Storage as a First Class Citizen in HPC Environments.

James S. Plank
University of Tennessee

CCGSC
September 9, 2010

A Personal Historical Perspective

Me – Erasure codes

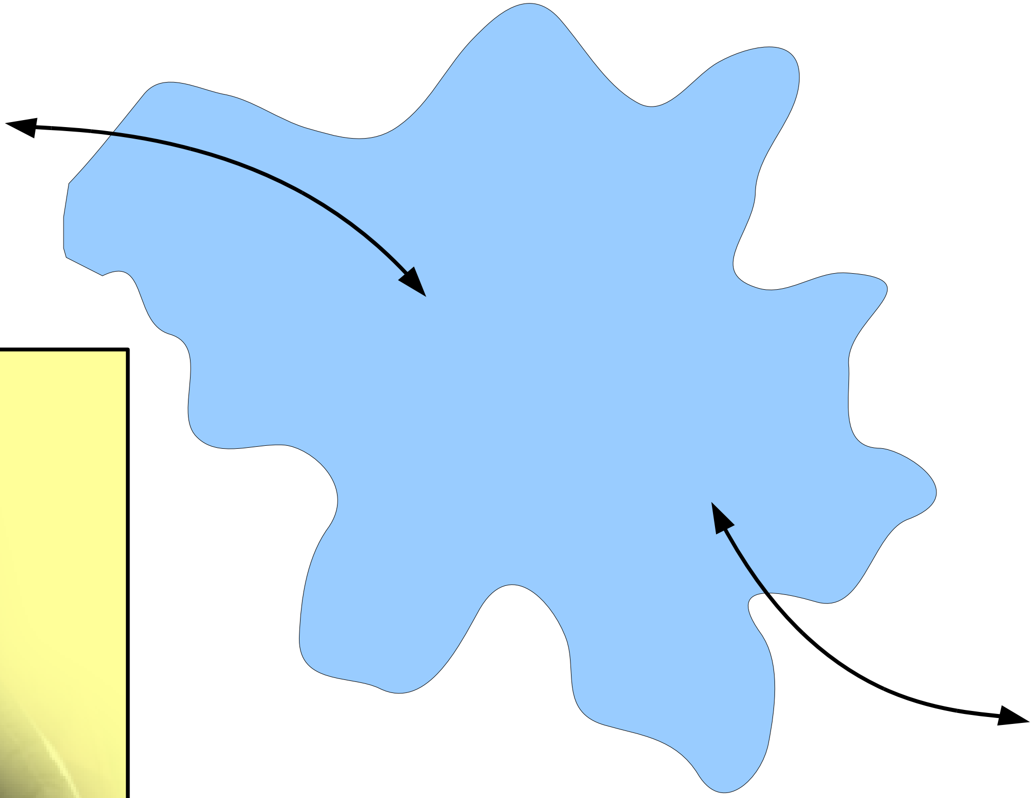
Y'all – HPC

A Personal Historical Perspective

Jim - 1987

A Personal Historical Perspective

LINDA: Parallel computing with a "tuple space."



Data tuples

Processing tuples

Tuple Space

Gelernter

Jim - 1987

A Personal Historical Perspective

LINDA: Parallel computing with a “tuple space.”

- “Linda processes aspire to know as little about each other as possible.

Gelernter

Jim - 1987

- They never interact directly with each other;

- they only deal with tuple space.”

A Personal Historical Perspective

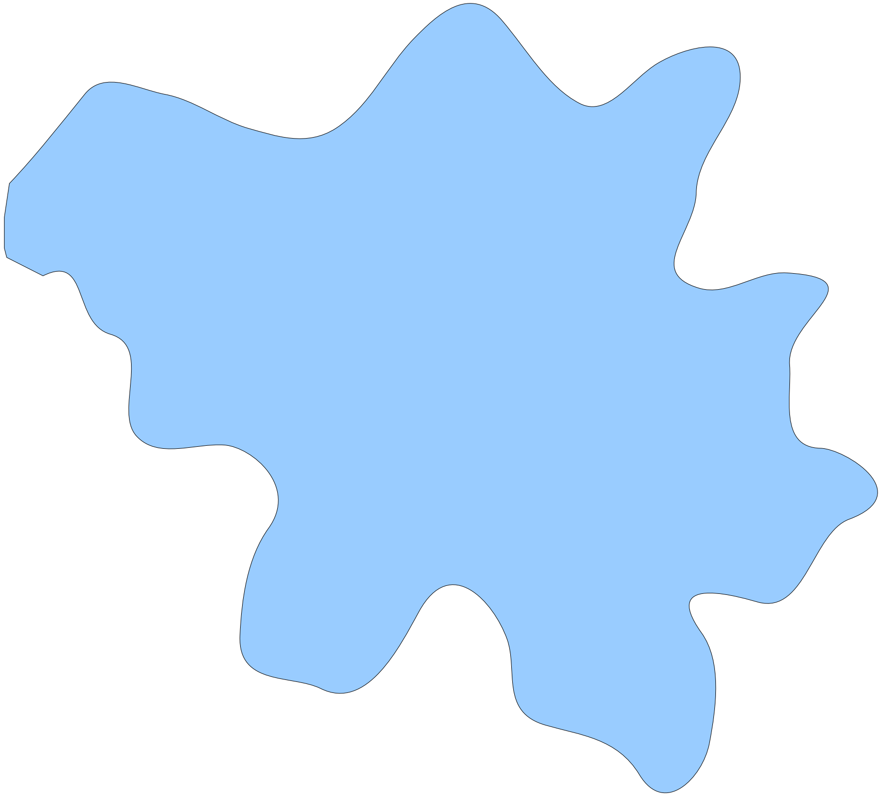
Jim - 1988

A Personal Historical Perspective

SSLS: Shared Single Level Store

Naughton

Jim - 1988

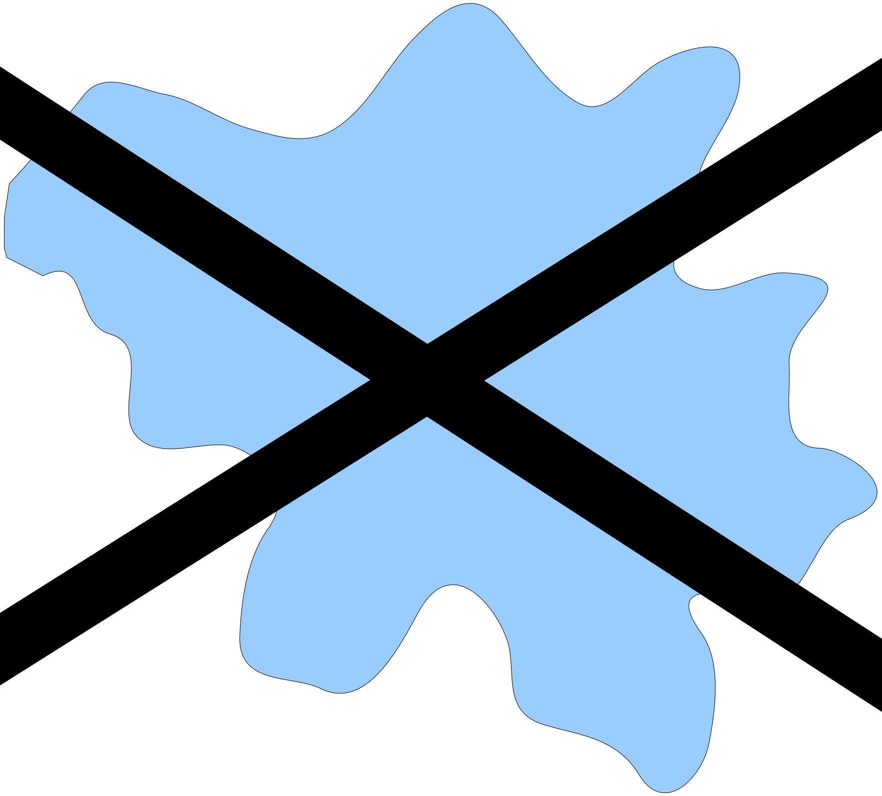


Gigantic shared, persistent address space

A Personal Historical Perspective

SSLS: Shared Single Level Store

Naughton

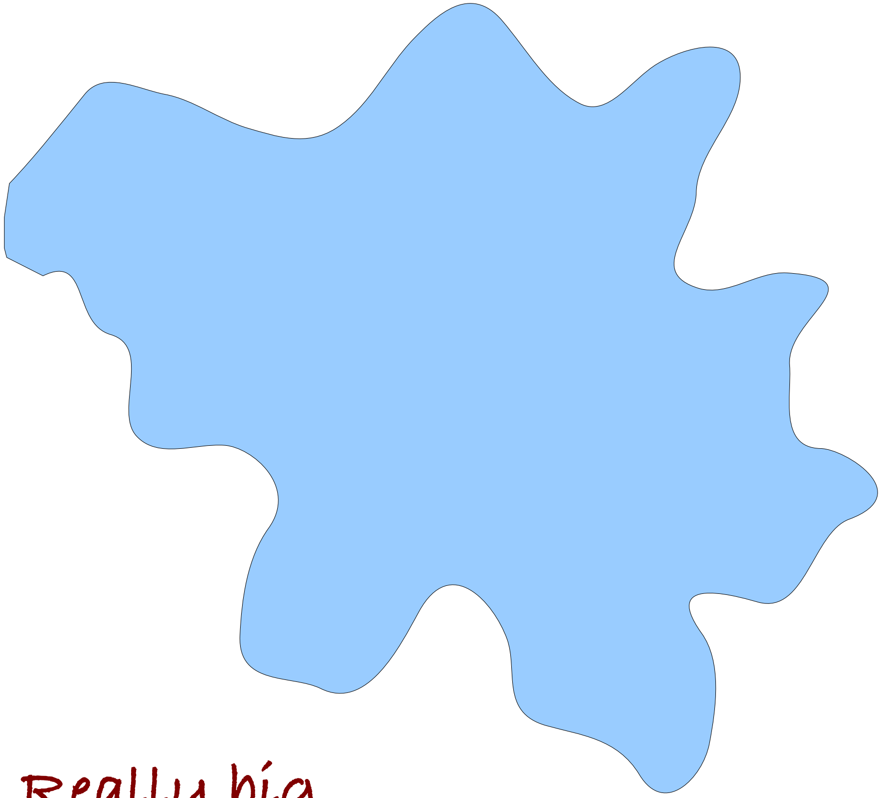


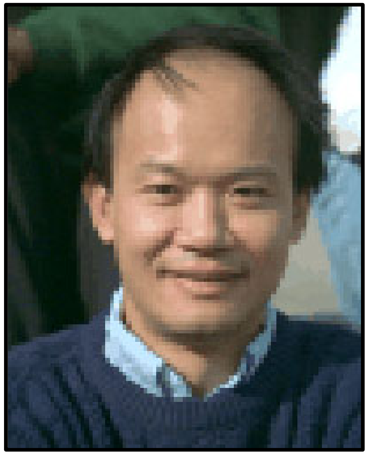
Jim - 1988

Gigantic shared, persistent address space

A Personal Historical Perspective

SVM: Shared Virtual Memory





Li

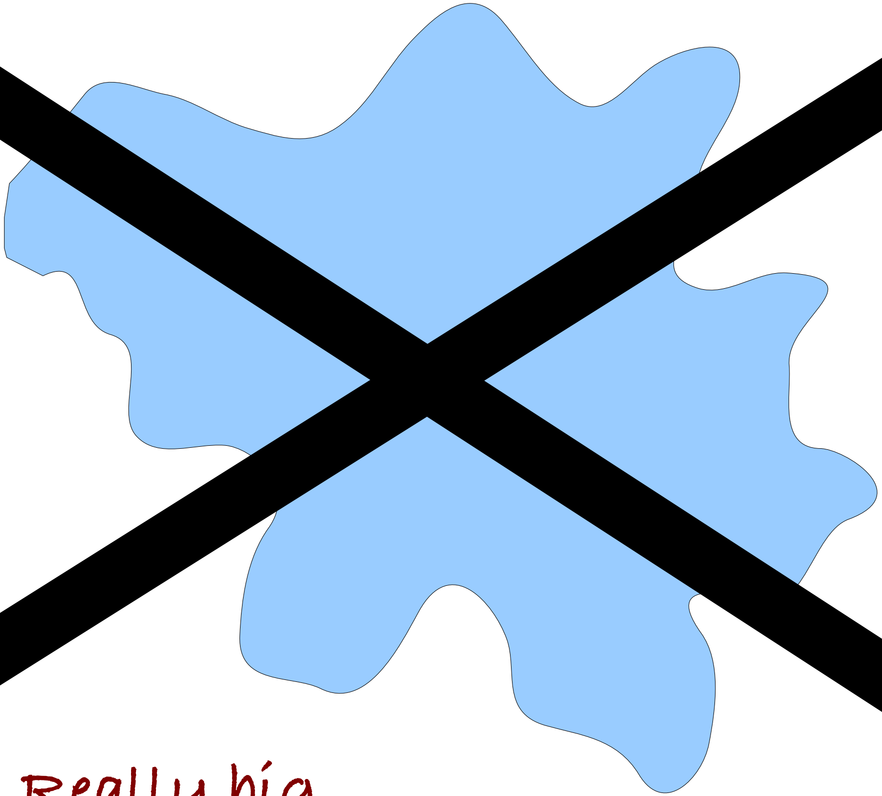
Jim - 1989

Really big

~~Gigantic~~ shared, ~~persistent~~ address space

A Personal Historical Perspective

SVM: Shared Virtual Memory



Li

Jim - 1989

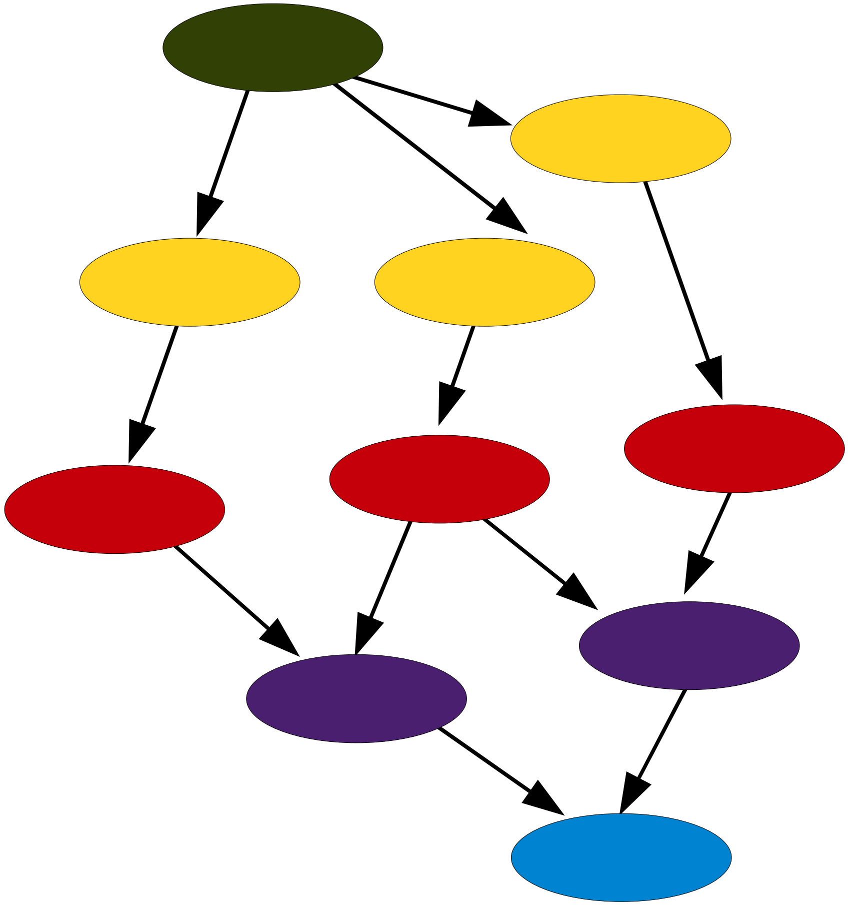
Really big

~~Gigantic~~ shared, ~~persistent~~ address space

A Personal Historical Perspective

HeNCE: Heterogeneous Network Computing Environment.

Grand Fromage



Jim - 1990

Functional Dataflow DAG Processing System

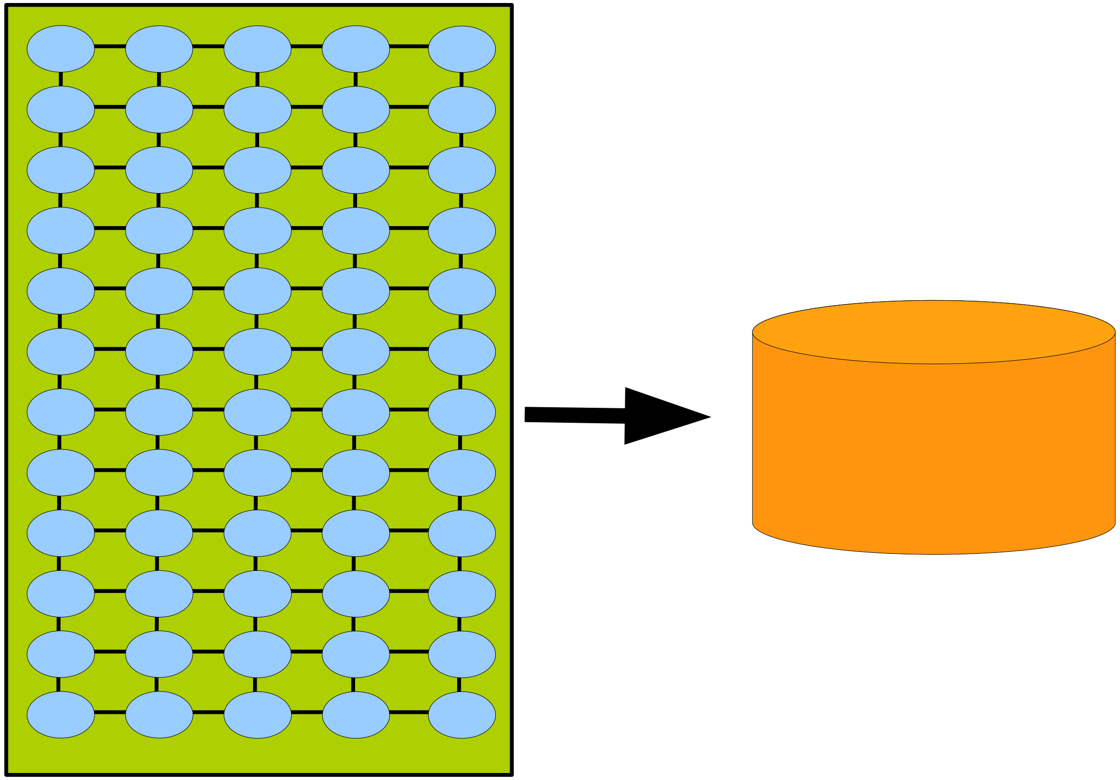
A Personal Historical Perspective

Jim - 1991-98

A Personal Historical Perspective

Mr. Checkpointing:

Jim - 1991-98



A Personal Historical Perspective

There are two major difficulties with checkpointing:

What did I learn:

1. Fighting the OS / Getting it to work.

2. Mitigating the overhead of getting all those bytes to disk.

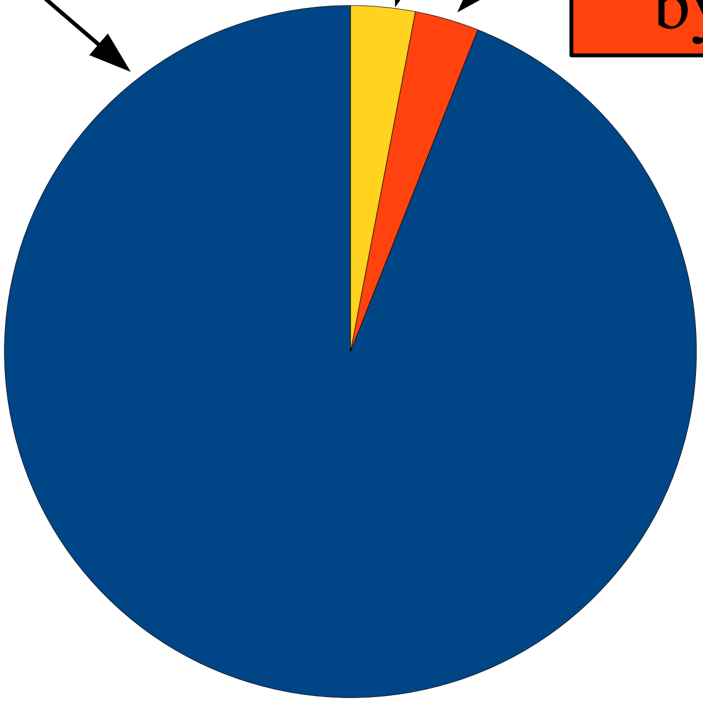
Everything else (synchronization, consistency, Lamport time, etc, etc) is in the noise.

Jim - 1991-98

A Personal Historical Perspective

Where's the research?

Jim - 1991-98



Getting it to work.

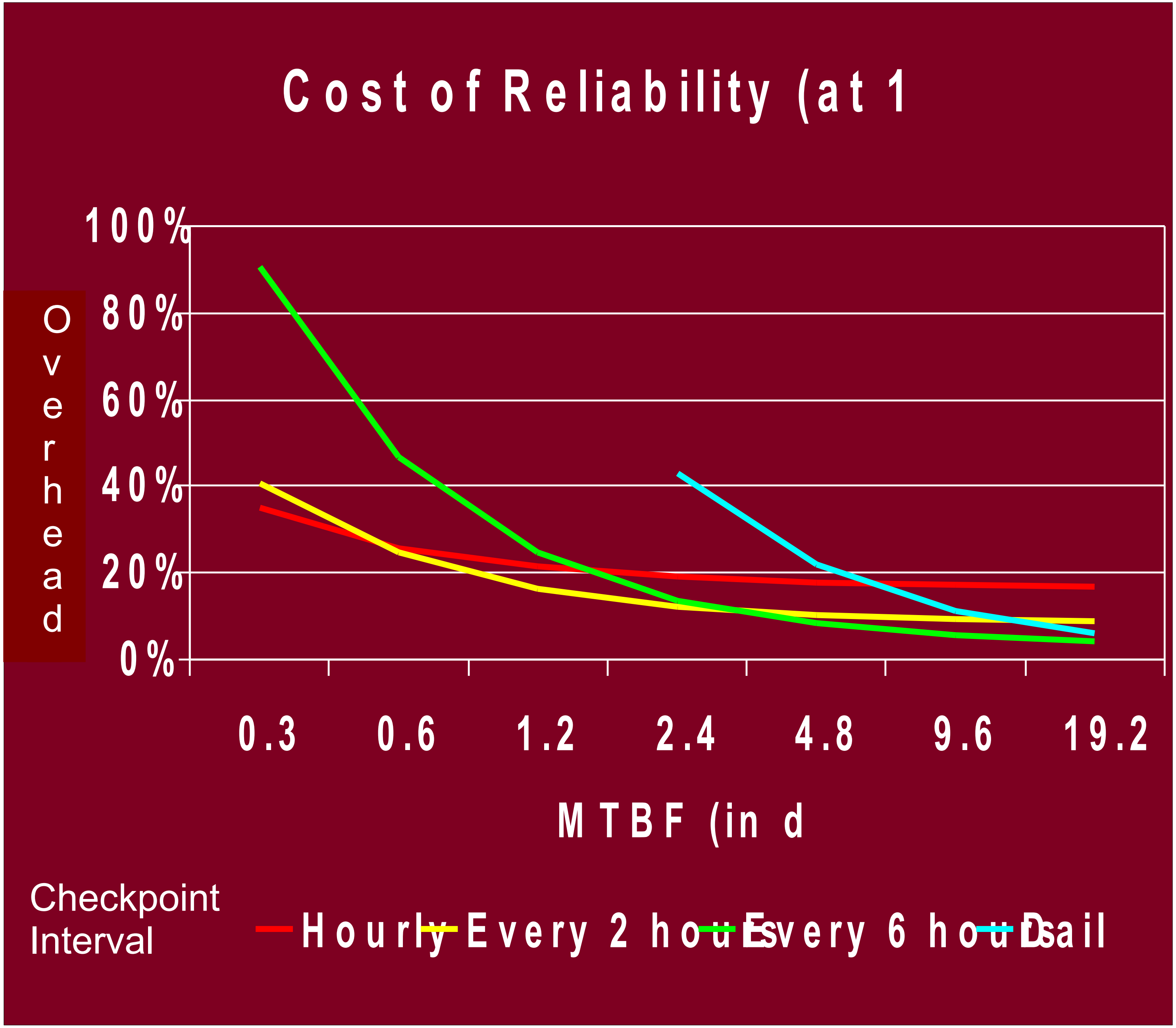
Synchronization, consistency, Lamport time, etc, etc.

Mitigating the overhead of getting all those bytes to disk.

A Personal Historical Perspective

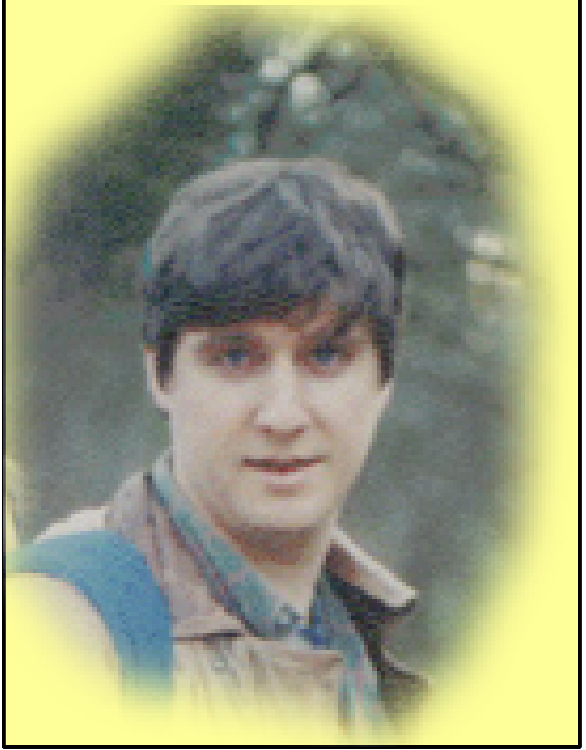
[Elnozahy/Plank 2004]

Jim - 1991-98



A Personal Historical Perspective

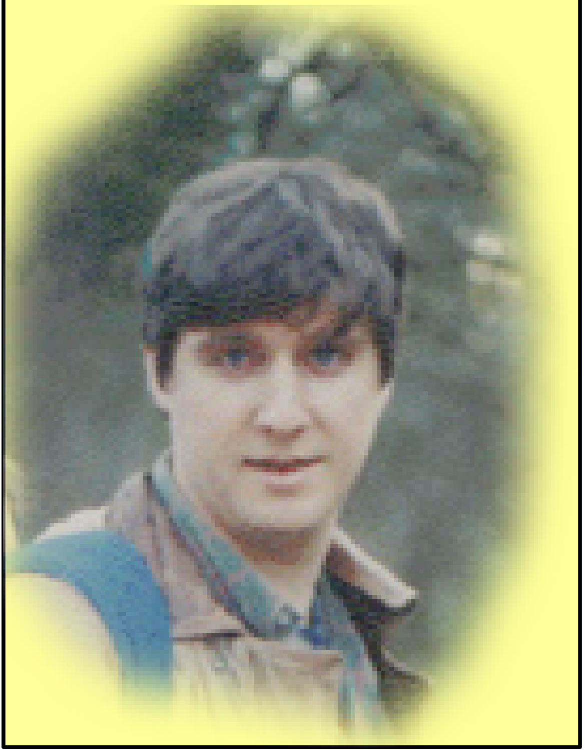
Jim – 1999

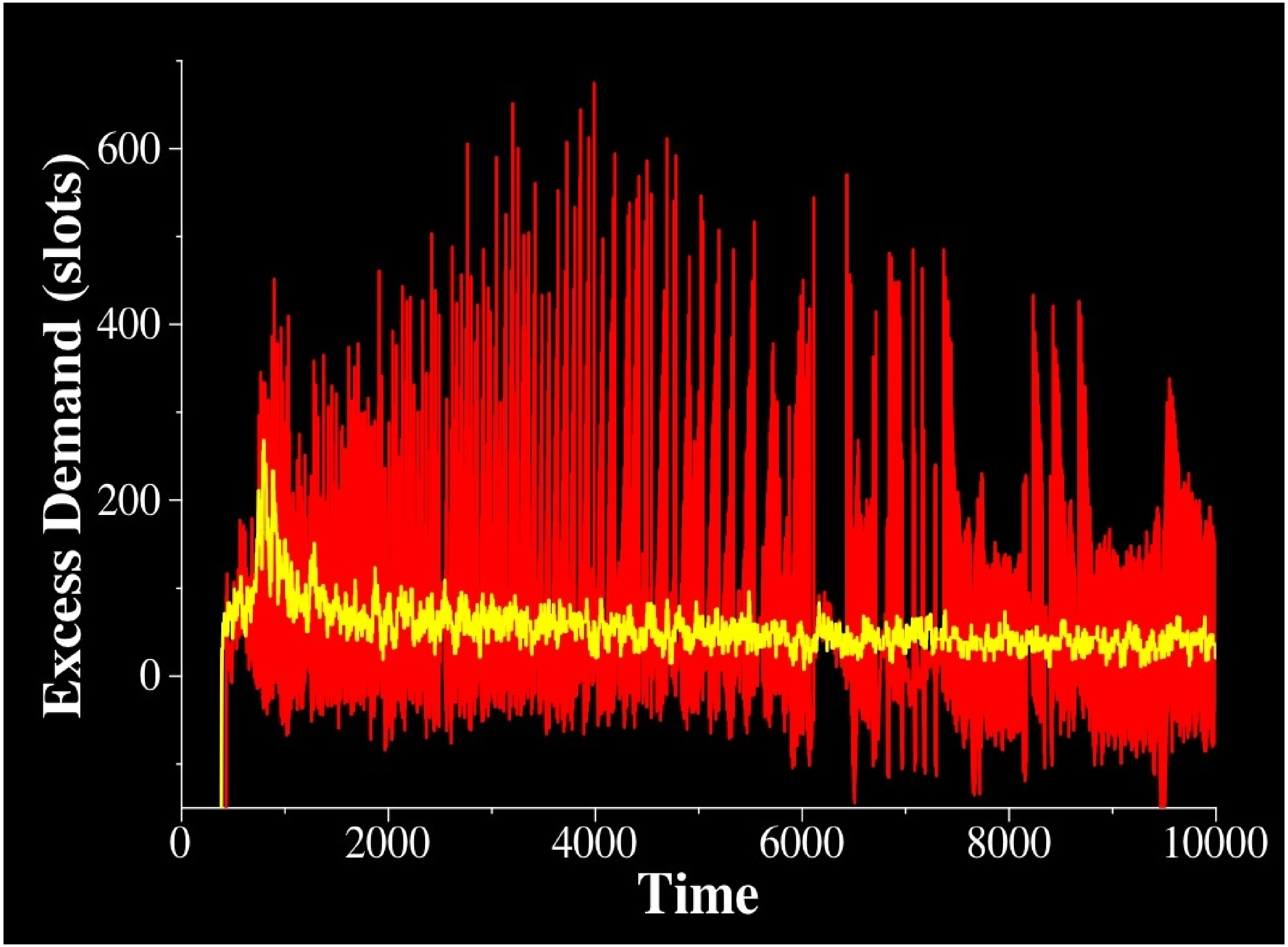


A Personal Historical Perspective

G-Commerce: Brief Foray into Grid Computing

Jim – 1999

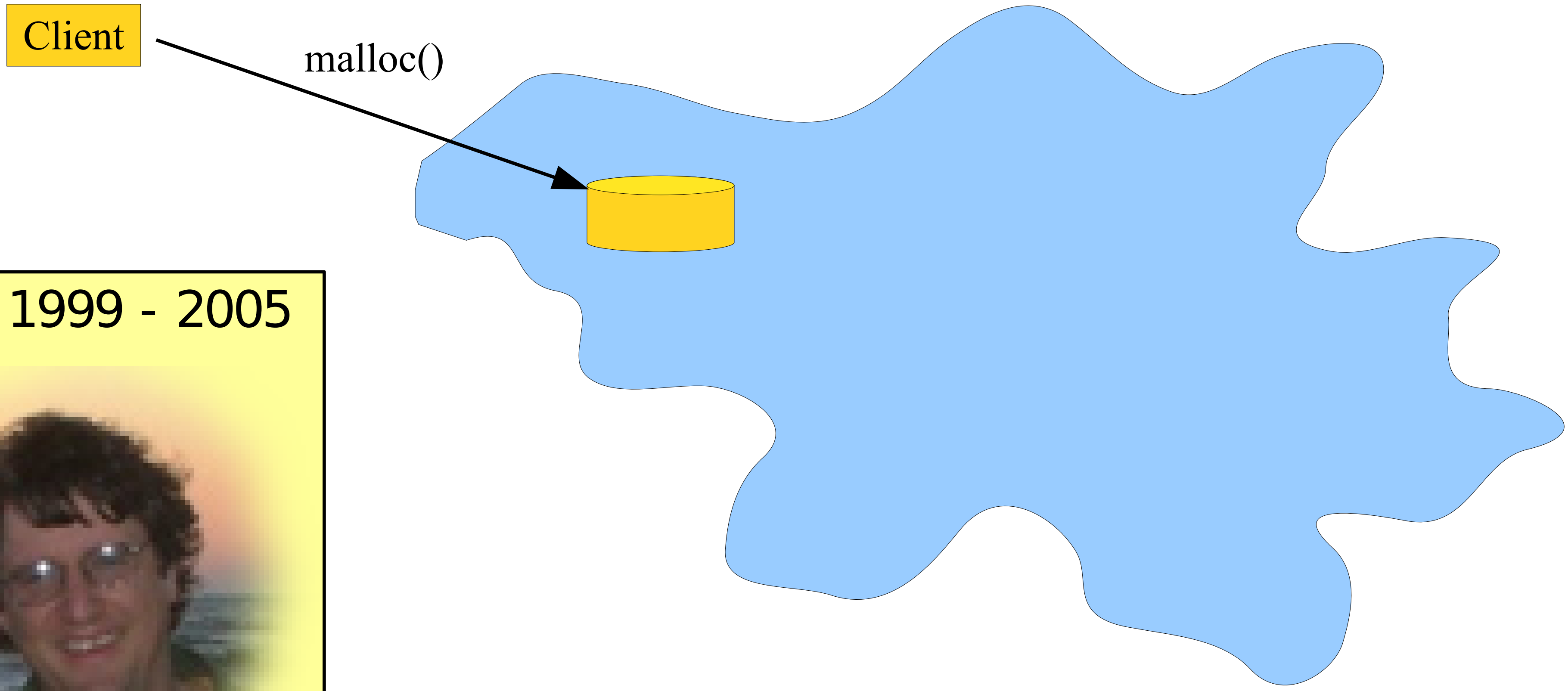




A Personal Historical Perspective

IBP: Internet Backplane Protocol (Logistical Networking)

w/ Micah Beck

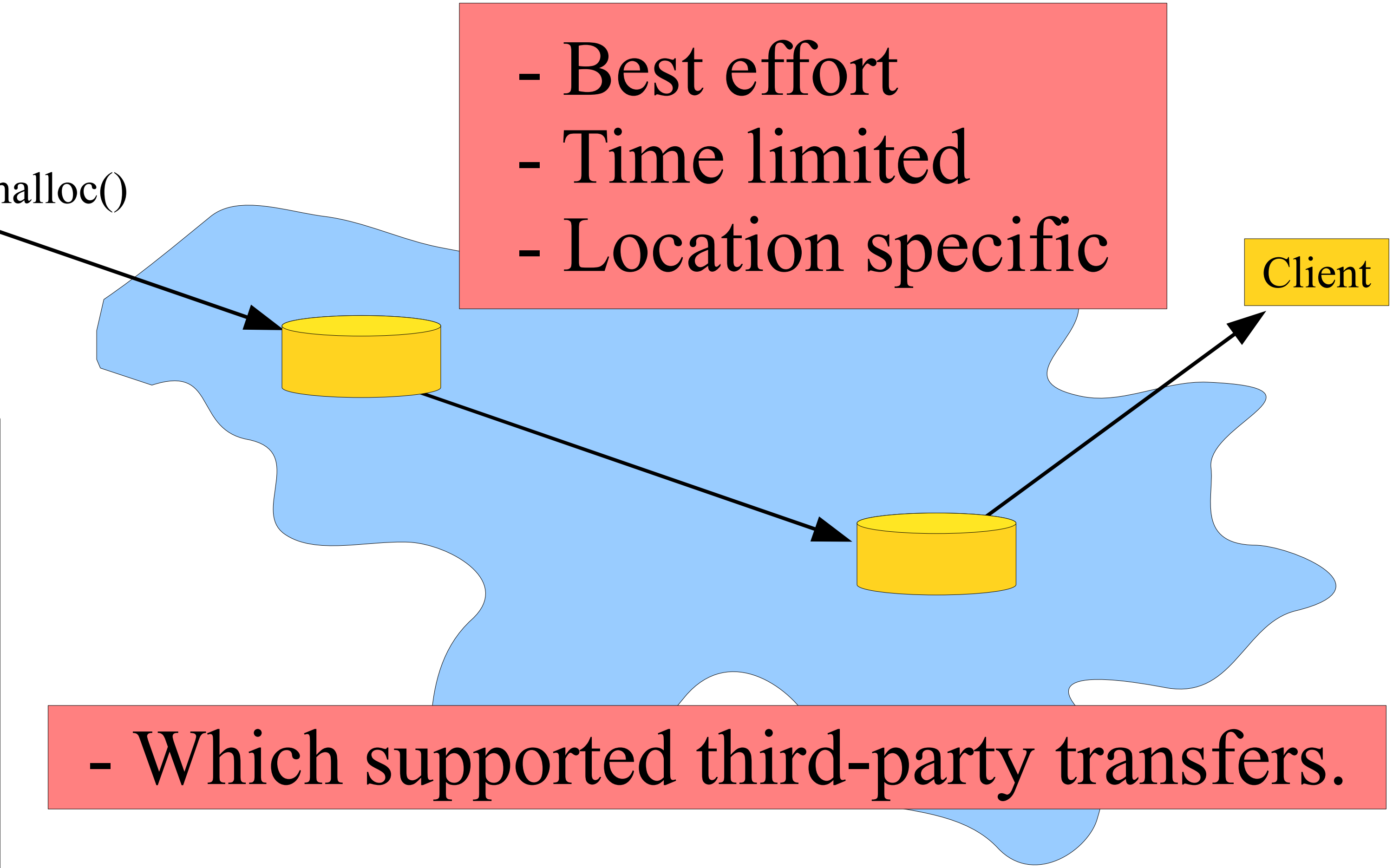


Jim – 1999 - 2005

A Personal Historical Perspective

IBP: Internet Backplane Protocol (Logistical Networking)

