SHANNON JOYNER

MODERN SYSTEMS: EXTENSIBLE KERNELS AND CONTAINERS
MOTIVATION

- Applications must conform to operating system interface
  - One operating system implementation is not ideal for everyone
- Cannot optimize for a given application
  - OSs did not work well for database management systems

---

1 Operating System Support for Database Management. Michael Stonebraker. Communications of the ACM.
MOTIVATION

- Operating systems are complex
  - Accessing resources require many system calls
- This complexity in UNIX lead to Mach
  - Give application more control over system

Exokernel: An Operating System Architecture for Application-Level Resource Management

Dawson R. Engler, M. Frans Kaashoek, and James O’Toole Jr.

M.I.T. Laboratory for Computer Science

EXOKERNEL GOALS

- Make small, fast kernel implementation by implementing simple primitives
- Secure, fast way to access hardware resources
- System abstractions can be implemented efficiently by application
- Applications can have fast, specialized implementations
EXOKERNEL

- Make small, fast kernel implementation by implementing simple primitives
  - Exokernel

- Secure, fast way to access hardware resources
  - TLB + Secure Bindings

- System abstractions can be implemented efficiently by application
  - LibOS + Exokernel

- Applications can have fast, specialized implementations
  - LibOS
EXOKERNEL

- LibOS
- Secure Bindings
- TLB
- Exokernel

Exokernel, Figure 1
LIBRARY OPERATING SYSTEMS

- Implements OS besides interaction with hardware
- Customize based on application needs
- Manages resource policies
- Example LibOS customizations
  - Virtual Memory
  - Scheduling
  - Networking

Exokernel, Figure 1

SECURE BINDINGS

- Used by LibOSes to access resources
- Provides connection to resource
- Decouples authorization from use of resource
- Hardware can implement protection checks quickly
SOFTWARE TRANSLATION LOOK-ASIDE BUFFER (TLB)

- Caches virtual memory to physical memory translations
- Cache secure bindings in TLB to reduce number of binding connections
- Improves performance
- Only handles resource sharing
- Rest left to application
- Does not handle resource polices
EXOKERNEL

- Make small, fast kernel implementation by implementing simple primitives
  - Exokernel

- Secure, fast way to access hardware resources
  - TLB + Secure Bindings

- System abstractions can be implemented efficiently by application
  - LibOS + Exokernel

- Applications can have fast, specialized implementations
  - LibOS
IMPLEMENTATION

- Aegis: Exokernel [Page 7]
- Processor Time Slices [Page 7]
- Exceptions [Page 8]
- Protected Control Transfer (PCT) [Page 9]
- Dynamic Packet Filter (DPF) [Page 10]

Exokernel, Dispatch Exceptions, Table 5

<table>
<thead>
<tr>
<th>Machine</th>
<th>OS</th>
<th>unalign</th>
<th>overflow</th>
<th>coproc</th>
<th>prot</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100</td>
<td>Ultrix</td>
<td>n/a</td>
<td>208.0</td>
<td>n/a</td>
<td>238.0</td>
</tr>
<tr>
<td>DEC2100</td>
<td>Aegis</td>
<td>2.8</td>
<td>2.8</td>
<td>2.8</td>
<td>3.0</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Ultrix</td>
<td>n/a</td>
<td>151.0</td>
<td>n/a</td>
<td>177.0</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Aegis</td>
<td>2.1</td>
<td>2.1</td>
<td>2.1</td>
<td>2.3</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Ultrix</td>
<td>n/a</td>
<td>130.0</td>
<td>n/a</td>
<td>154.0</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Aegis</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>
IMPLEMENTATION

- ExOS: LibOS [Page 10]
- Interprocess communication [Page 10]
- Application Specific Safe Handlers [Page 11]
- Virtual Memory [Page 11]
QUESTIONS

- Who handles resource policies?
- Is there a problem with how exokernels handle resource policies?
- Why do we need secure bindings?
Who handles resource policies?

Is there a problem with how exokernels handle resource policies?

Why do we need secure bindings?
Who handles resource policies?

- Application handles resource policies.
QUESTIONS

- Who handles resource policies?
- Is there a problem with how exokernels handle resource policies?
- Why do we need secure bindings?
QUESTIONS

- Is there a problem with how exokernels handle resource policies?
  - Poor isolation. Applications can have conflicting policies.
QUESTIONS

‣ Who handles resource policies?

‣ Is there a problem with how exokernels handle resource policies?

‣ Why do we need secure bindings?
Why do we need secure bindings?

- Protection / authorization of resources. Faster to do in kernel and kernel does not need to understand resources.
EXOKERNEL TAKEAWAYS

Strengths

› Minimal kernel and customizable operating system
› Fast

Weakness

› Poor isolation
   › Each application implements own LibOS
   › No way to prevent systems from conflicting
› Hardware compatibility
   › Need to change LibOS depending on hardware interface
Hypervisor

- Hardware abstraction

Previous virtual machines ran on top of hypervisors

- No isolation

---

1 Container-based Operating System Virtualization: A Scalable, High performance Alternative to Hypervisors
CONTAINERS

- Grouping of processes

- Strength
  - Provide isolation between groups

- Weakness
  - Containers cannot customize operating systems

Diagram:

- Container
  - MySQL
- Container
  - Web Server
- