SCALABLE WORKSTATIONS AND EXOKERNELS

Frans Kaashoek, Anant Agarwal, Dawson Engler,

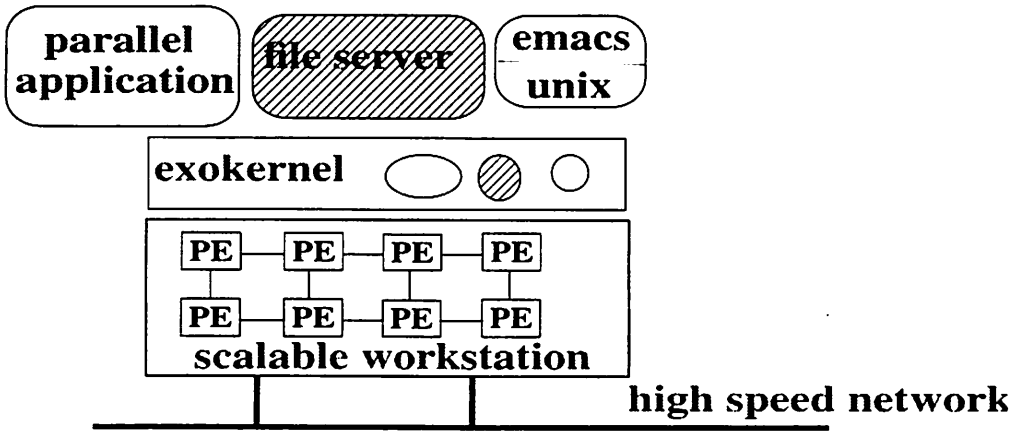
David Kranz, Ken Mackenzie

MIT Laboratory for Computer Science

High-performance computing

- **Old solution: supersupercomputers with an ad-hoc OS**
  1. **Not cost effective**
  2. **Bad failure properties**
  3. **Computing-center trauma**
- **NOW solution: uniprocessor workstations connected by fast network running UNIX plus goop**
  1. **Latency is too high; processor is too far from network**
  2. **Operating system buries hardware performance in layers of abstractions**
- **Better solution needed!**

Our solution: scalable workstations



- Applications access hardware resources directly
- Commodity process nodes
- Protected low-latency communication

Overview

- Hardware architecture
  - * NOW is an opportunistic short-term solution
  - * Scalable workstations are an economically-viable high-performance computing solution
  - * Fugu: a multimodel multiuser scalable workstation
- Software architecture
  - * Current OS are slow, inflexible, and big
  - * Exokernel: a secure programmable operating system
  - * Application libraries manage hardware resources

NOW is not the future

- Compared to scalable workstations, NOWs have inherently slow communication performance
- Communication performance is limited by:
  1. Standards committees
  2. Fault-tolerance requirements
  3. Security requirements
  4. Network interface distant from processor

NOWs communicate slowly

Architecture	Instance	Roundtrip latency (μsec)	Bandwidth (Mbyte/s)
NOW	ATM+AM	20	5.5
	RemoteAccess	45	4.4
	HPAM	29	12
	Myrinet	18	8 - 50
Scalable WS	Alewife	2	42
	Dash	3 - 4	5 - 32

- Be careful when comparing numbers directly!

What is a scalable workstation?

- Multiuser
  - * General purpose computing
- Multimodel
  - * Shared memory
  - * Message passing
- Scalable
  - * Low-overhead communication
  - * Low-latency network instead of bus
  - * Add process nodes like adding memory

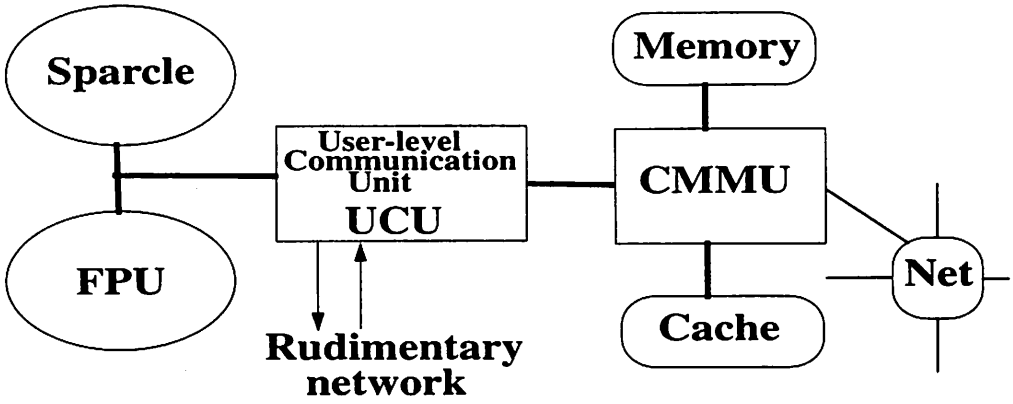
Why scalable workstations?