w/ Micah Beck



- Best effort
- Time limited
- Location specific

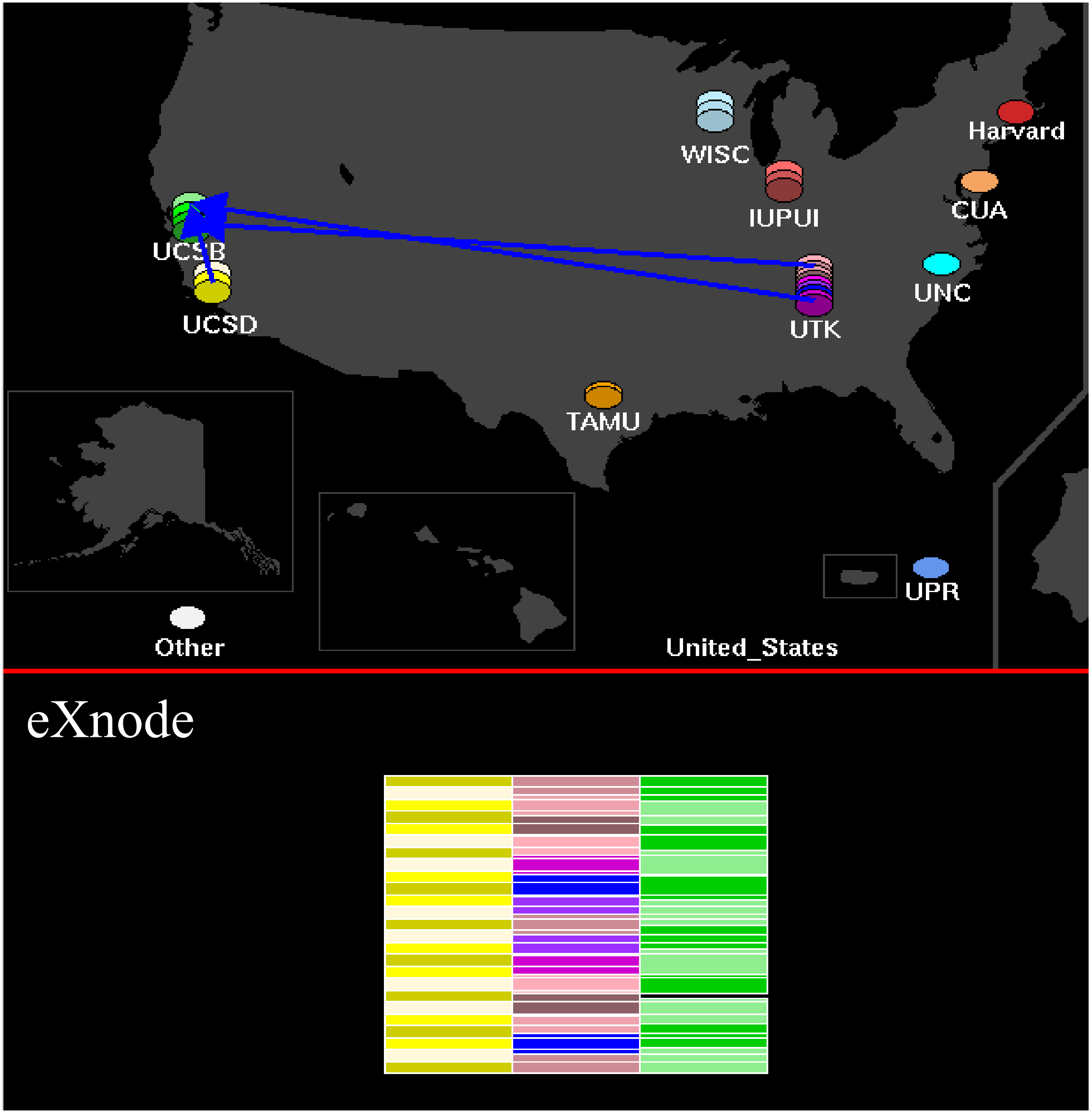
- Which supported third-party transfers.

Jim – 1999 - 2005

A Personal Historical Perspective

IBP gave data a place to "live" on the network, perhaps moving from site to site.

Jim – 1999 - 2005



A Personal Historical Perspective

Into the land of erasure coding.

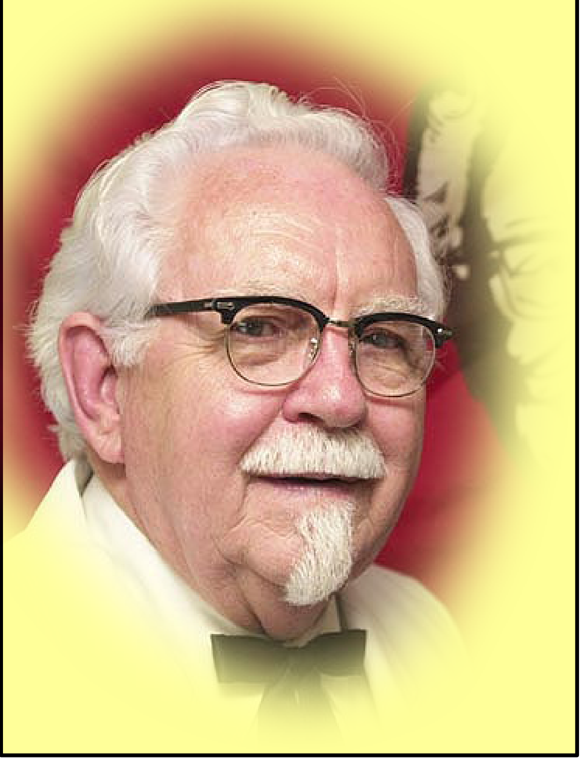
Jim – 2005 - ???

I won't bore you with it.

A Personal Historical Perspective

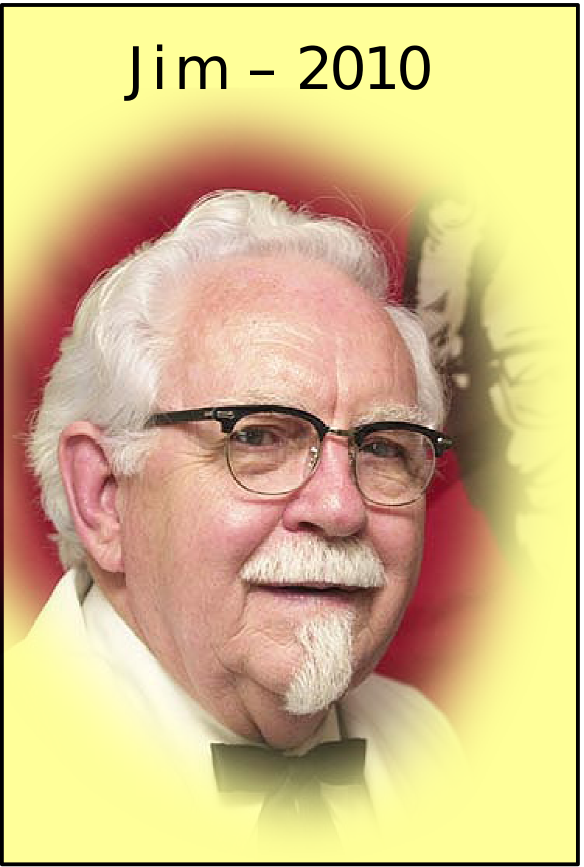
But there's more...

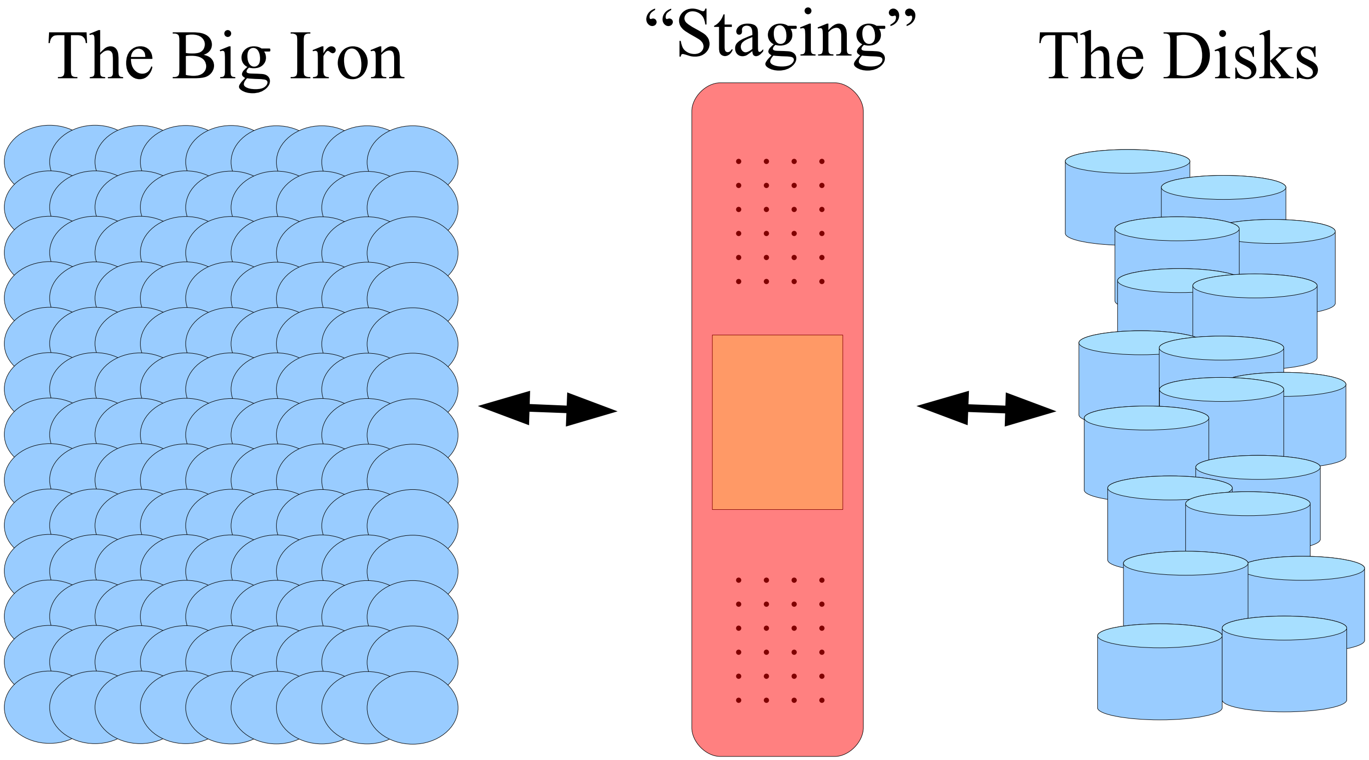
Jim – 2010



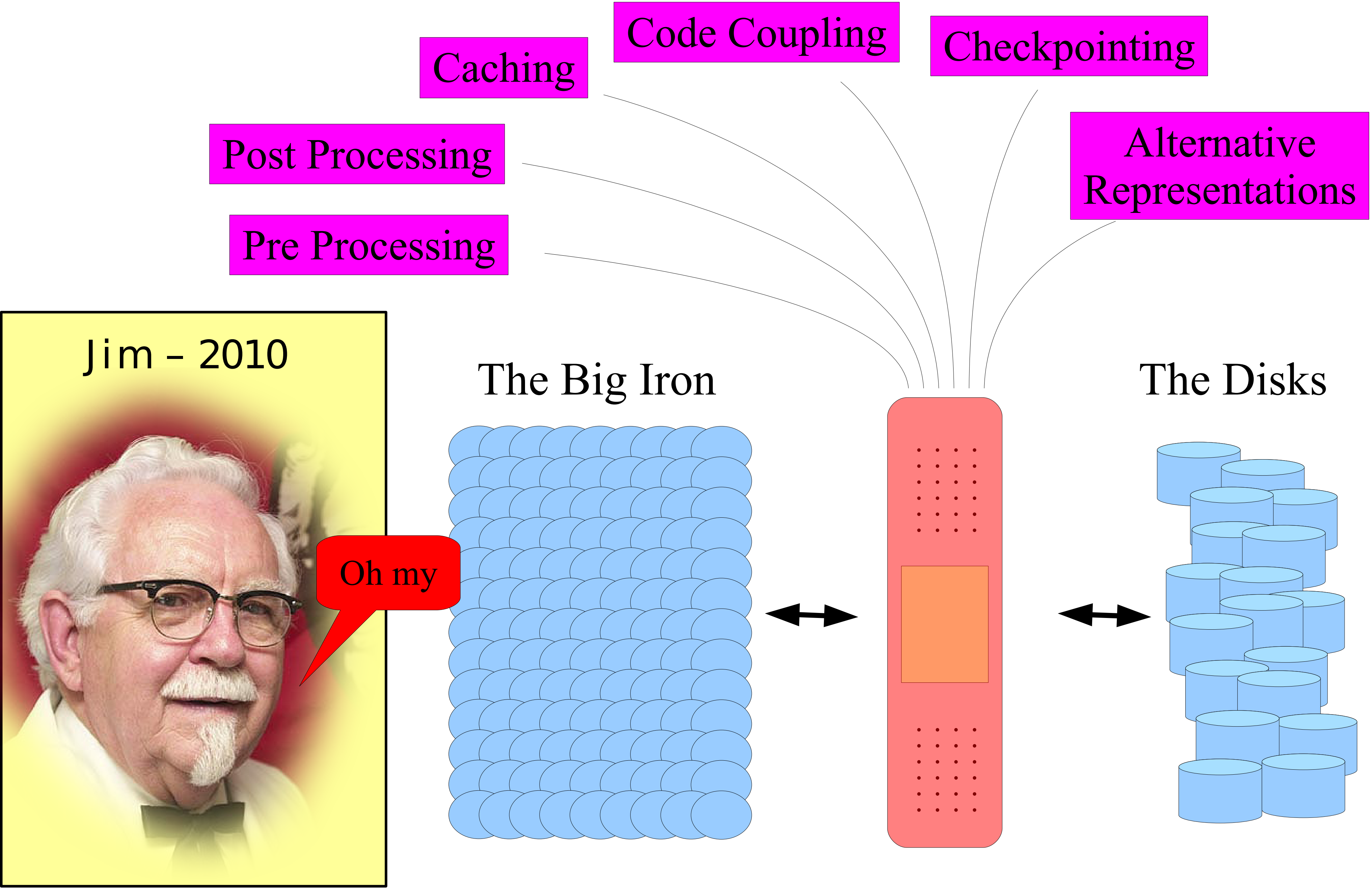
A Personal Historical Perspective

2010 Meeting on
Staging for HPC

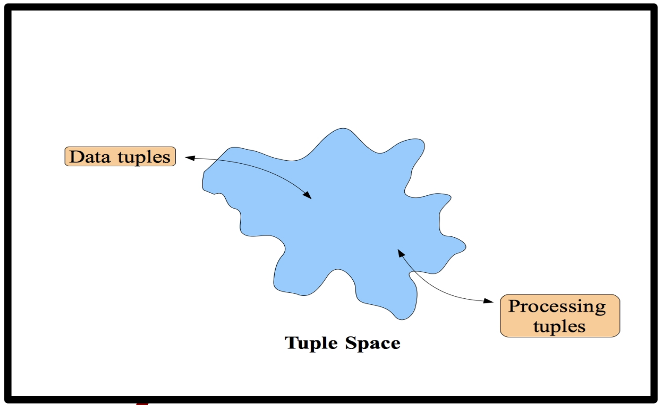




A Personal Historical Perspective

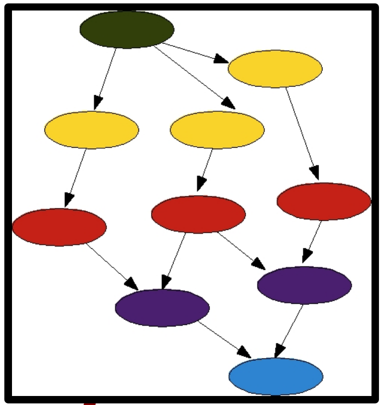


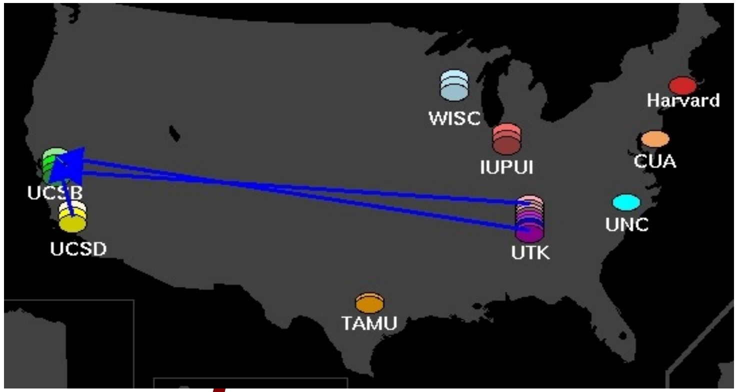
What do we make of all this?

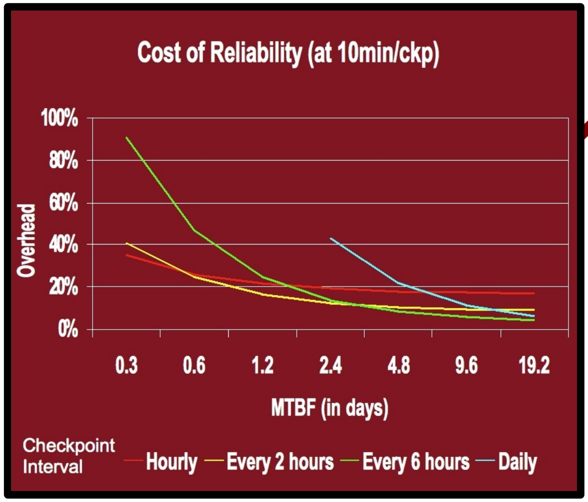

Data tuples

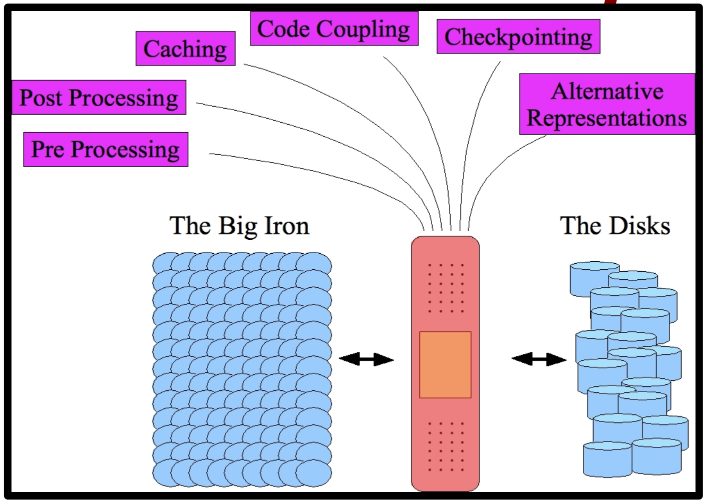
Processing tuples

Tuple Space








Caching

Code Coupling

Checkpointing

Post Processing

Alternative Representations

Pre Processing

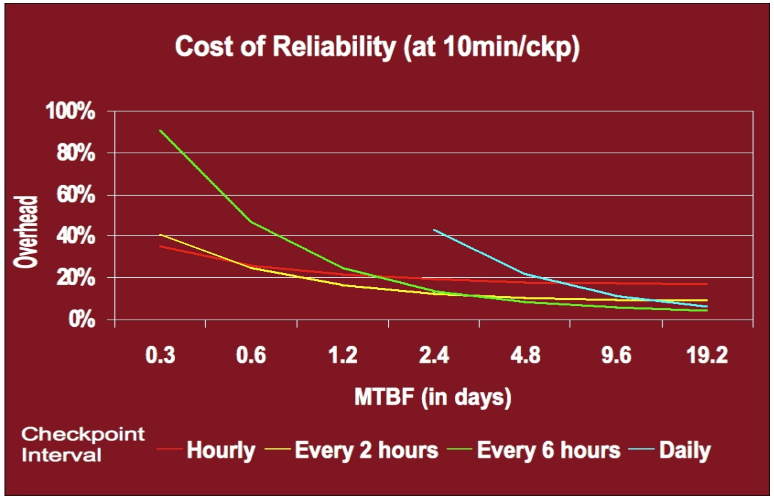
The Big Iron

The Disks

What do we make of all this?

1. Checkpointing Sucks.

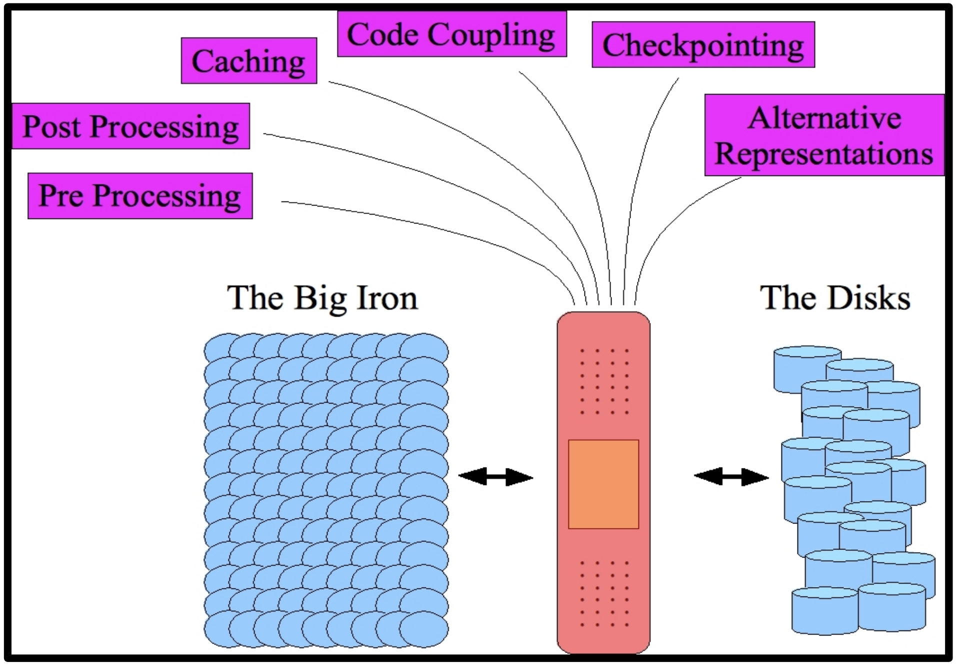
- Slow
- Inelegant
- Swamps disks and networks to store gigantic files that are almost never read.
- Enables you to perform “bad fault-tolerance.”
- Is a manifestation that something is wrong.



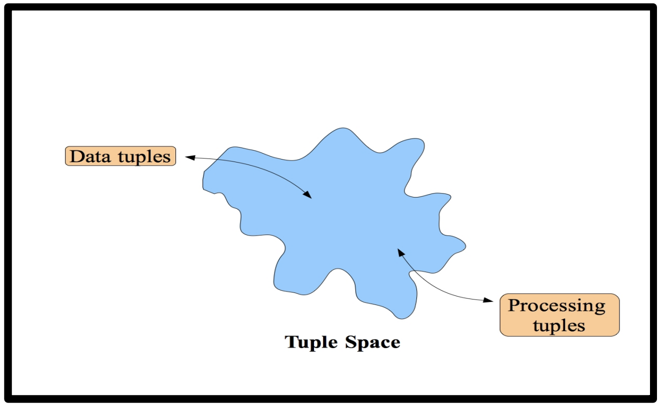
What do we make of all this?

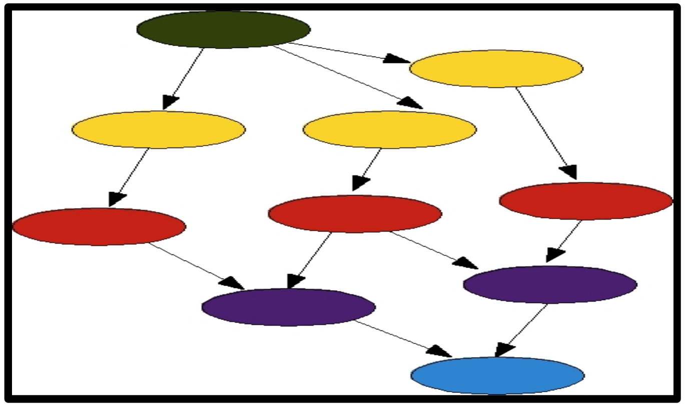
2. Band-Aids Are Only Temporary Solutions

- Non-reusable
- Cover the wounds but don't address the root cause
- Are a manifestation that something is wrong.



What do we make of all this?



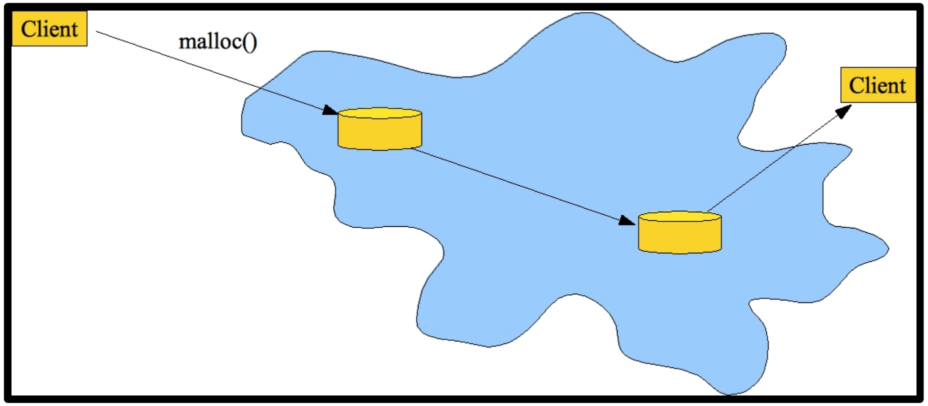


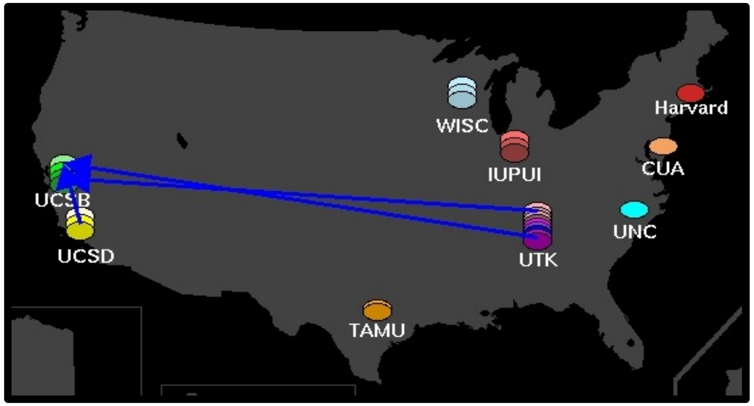
3. Saving State Sure is Attractive

- Lets you reason about programs
- (In theory) lets balance load
- Allows fault tolerance to fall out naturally

- However, it's really difficult to do.
- This is why the MPI model throws it in the trash can.

What do we make of all this?

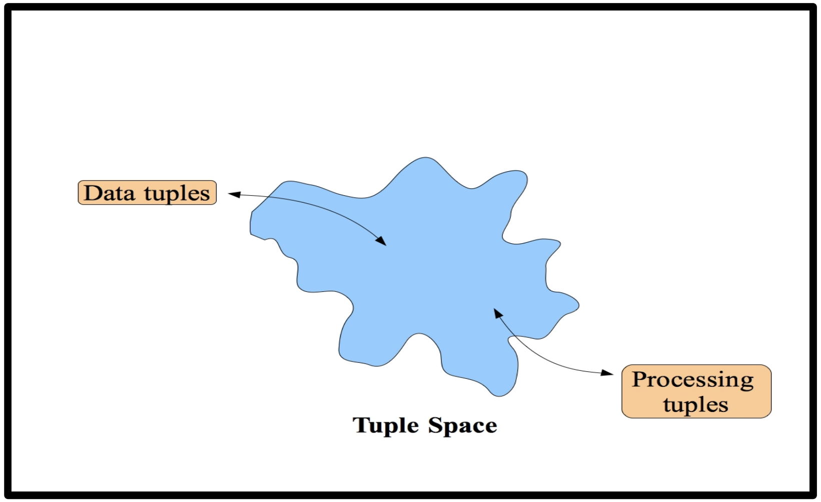




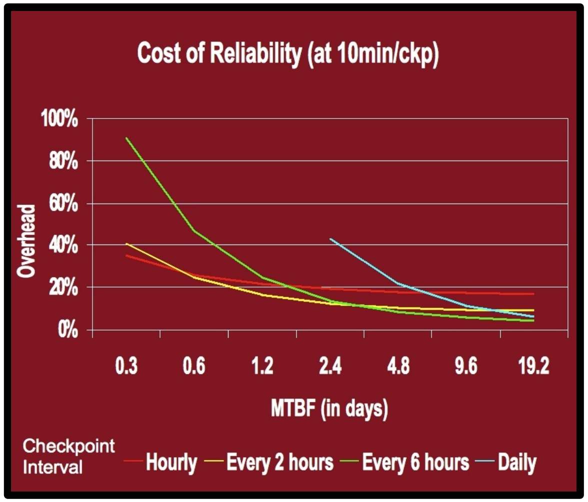
4. I Still Think IBP is Pretty Cool & That There Are Lessons To Be Learned From It

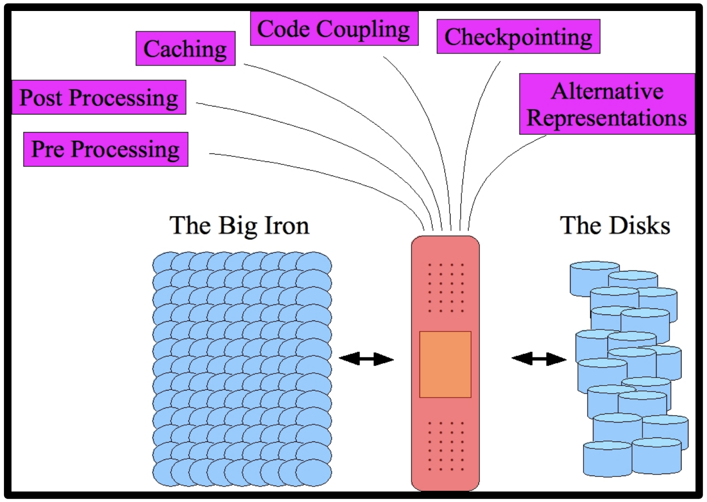
- Why do we constrain our view of storage as either the file or the memory segment?
- Why is storage either permanent or limited by program lifetime?
- Why do we jettison best-effort storage resources?
- Why don't we manage the location of storage?

