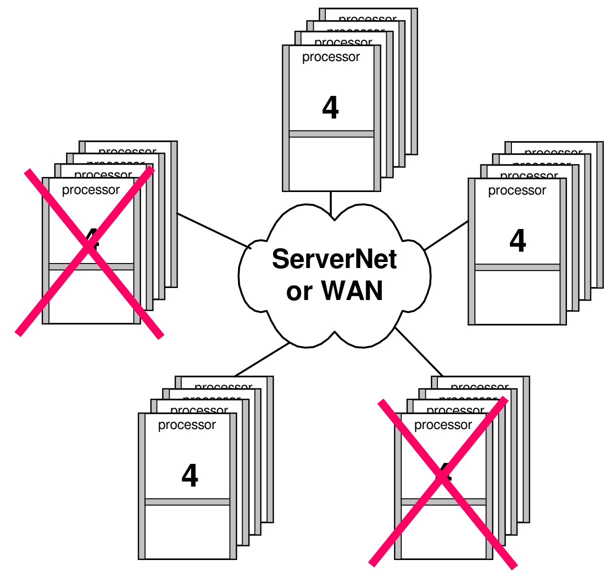
HP WORLD 2004
Solutions and Technology Conference & Expo

The Sombers Group

System Splitting

And we can improve an active/passive system





Lose 100%
every 500 centuries

Lose 25%
every 12,500 centuries
… at a fraction of the cost

Can This Be Extended to the UNIX World?

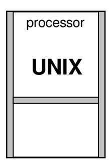
Yes, to a lesser but still effective extent

The 9's Game -

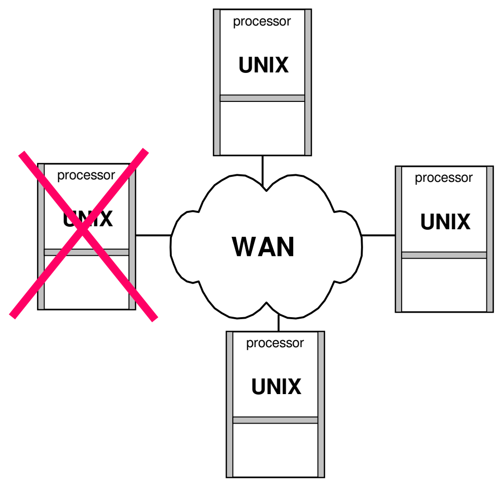
Himalaya	.9999	.8 hours/year
Mainframe	.999	8 hours/year
OpenVMS	.998	16 hours/year
AS400	.998	16 hours/year
HPUX	.996	32 hours/year
Tru64	.996	32 hours/year
Solaris	.995	40 hours/year
NT Cluster	.992 - .995	40 – 64 hours/year

System Splitting in the UNIX World

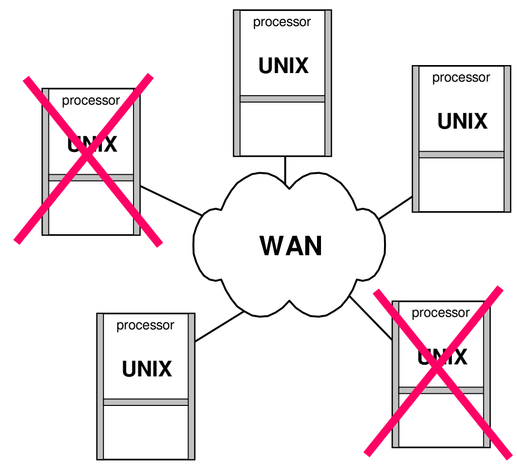
Let's give UNIX systems 3 9s (.999)



Unix: Lose 100% every 6 months (versus every 5 years)



UNIX: Lose 25% every 6 months (versus every 25 years)

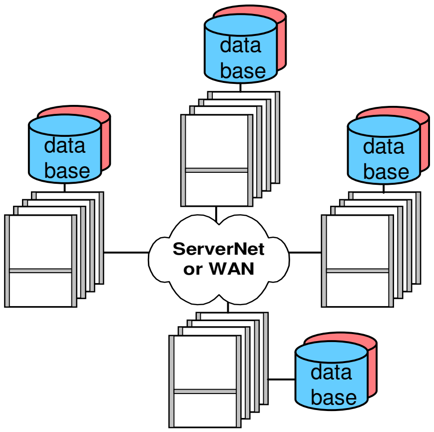


UNIX: Lose 25% every 5 centuries (versus every 12,500 centuries)

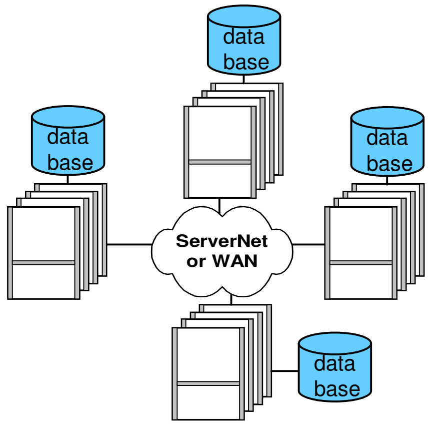
The Sombers Group

But what about the disk farms?

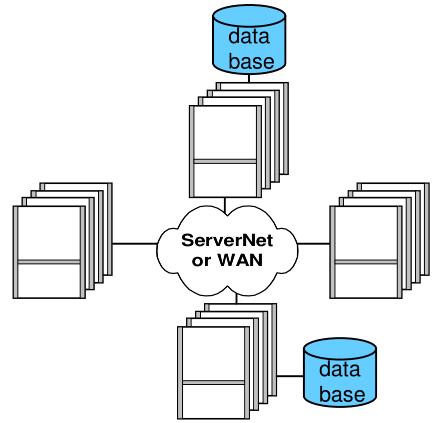
Often the predominant cost



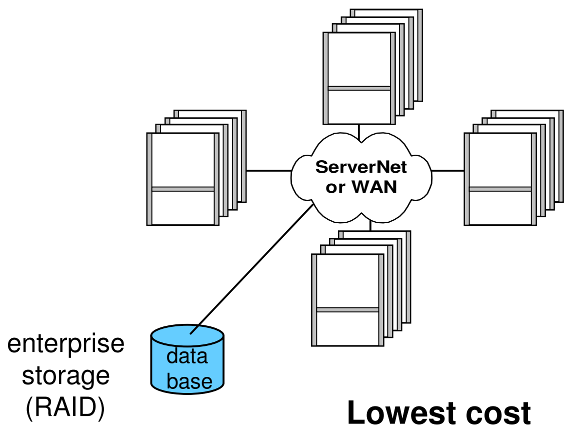
High cost
High performance
Highest availability



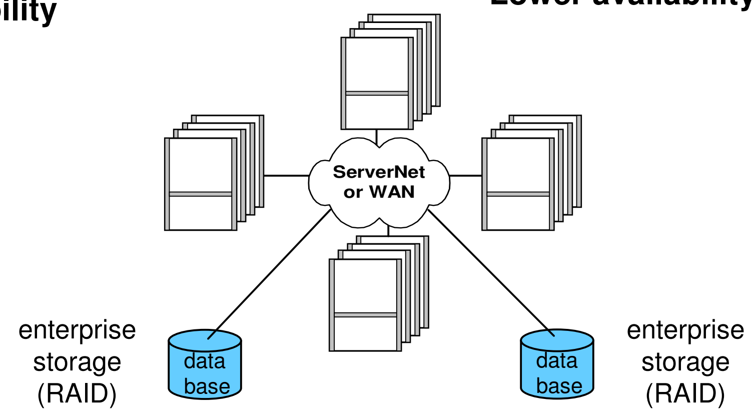
Lower cost
High performance
High availability