- High-performance communication
  - * No committees that define internal network
  - * Tight coupling of network and processor
- Multiple processors in a box is cost effective
  - * Amortize cost of screen, disk, packaging, power supply, etc.
- With high-performance local communication, locality-aware applications can tolerate slower remote communication

Fugu: a prototype scalable workstation

- Multiuser multimodel scalable workstation
- Novel ideas:
  1. Protected user-level message passing for both sending and receiving
  2. Scalable TLB management as a side-effect of DSM
  3. Rudimentary second network
  4. Exokernel OS that arbitrates resources, application libraries establish resource policies

Fugu node architecture

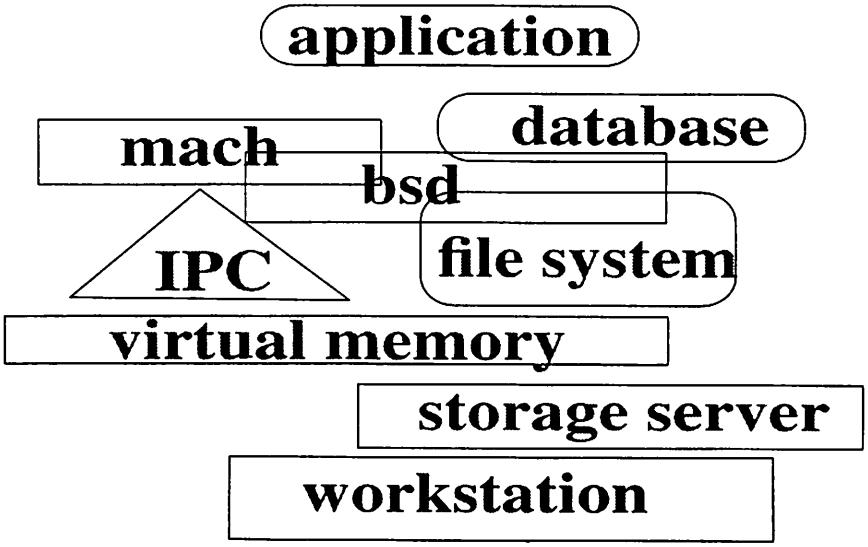


- Prototype leverages off Alewife; UCU is new
- TLB consistency protocol piggybacked on LimitLESS
- Rudimentary net: a "trap door" for OS to get control back

Exokernel: a new OS architecture

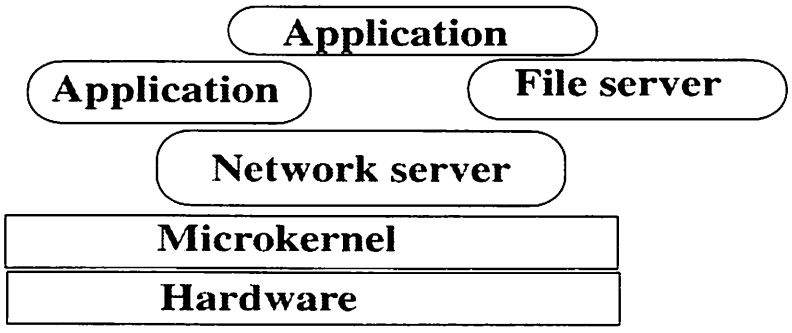
- Applications access physical hardware resources directly
- OS is a protected low-level machine
- Application libraries set resource management policies

What is wrong with today's OSs?



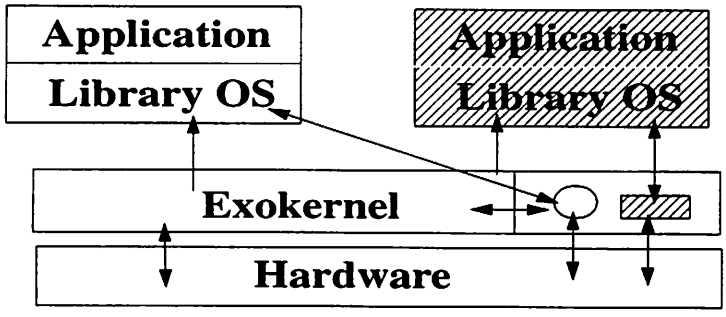
- They are big, complex, and inflexible!!!

Microkernel-based OSs



- More modular, more flexible, but:
  1. Supervisor mode defines "high-level" interface
  2. Address space crossings are expensive
  3. Servers define fixed policies

Exokernel structure:



- Exokernel is a low-level protected kernel that makes hardware resources directly accessible to applications
- Applications set policies through OS libraries
- Applications can safely extend exokernel
- Can support traditional clients and servers too

Advantages

- Can build ambitious systems without fighting OS
  - * E.g., database package
- Can specialize abstractions to given applications
  - * E.g, linear page tables for small address space
- Experiment without affecting rest of system
  - * Implementing library easier than implementing OS
- Good performance
  - * Applications access resources directly and specialize
  - * Most operations occur in same address space

Aegis: A prototype exokernel

- A capability-based protected machine with few system calls
- Application code running in Aegis is "sandboxed"
- Library OS defines its own page-table organization; e.g., hierarchical or flat
- Library OS implements several IPC primitives; e.g., some trust server to save and restore registers
- Currently runs on DECstations

Conclusions

- **Cost-effective high-performance computing systems require rethinking both hardware and software architectures**
- **Clusters of workstations running UNIX are not going to cut it**
- **Alternative solution:**
  1. **Hardware: scalable workstations**
  2. **Software: exokernel operating system**

  *** Plug in commodity process nodes**

  *** Protected low-latency communication**

  *** Hardware performance is delivered to application**