What do we make of all this?



Why are storage and processing not equal first-class citizens in HPC?

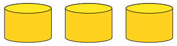




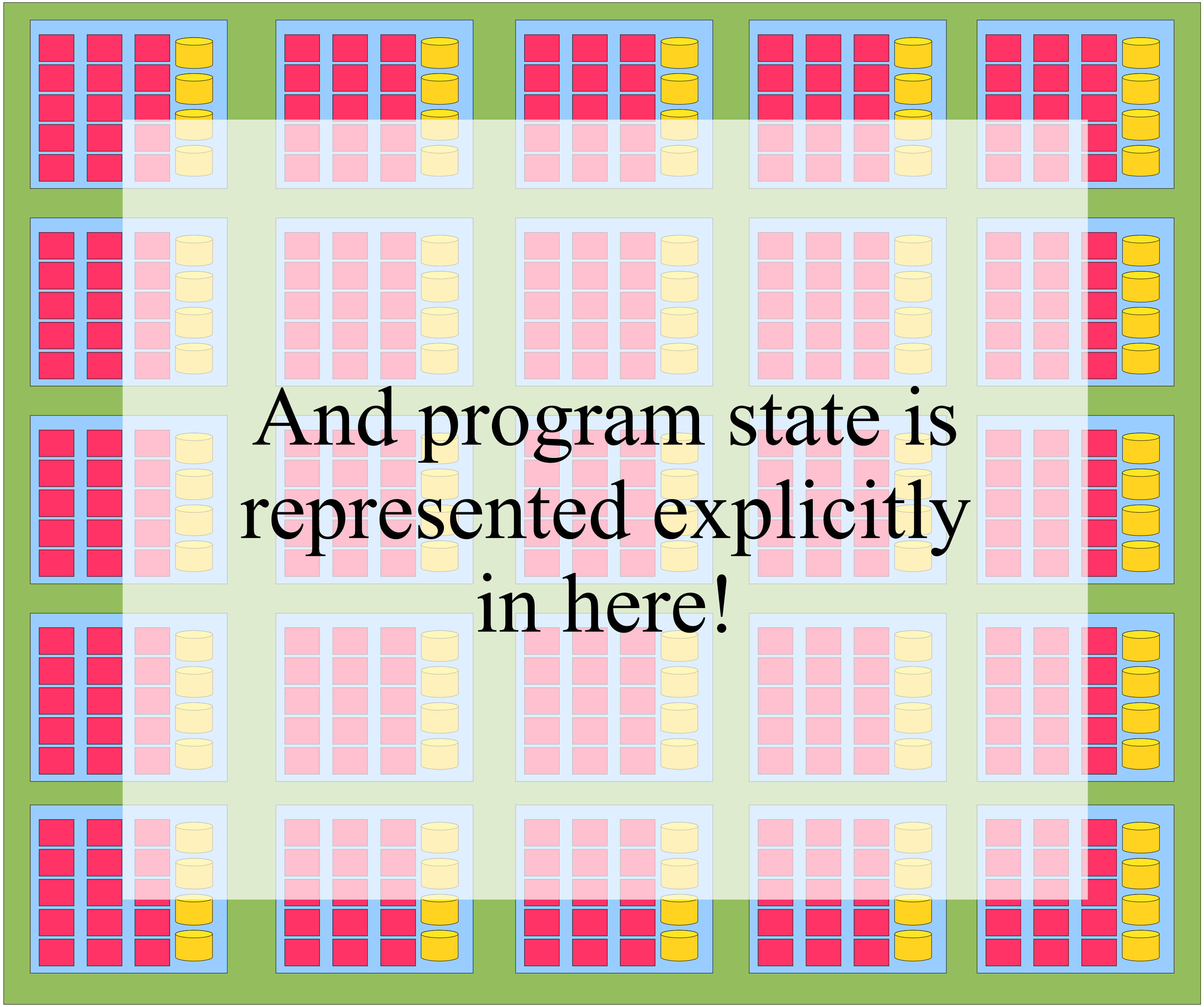
When I Close My Eyes and Dream ......

The
Big
Iron
looks
like
this.

And these
guys:



are promoted
to first
class
citizens.

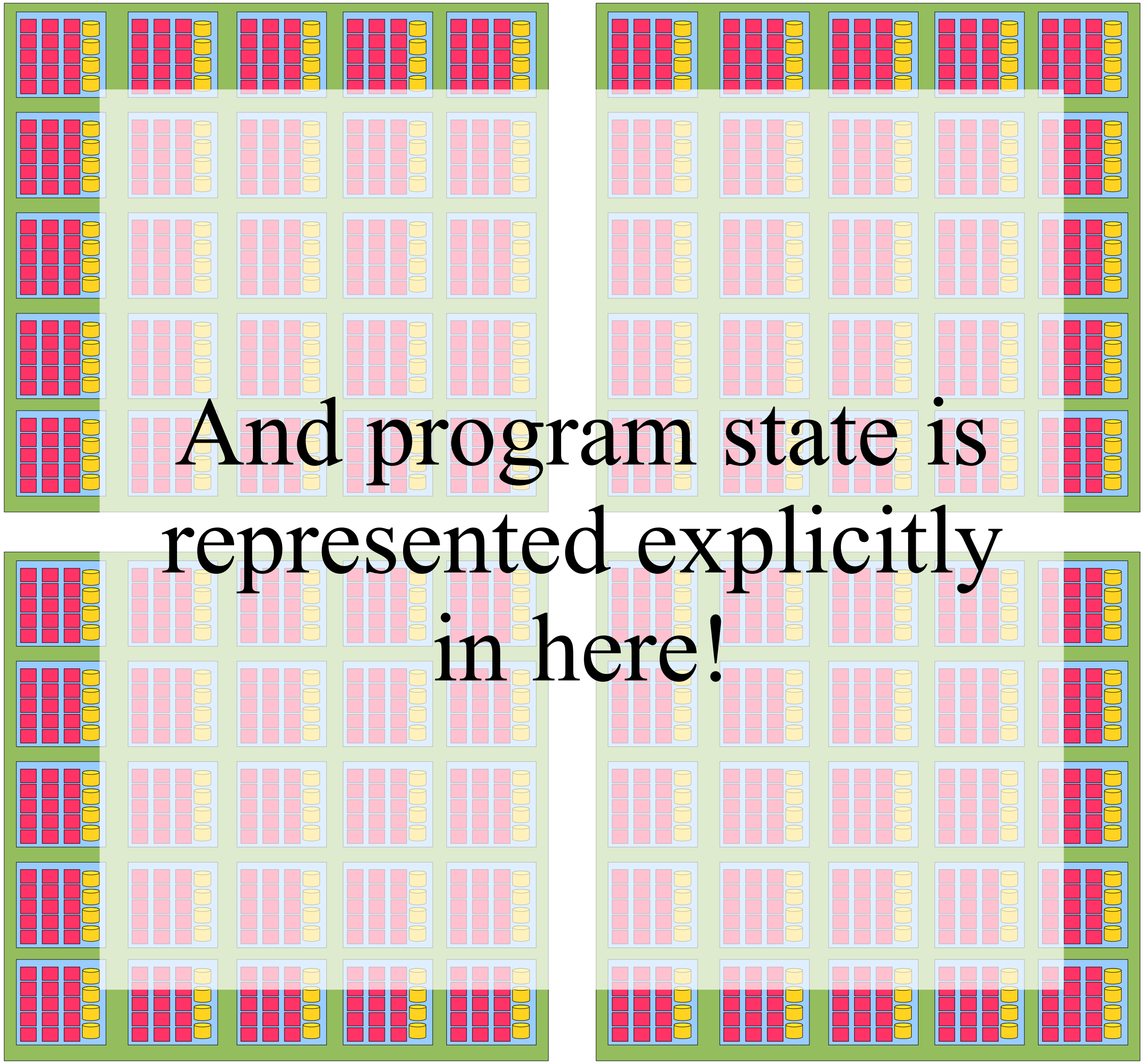


And program state is represented explicitly in here!

When I Close My Eyes and Dream ......

And these guys compose seamlessly.

Over extremely wide areas ....



And program state is represented explicitly in here!

When I Close My Eyes and Dream ......

And the Eagles win the Super Bowl... Every Year...

And I retire to that mansion in Capri...

And then I wake up and go back to studying erasure codes.

Me – Erasure codes

Y'all – HPC

Storage as a First Class Citizen in HPC Environments.

James S. Plank
University of Tennessee

CCGSC
September 9, 2010