Lowest cost
High to Lower performance*
Lower availability



Lowest cost
High to lower performance*
High availability
Not Disaster Tolerant

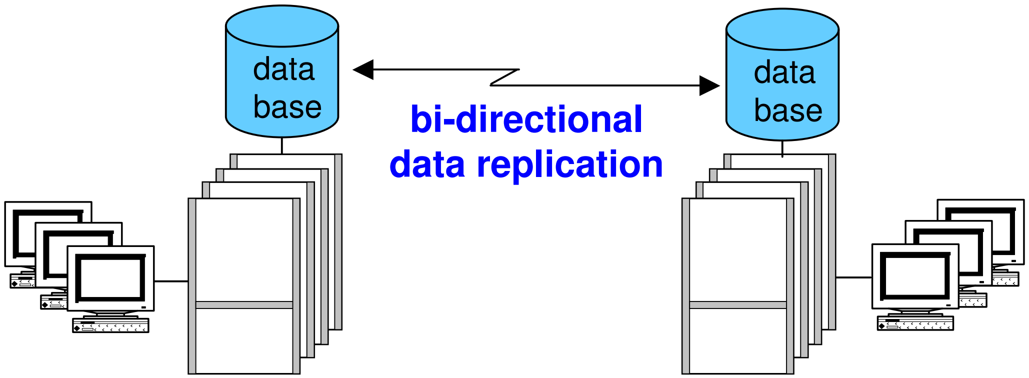


Lower cost
High to lower performance*
High availability
(coming)

***high over ServerNet, lower over WAN**

We've Selected a Disk Configuration

So how do we keep the networked disks in synchronization?



There are some issues with bi-directional data replication:

- RDF won't do it:
  - Can't open standby database for write.
  - Look to third party products.

- Data loss on failover
- Ping-ponging
- Collisions
- Referential integrity must be maintained.

If zero data loss is required, use synchronous replication ➡

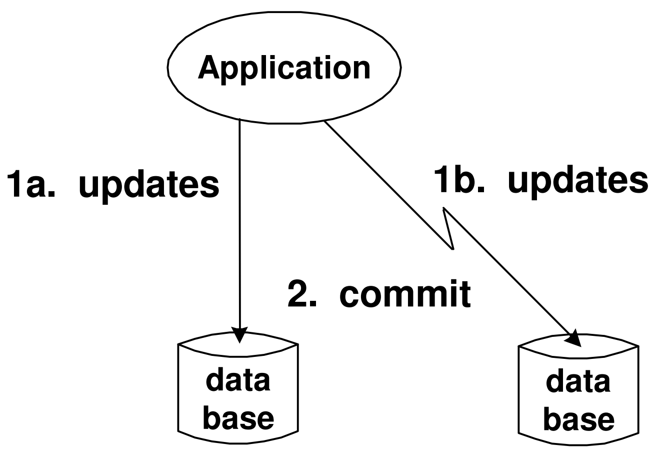
Synchronous Replication

Avoiding Data Loss, Avoiding Data Collisions

Data loss (RPO = 0) and data collisions are avoided by ensuring that all copies of a data item in the network are locked before any are updated.

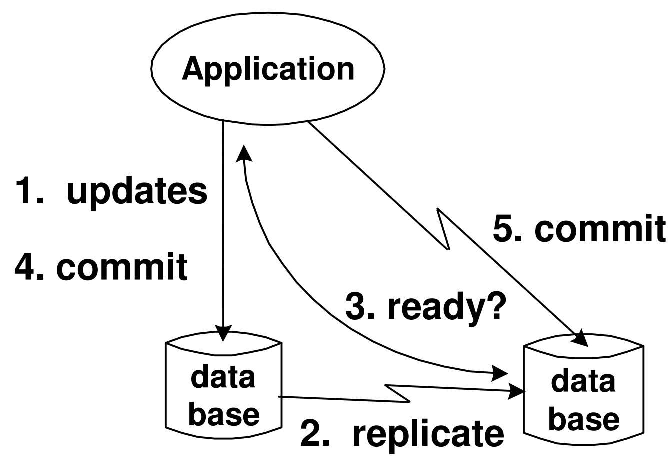
This can be done by making them part of the same transaction:

Single Transaction



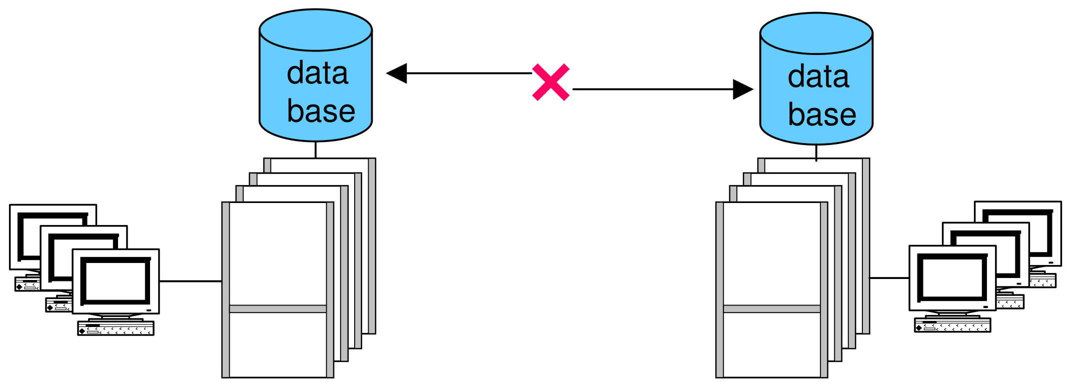
Network TMF

Coordinated Transactions



Coordinated Commits

What if we lose the network?



Option 1:

- Continue in operation at full capacity.
- Systems will get out of sync (split brain)
- On recovery, replication queues will drain, updating the nodes.
- Must resolve collisions

Option 2:

- Switch downed users to a surviving system
- Continue in operation at reduced capacity.
- Shed load if necessary.
- On recovery, replication queues will drain, updating the downed node.
- Switch users back.
- No collisions

The Sombers Group

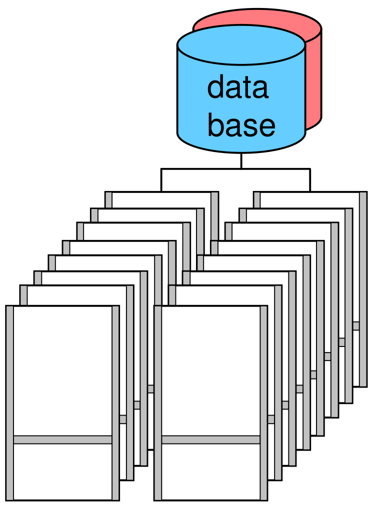
Other issues of which to be aware:

- Rerouting of users (same problem as with active/passive):
  - switches and routers
  - virtual IP (gratuitous ARP)
- Load shedding if additional capacity not provided.
- Licensing
- Network costs
- People costs

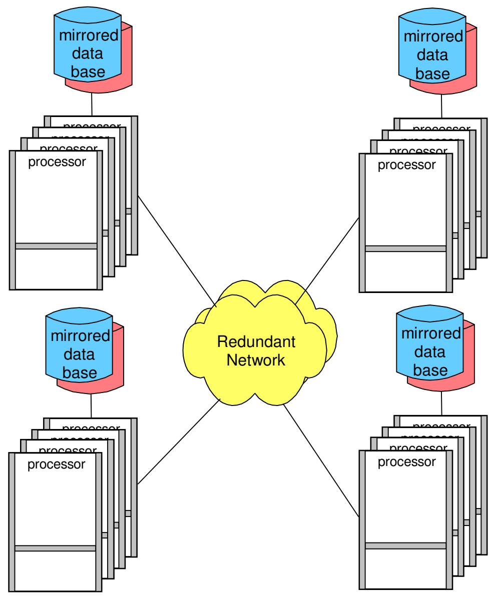
HP WORLD 2004
Solutions and Technology Conference & Expo

