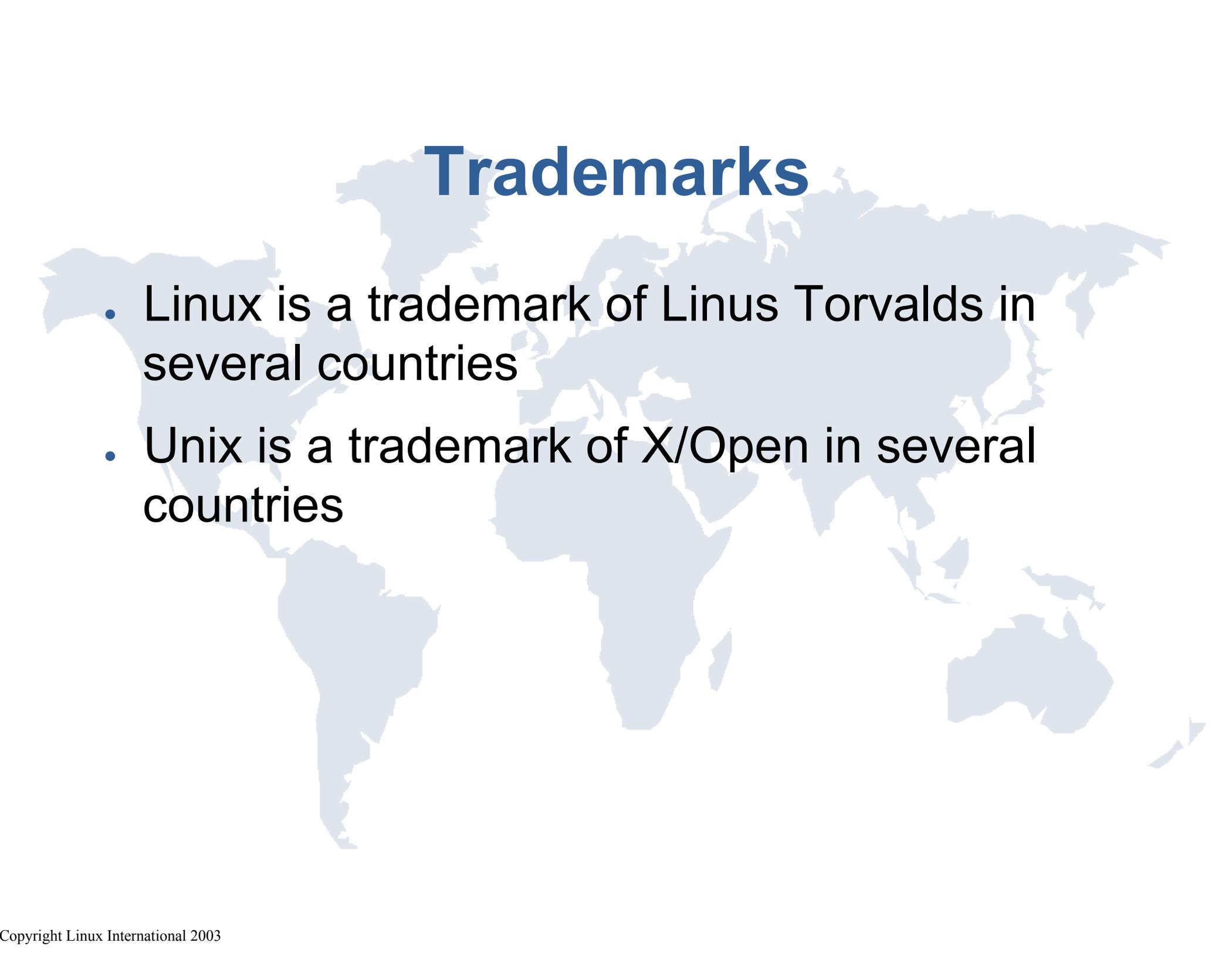


# **Linux: A Continuum From PDAs to Supercomputers**

by  
Jon "maddog" Hall  
Executive Director  
Linux International

# Trademarks



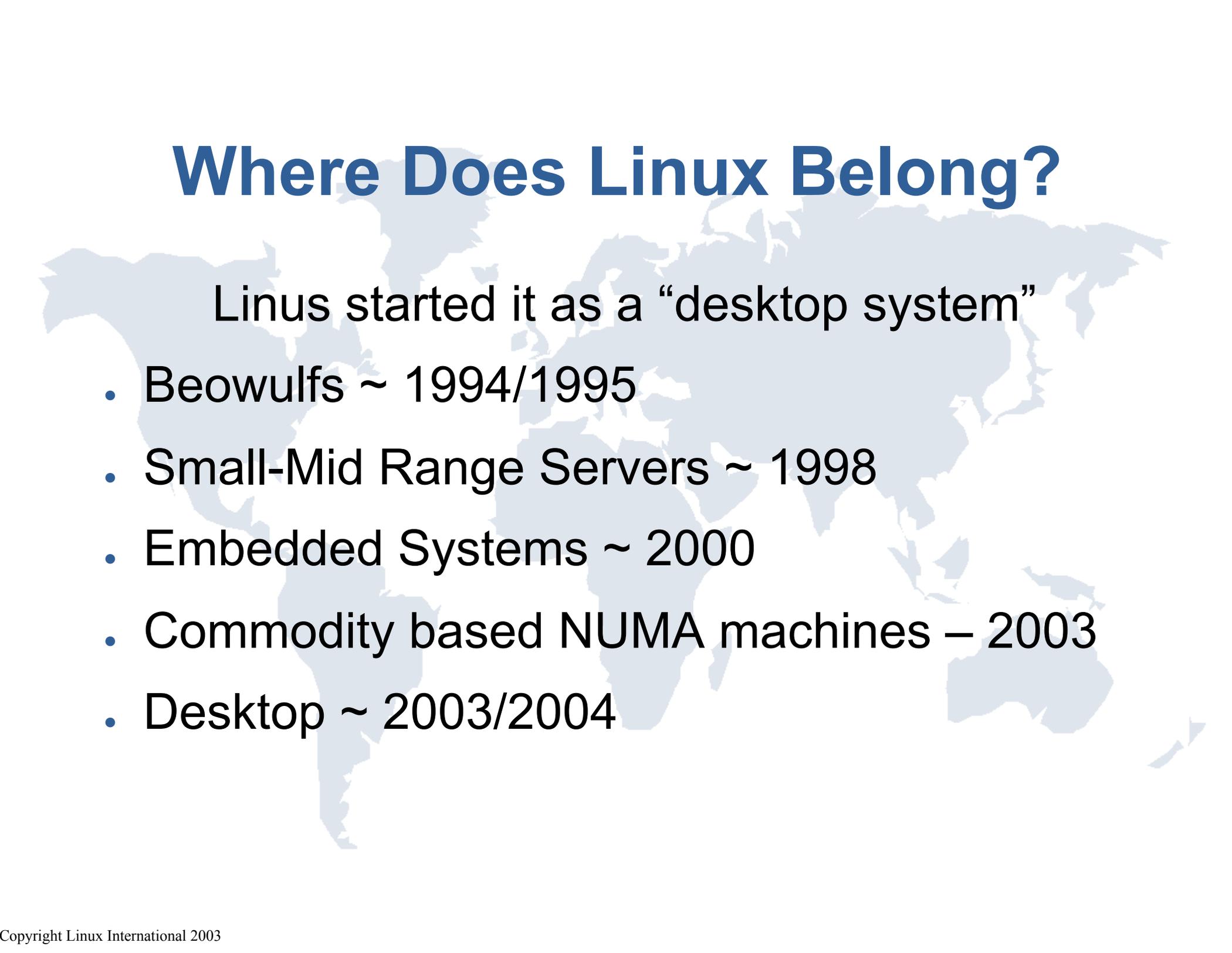
- Linux is a trademark of Linus Torvalds in several countries
- Unix is a trademark of X/Open in several countries

# Who Am I?

## And Why Should You Listen To Me?

- Thirty-four years in the computer industry
  - Programmer, Systems Analyst, Systems Administrator, Product Manager, Technical Marketing Manager
- Seven years teaching experience at university level
  - Operating Design, Compiler Design, Database Design
- Many operating systems
- Large and small companies
- Vendor and customer

# Where Does Linux Belong?



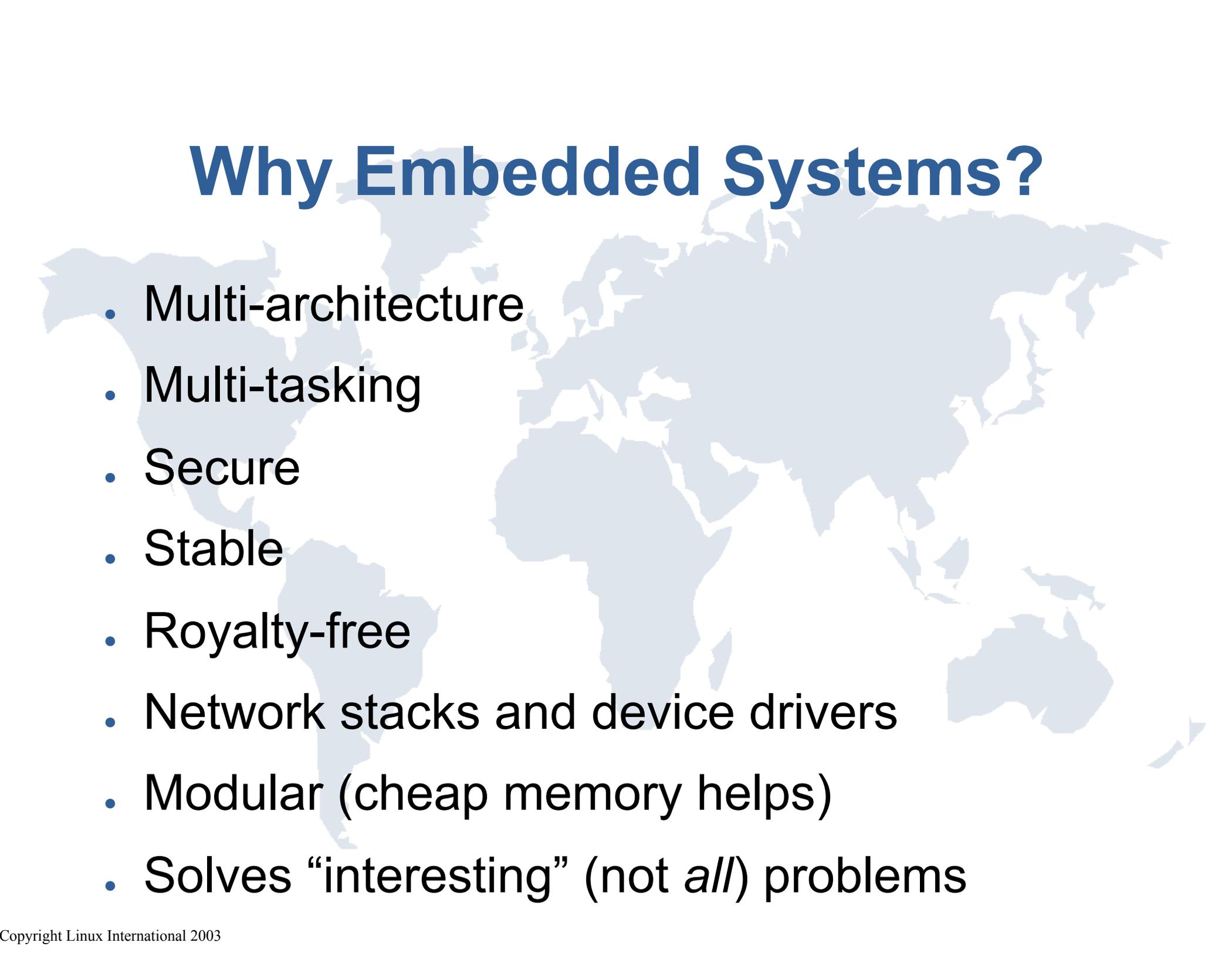
Linus started it as a “desktop system”

- Beowulfs ~ 1994/1995
- Small-Mid Range Servers ~ 1998
- Embedded Systems ~ 2000
- Commodity based NUMA machines – 2003
- Desktop ~ 2003/2004

# Why Small-Mid Range Servers?

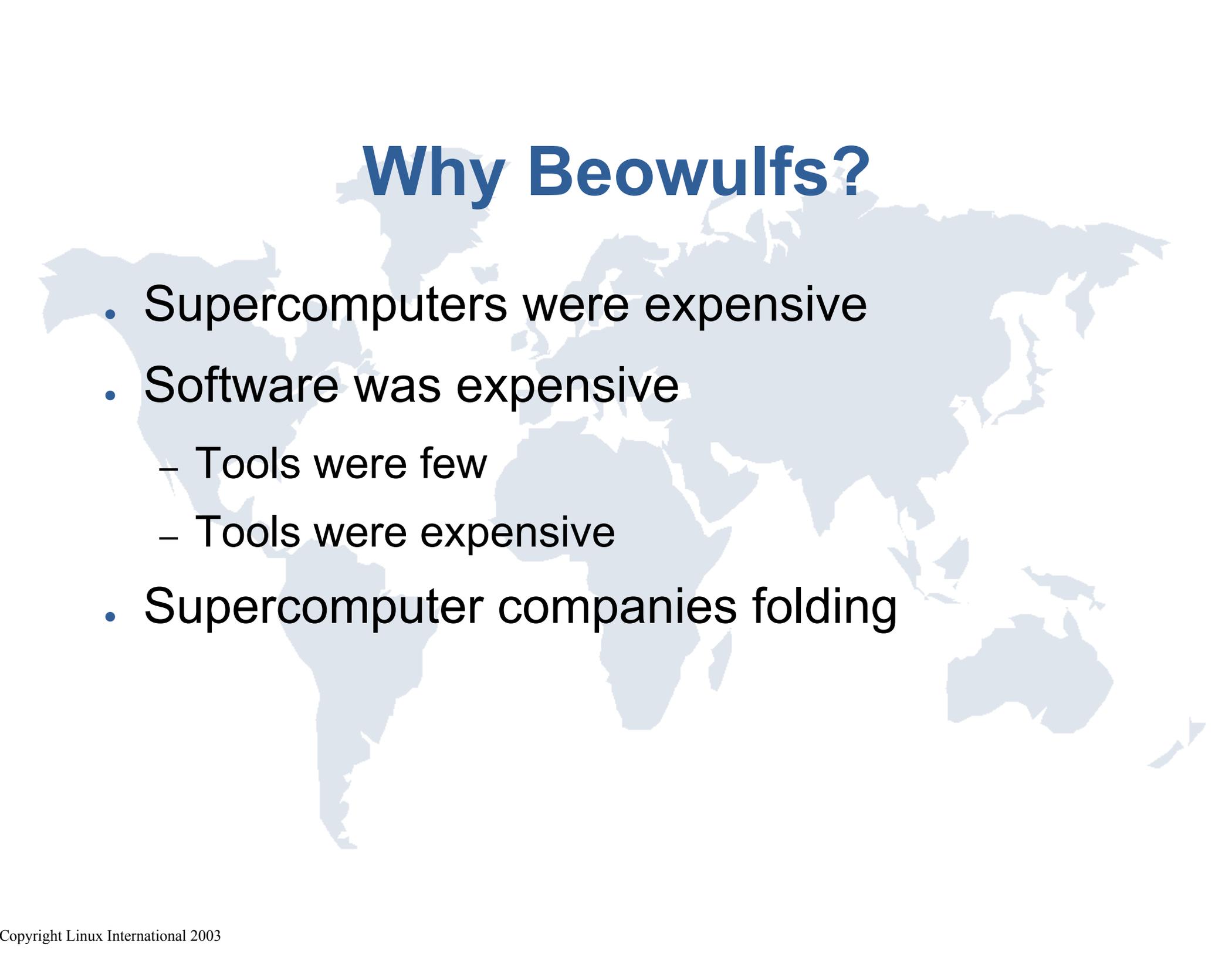
- Cheaper than proprietary hardware and software
- More stable than NT (and cheaper too)
- Had source code
- Perfect ISP machine
- Databases ported in 1998

# Why Embedded Systems?



- Multi-architecture
- Multi-tasking
- Secure
- Stable
- Royalty-free
- Network stacks and device drivers
- Modular (cheap memory helps)
- Solves “interesting” (not *all*) problems

# Why Beowulfs?



- Supercomputers were expensive
- Software was expensive
  - Tools were few
  - Tools were expensive
- Supercomputer companies folding

# What Types of Problems?

- *Image rendering*
- *Image recognition*
- Weather forecasting
- Global warming
- Modeling and meteors
- Resource prospecting through seismic imaging
- Data Mining
- Genome research (MySQL)
- Searching document image databases
- Molecular dynamics simulations
- Virtual Reality
- Calculating Financial Reserves (12 hrs to 15 min)