In Conclusion

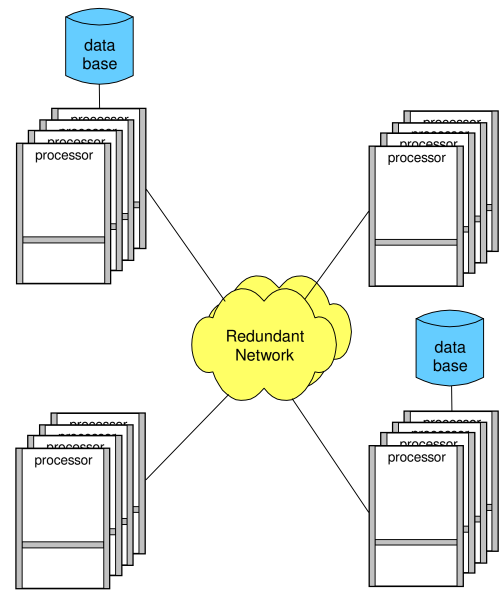
You can optimize availability, performance, and cost. Pick any two.



fast performance, low cost



high availability, fast performance



high availability, low cost

Remember: An unavailable system has zero performance. And its cost may be incalculable.

The Sombers Group

In Conclusion

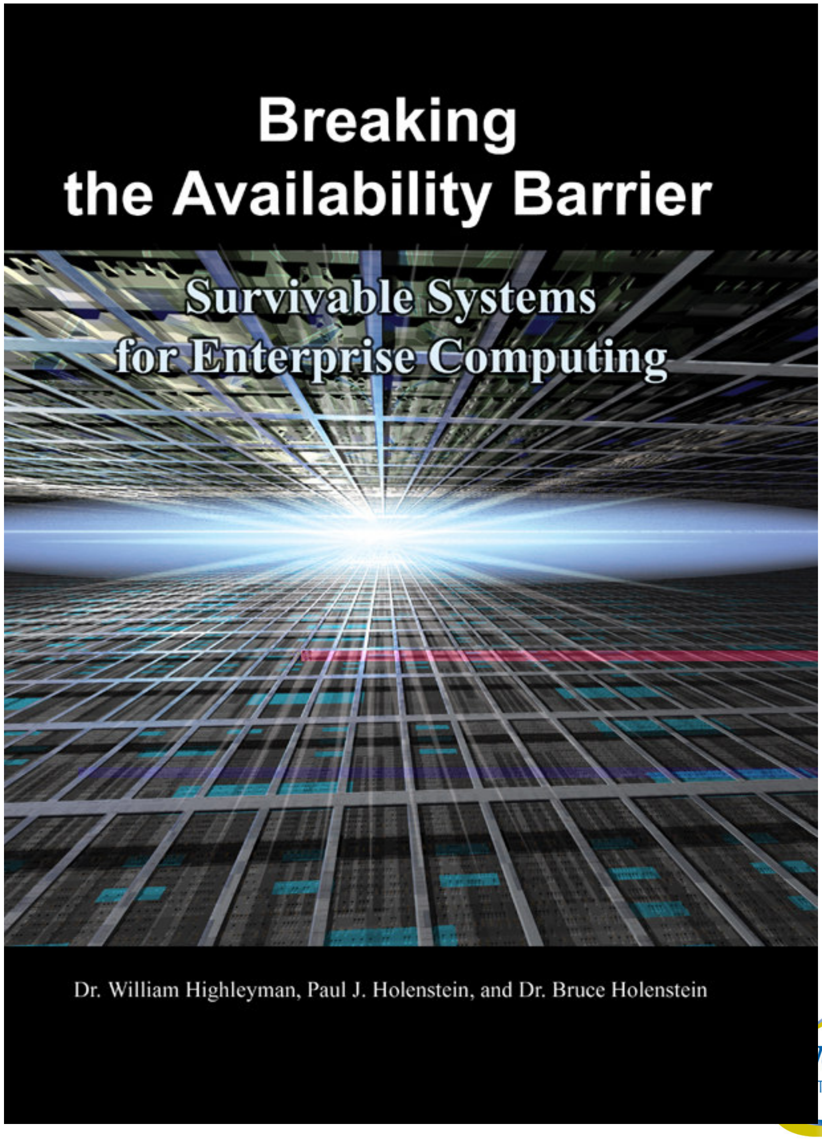
Find details in the 6-part series on Availability published in The Connection starting with the November/December 2002 issue:

Availability Part 1 – The 9s Game
Availability Part 2 – Splitting Systems
Availability Part 3 – Synchronous Replication
Availability Part 4 – The Facts of Life
Availability Part 5 – The Ultimate Architecture
Availability Part 6 – RPO versus RTO

In Conclusion

And even more in our book about active/active systems

(ISBN 1-4107-923-1)



HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

interex

shared knowledge • shared power

encompass

AN HP USER GROUP

Addendum

Availability Theory

Following is some mathematical background supporting the conclusions in this presentation.

The Sombers Group

Downtime

Failure Modes

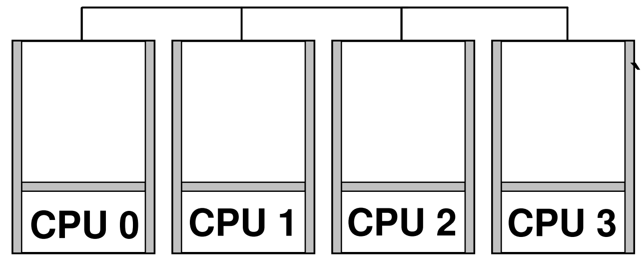
In general, if

n = number of processors in the system

then

$$\text{failure modes} = \frac{n(n-1)}{2}$$

n	failure modes	downtime
2	1	x1
4	6	x6
8	28	x28
16	120	x120



X	X		
X		X	
X			X
	X	X	
	X		X
		X	X

Downtime grows faster than the square of the system size

HP WORLD 2004
Solutions and Technology Conference & Expo

The Sombers Group

System Splitting
Reducing Downtime

If a system of *n* processors is split into *k* nodes, downtime is reduced by more than a factor of *k:*

$$\textit{Downtime reduction} = \frac{\frac{n(n-1)}{2}}{k\frac{n/k(n/k-1)}{2}} = k\frac{n-1}{n-k} > k$$

Example:

16 processors (n = 16)
4 nodes (k = 4)
5 times reduction in downtime (4x15/12)

Note: Outage is defined as failure of just one node.

System Availability

processor availability = a

probability of failure of a processor = (1-a)

probability of a dual processor failure = $(1\text{-}a)^2$

number of failure modes = f

probability of outage = $f(1\text{-}a)^2$

system availability = A = 1 – $f(1\text{-}a)^2$

Availability Example

processor availability = a = .995

probability of failure of a processor = (1-a) = .005

probability of a dual processor failure = $(1\text{-}a)^2$ = .000025

number of failure modes = f = 6 (4 processors)

probability of outage = $f(1\text{-}a)^2$ = .00015

system availability = A = 1 – $f(1\text{-}a)^2$ = .99985 ≈ 4 9s

MTBF

Mean Time Between Failure = MTBF

Mean Time to Repair = MTR

Availability = A = ______________

MTBF = MTR/(1-A)

```
ERROR: rangecheck
OFFENDING COMMAND: .buildcmap

STACK:

-dictionary-
/WinCharSetFFFF-V2TT786613C3t
/CMap
-dictionary-
/WinCharSetFFFF-V2TT786613C3t
```