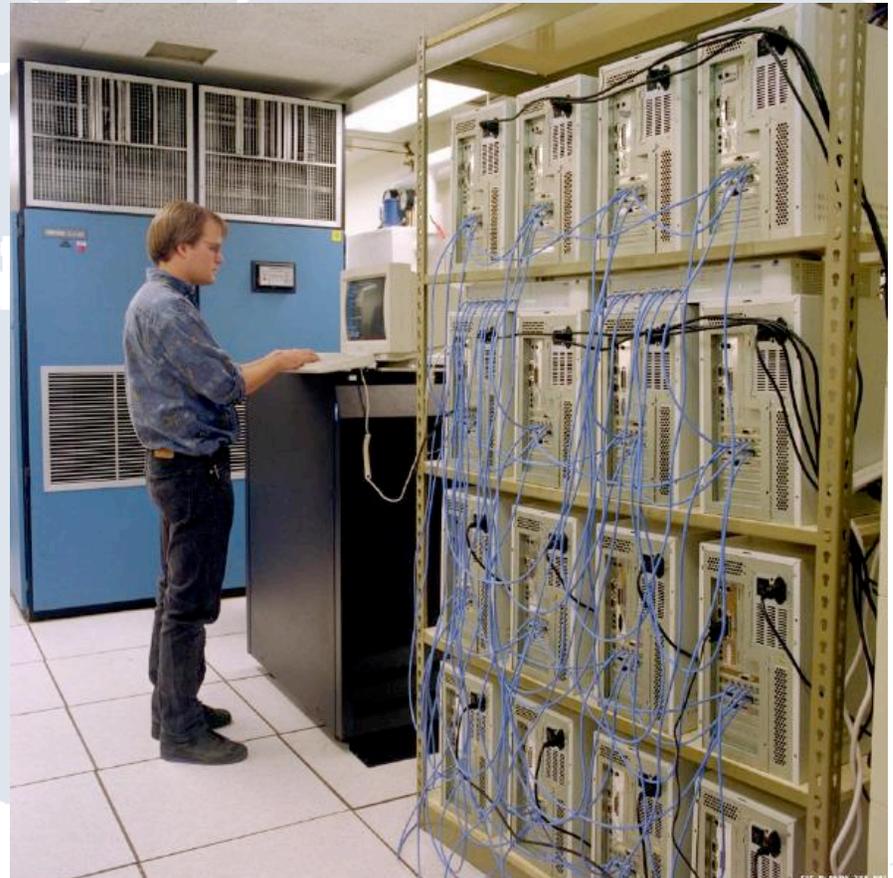
# Image Rendering

- Titanic
- Matrix
- Shrek



# Image Recognition: University of Sao Paulo

- Mammograms
- Homeland Security –  
Sailboats and  
conning towers



"The power of a IBM SP2 for 1/40<sup>th</sup> of the price"

- Pat Goda, Los Alamos Labs

# "And Quark Said, 'Hi'"

- Fermilab
- Six quarks, five found
- 60-250 Mbytes/sec filtered to 15 Mbytes/sec for tape storage

*"It is not that we can't do it, it is just that we can't afford it." - GP Yeh*

# Some Unusual Ones

- “Realtime adaptive control of earthquakes”
- “Simulation of airflow around aircraft....including time accurate simulation of release of bombs and missiles”
- Simulation of galaxies

*We currently run on large <vendor deleted> systems and are seeking to build a local capability. We cannot afford the <systems> we need with the budget we have. We hope to expand the cluster to > 128 nodes within a couple years.*

# Linux: A Kernel (But also much more...)

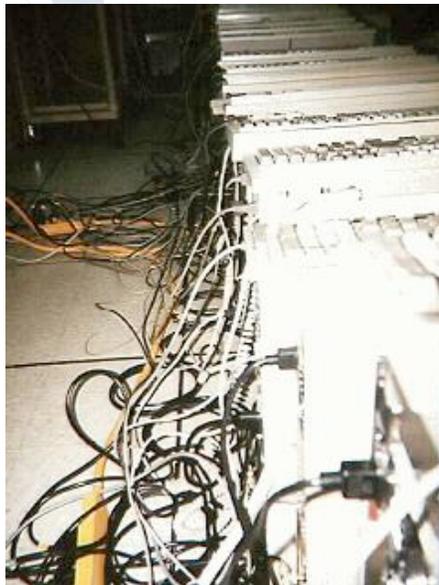
- Multi
  - User
  - Tasking
  - CPU
  - Architecture
- Demand-paged virtual memory (32 or 64 bit)
- Modular
- Standard (from embedded to supercomputers)

# SuperParallelism: The Beowulf Concept

- Inexpensive hardware
- Complex software replicated
- New Algorithms
- Hard to program
  - Bandwidth
  - Latency
  - Overhead

# Stone SouperComputer

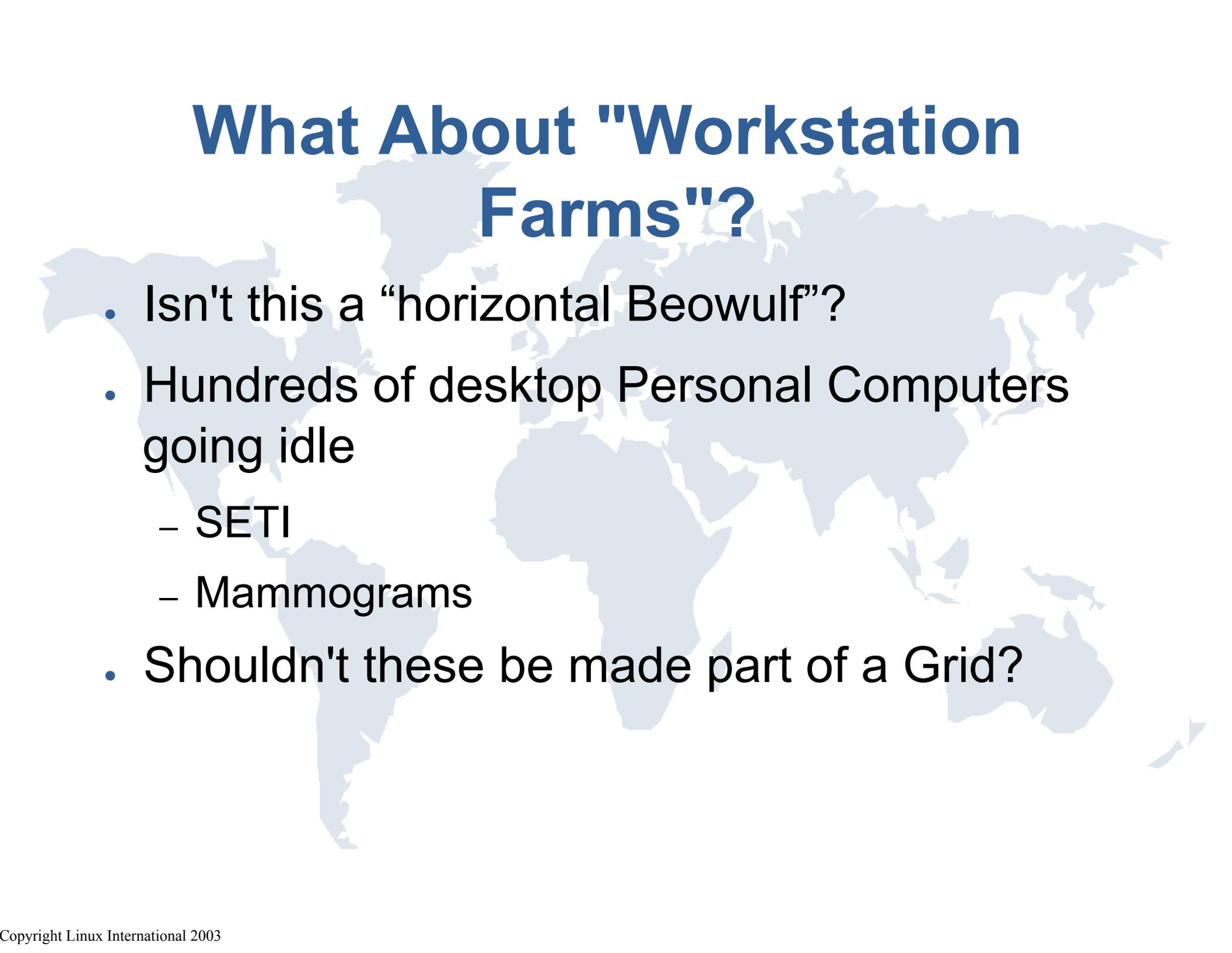
- 48 cast off machines
- Solved real work



# How Effective Are Beowulfs?



# What About "Workstation Farms"?

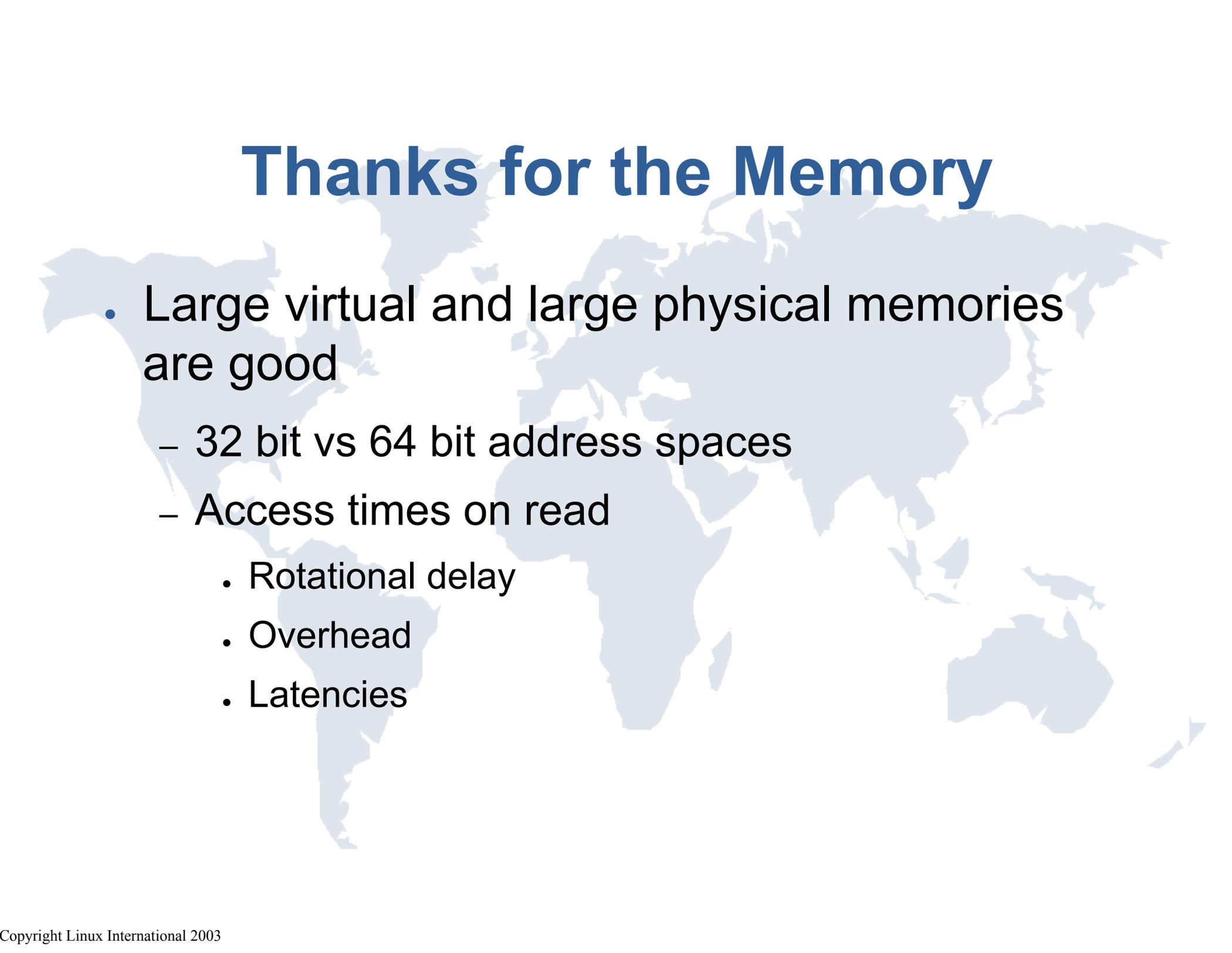


- Isn't this a "horizontal Beowulf"?
- Hundreds of desktop Personal Computers going idle
  - SETI
  - Mammograms
- Shouldn't these be made part of a Grid?

# NUMA Machines: Reducing Bottlenecks

- Easier to program, less chance for error
  - Lower latency
  - Higher throughput
  - Lower overhead
- Large number of CPUs than traditional SMP
- Possibility of much larger ratios of RAM/CPU than with traditional Beowulf

# Thanks for the Memory



- Large virtual and large physical memories are good
  - 32 bit vs 64 bit address spaces
  - Access times on read
    - Rotational delay
    - Overhead
    - Latencies

# 12 to 16 to 32 to 64

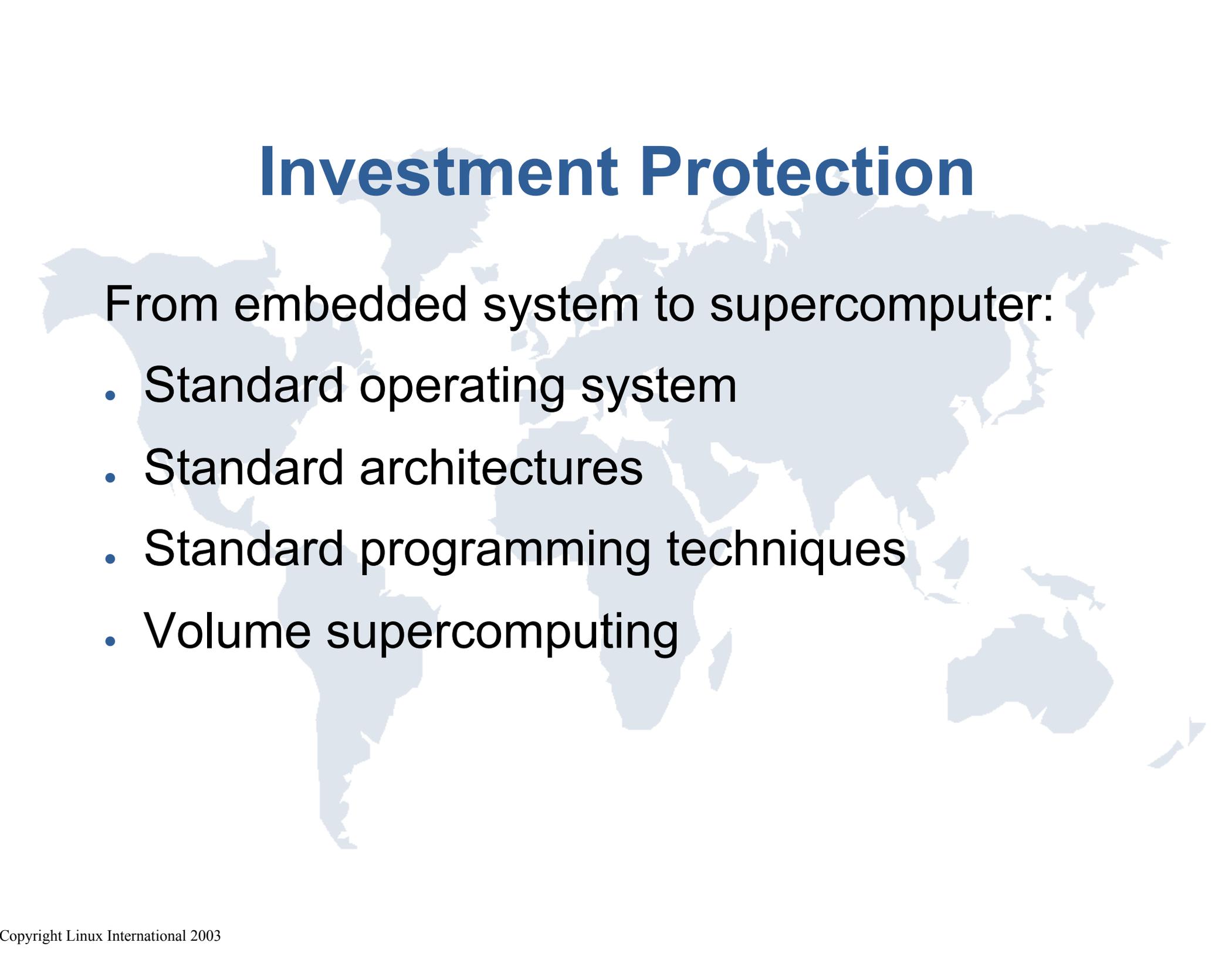
- Ten hours to three minutes
- A cockpit, or a whole plane
- A frame or a movie
- A six way join
- 128 bytes per square millimeter

# One Set of API/ABIs:

## From Embedded System to Supercomputer

- An emerging set of tools
- An emerging set of programmers
- An emerging set of applications

# Investment Protection



From embedded system to supercomputer:

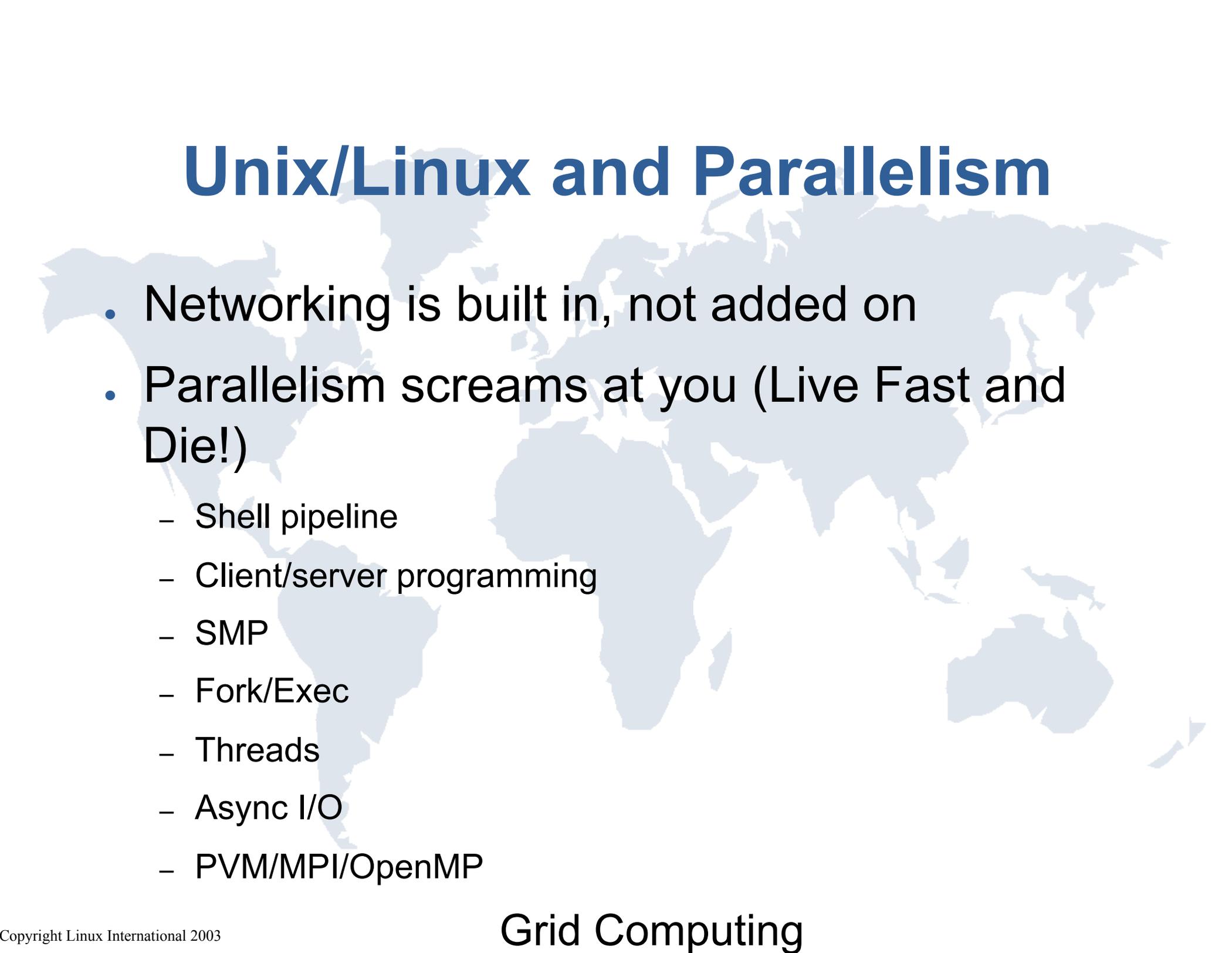
- Standard operating system
- Standard architectures
- Standard programming techniques
- Volume supercomputing



# Warning!

Highly Opinionated and Slides Follow!

# Unix/Linux and Parallelism

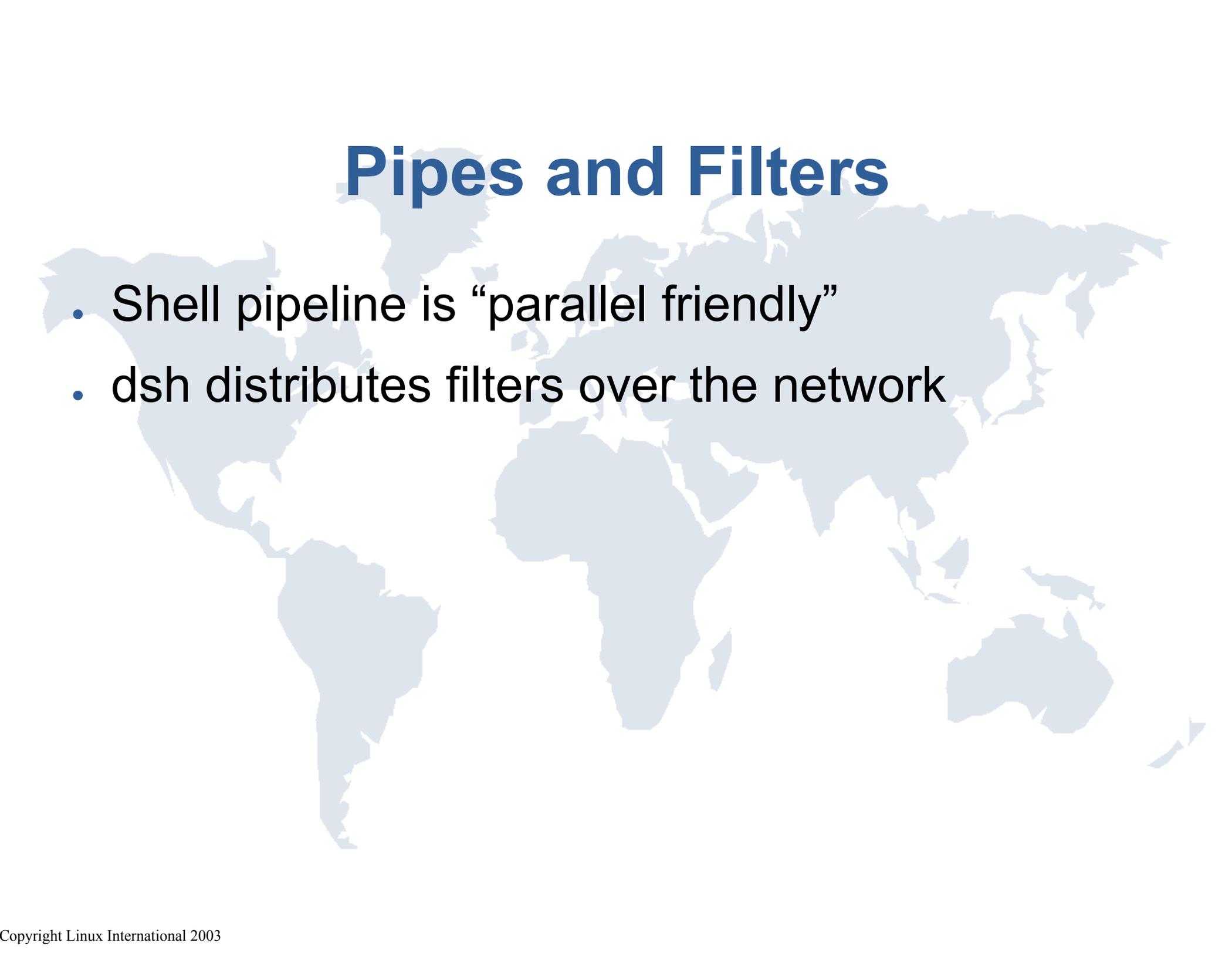


- Networking is built in, not added on
- Parallelism screams at you (Live Fast and Die!)
  - Shell pipeline
  - Client/server programming
  - SMP
  - Fork/Exec
  - Threads
  - Async I/O
  - PVM/MPI/OpenMP

# Even Single-CPU machines

- Parallelism in single program images speeds up wall clock time execution
  - Cuts down on I/O wait time
  - Keeps memory and cache "warmer"

# Pipes and Filters



- Shell pipeline is “parallel friendly”
- dsh distributes filters over the network

# fork(2)ing and exec(2)ing

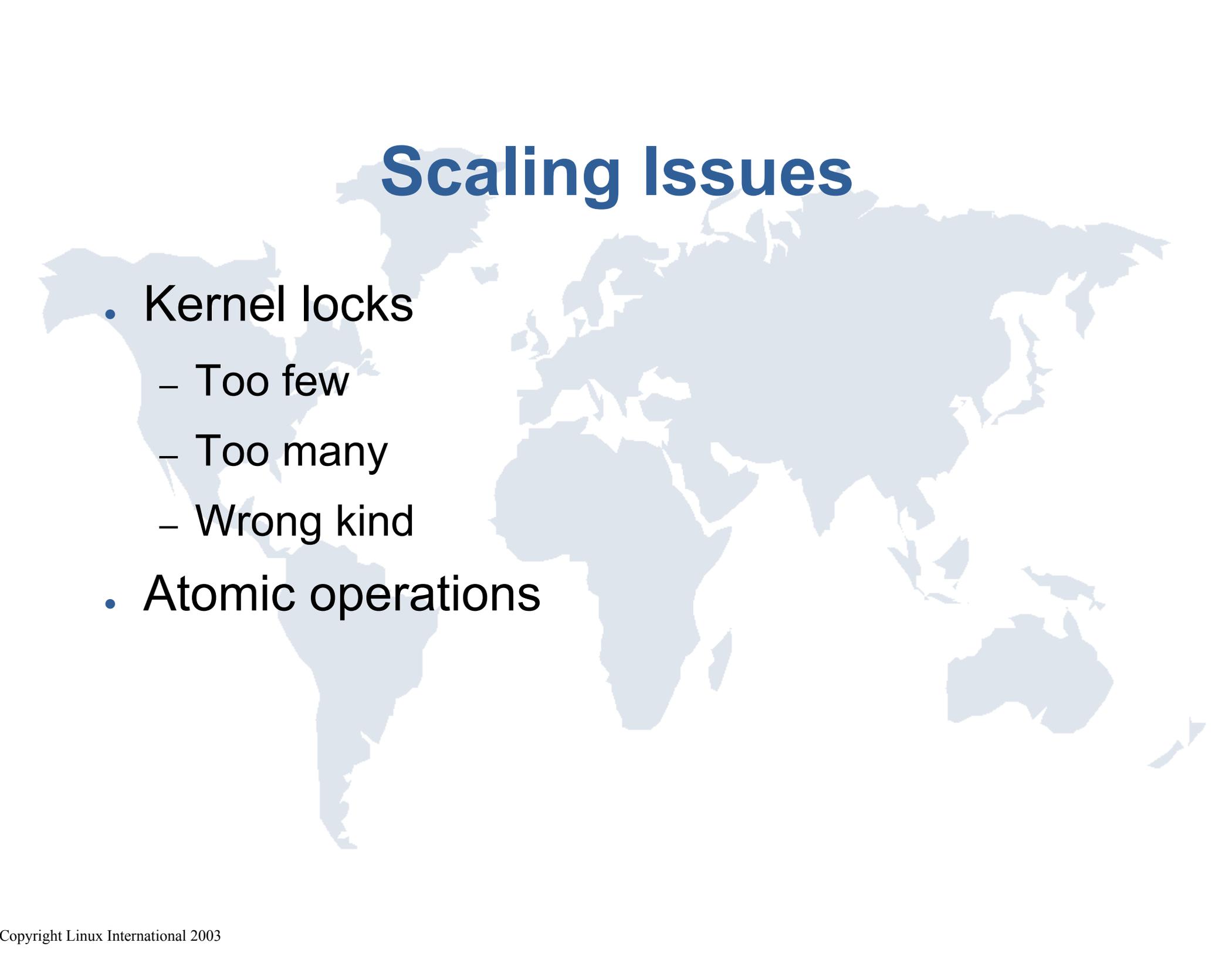
- “Light-weight” parallelism
- Work continues to make them lighter weight

# SMP Machines Rock!

(And Linux has been so since v2.0)

- There are only so many CPUs that you can fit in a memory bus
- Race conditions raise ugly head
  - Locks
    - Spin
    - Semaphore
    - Counting Semaphore
- Thread safe vs Multi-thread
- Process or thread-based scheduling

# Scaling Issues



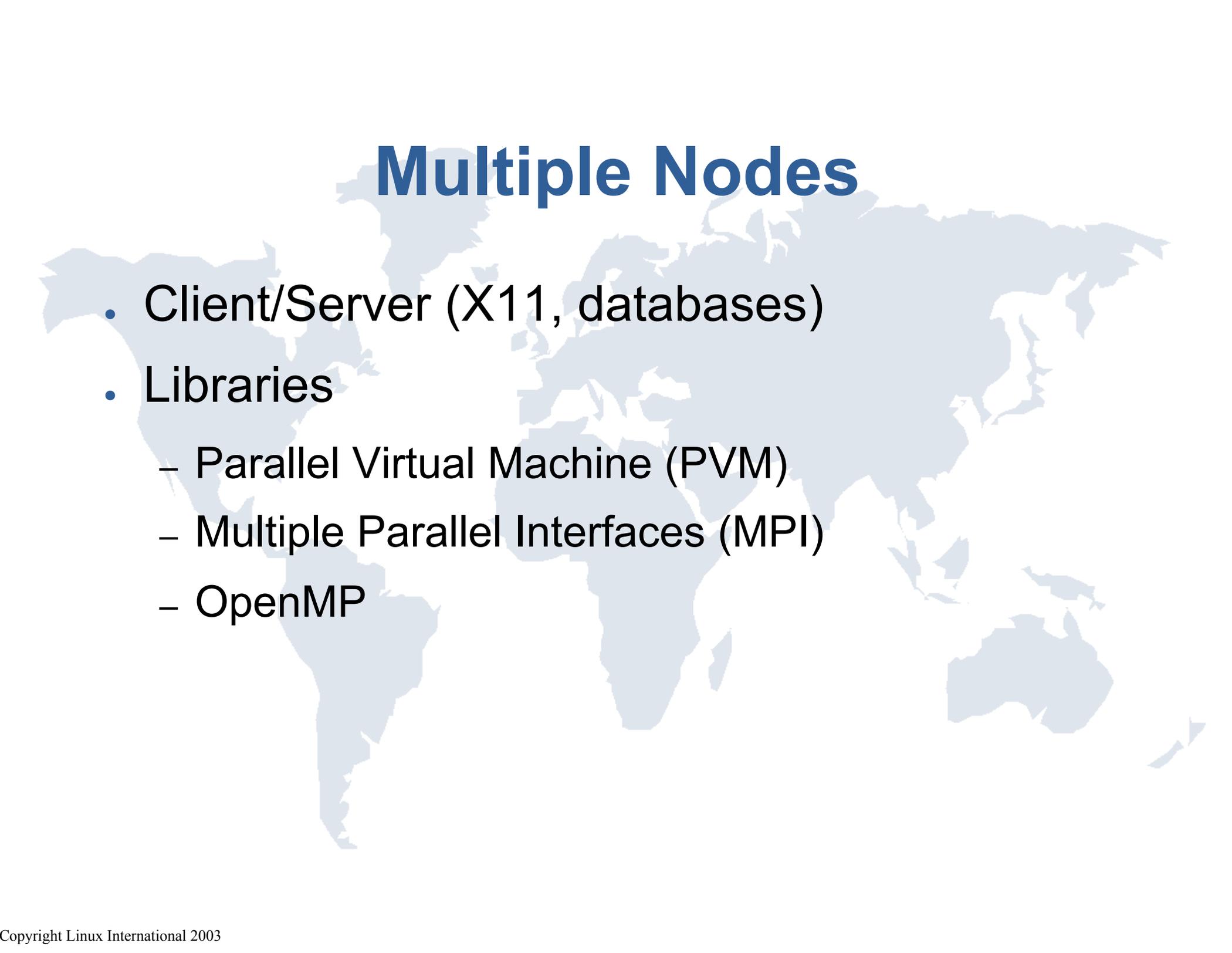
- Kernel locks
  - Too few
  - Too many
  - Wrong kind
- Atomic operations

# Async I/O



- Set up your I/O buffers
- Set up control blocks
- Tell OS “Go, go, go”
- Applications
  - Demons
  - getty

# Multiple Nodes



- Client/Server (X11, databases)
- Libraries
  - Parallel Virtual Machine (PVM)
  - Multiple Parallel Interfaces (MPI)
  - OpenMP

# Clusters

- High Reliability/Availability/Scalability (RAS)
  - High throughput
  - Checkpoint/Restart
- High Performance
  - One problem, high parallelism
- Single System Image or Multiple systems
- Failover capability (Various types)
- Process migration (Mosix)

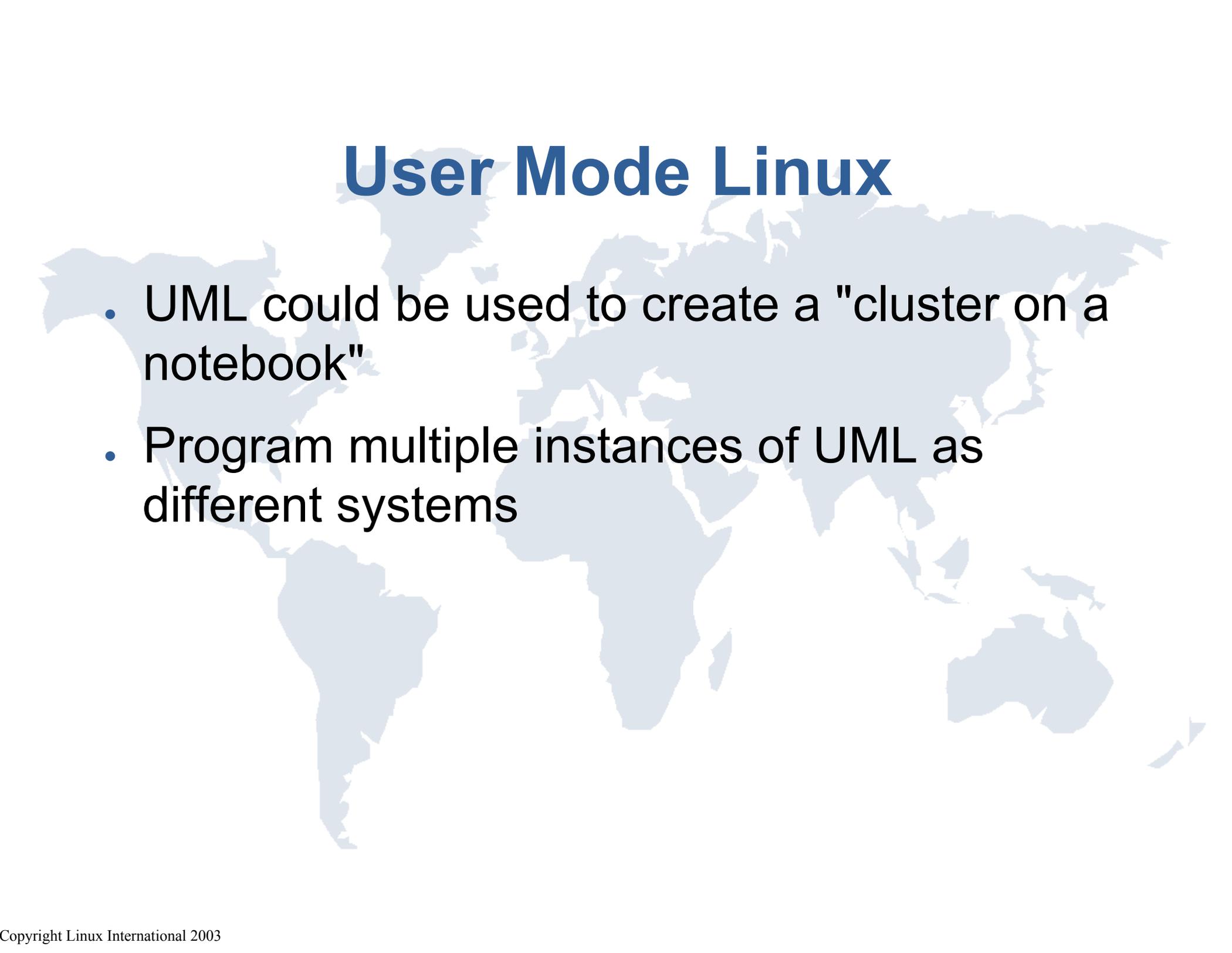
# Various Programs for Creating Beowulf Cluster Systems

- cfengine
- OSCAR
- Rocks ([rocks.npaci.edu](http://rocks.npaci.edu))
- SCMS (Smile Cluster Management System)
- OpenSSI (Alpha Code)
- System[Imager|Installer|Configurator]
- OpenMosix – Single-system image
- Ganglia - monitoring

# And Commercial Software

- Platform Computing's ([www.platform.com](http://www.platform.com)) Load Sharing Facility
- Scyld

# User Mode Linux



- UML could be used to create a "cluster on a notebook"
- Program multiple instances of UML as different systems

# The Grid: Are we ready?

- Do we have the skills to really take advantage of it?
- Should we be building any more Beowulfs?
  - Should we teach people to better utilize the systems we have today?
  - Should we be putting money into more specialized equipment?

*Should we be bringing supercomputing to the masses?*

# "Where Are My Games?"

- CS Department Head that did not understand "recursive and reentrant"
- CS degrees that believe "JAVA is enough"
- IM degrees that teach "Business through Microsoft"

# What About This Kid?

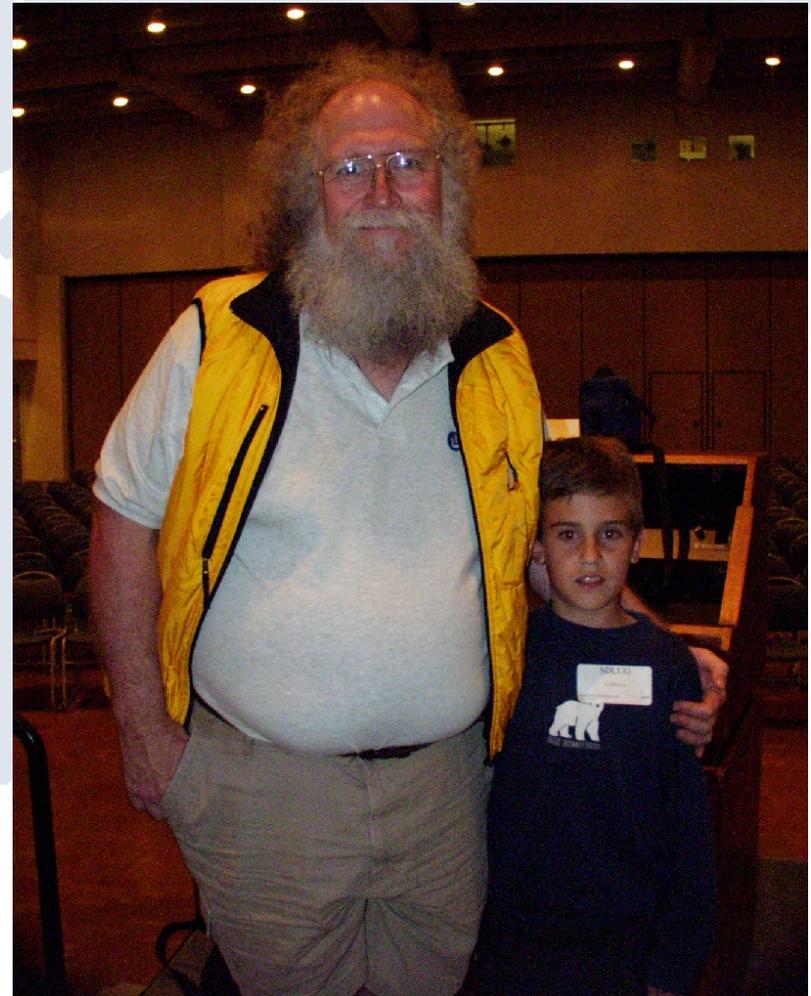
- Programming at 6
- Linux at 12
- PHAT distribution at 14
- CS at Ohio State
  - Musician
  - Wants to "learn hardware"

Do you think he is worried about the lack of games?.....  
....or is he writing his own?



# Or This Kid?

Proper way to  
decompose a  
problem for a  
Beowulf – age 10

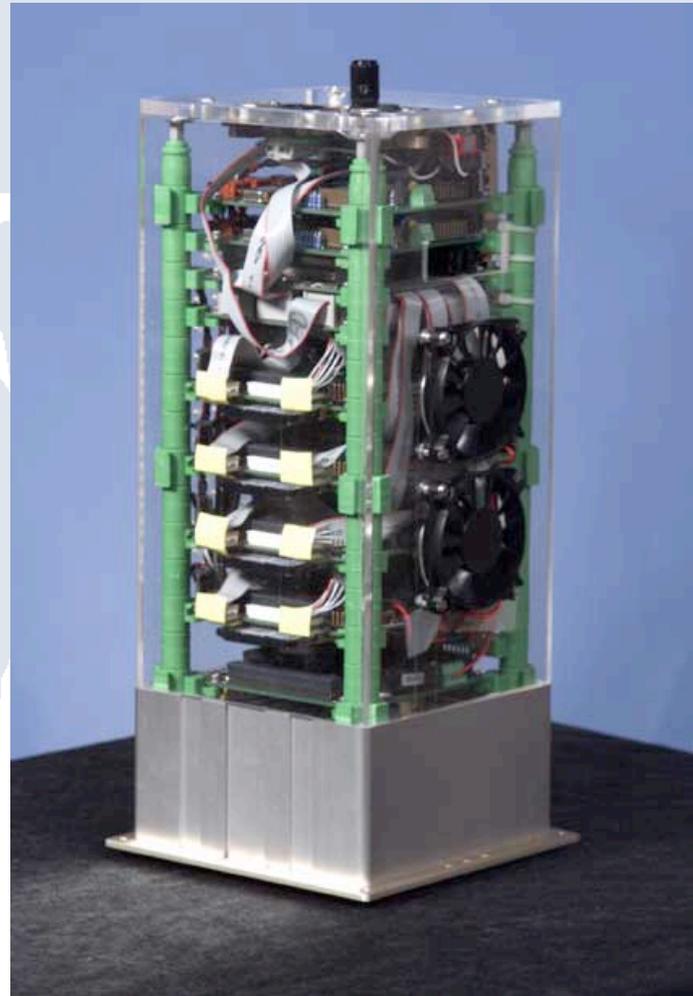


# Extreme Linux: A Revitalization of Fun

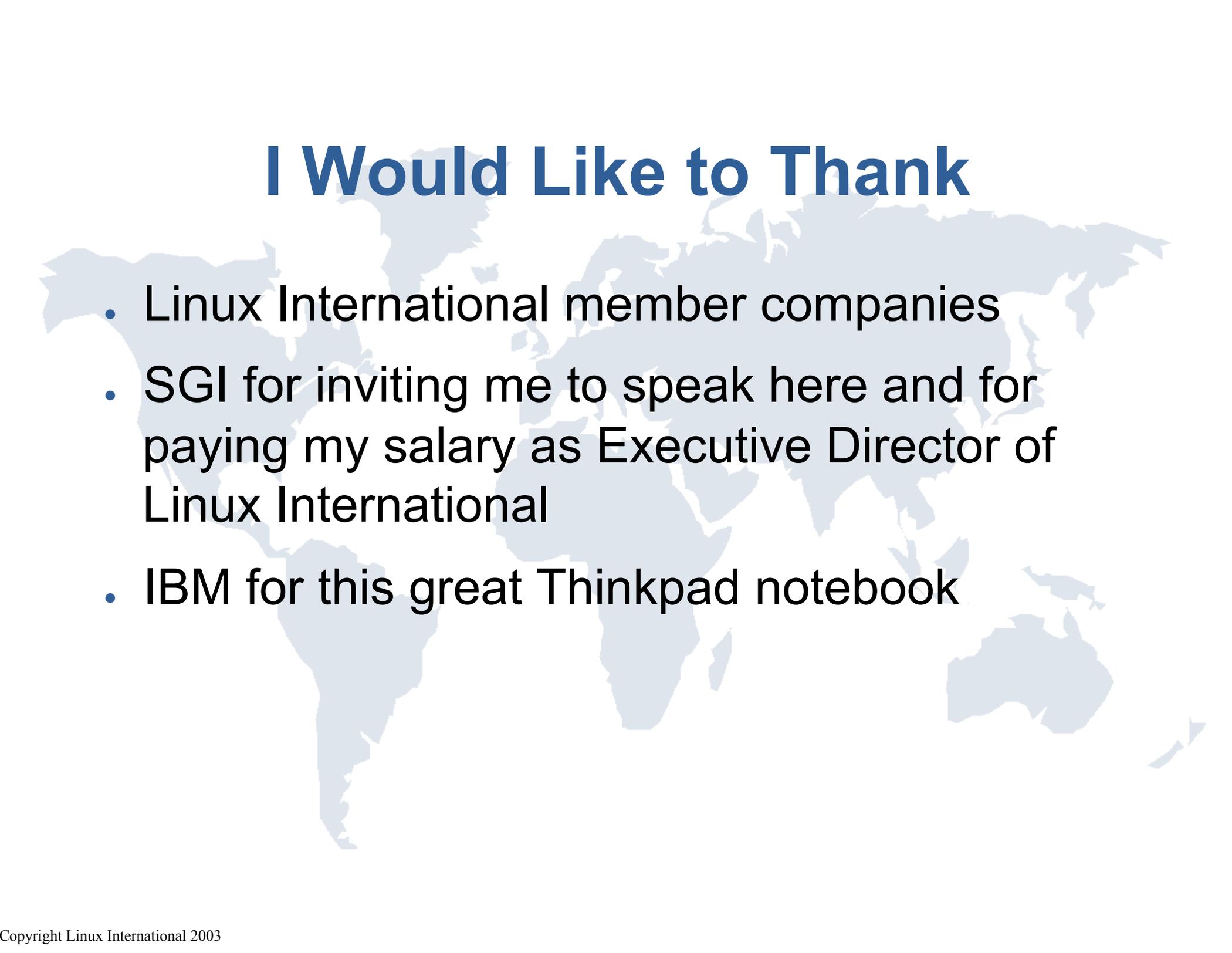
Should we have

- Extreme Linux boot camps? (Beer walkings?)
- Extreme Linux workshops?
- Extreme Linux mini-clusters?
- Extreme Linux "Open" teaching materials?
- Extreme Linux contests?

I think the answer is "yes" to all of these.

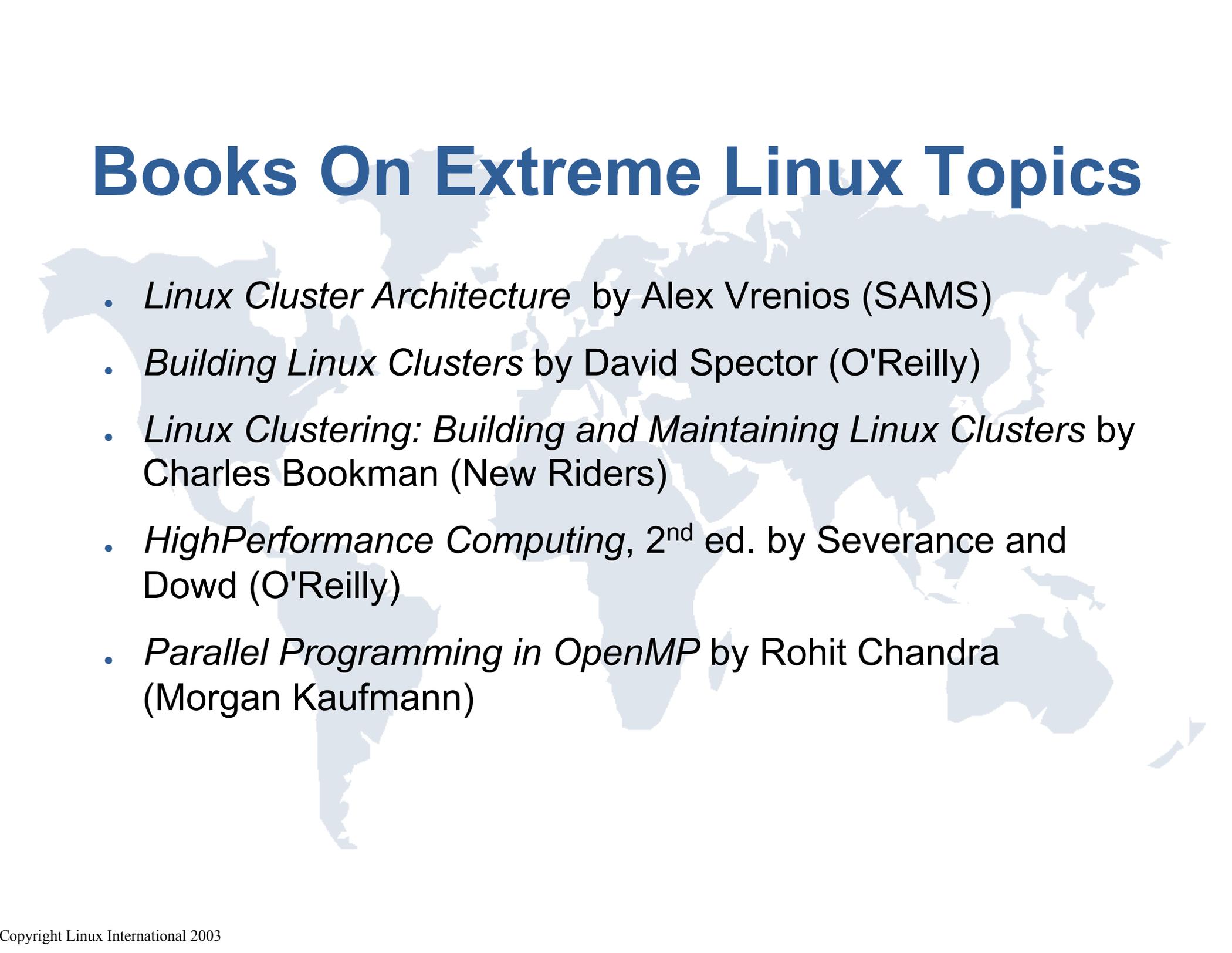


# I Would Like to Thank



- Linux International member companies
- SGI for inviting me to speak here and for paying my salary as Executive Director of Linux International
- IBM for this great Thinkpad notebook

# Books On Extreme Linux Topics



- *Linux Cluster Architecture* by Alex Vrenios (SAMS)
- *Building Linux Clusters* by David Spector (O'Reilly)
- *Linux Clustering: Building and Maintaining Linux Clusters* by Charles Bookman (New Riders)
- *HighPerformance Computing*, 2<sup>nd</sup> ed. by Severance and Dowd (O'Reilly)
- *Parallel Programming in OpenMP* by Rohit Chandra (Morgan Kaufmann)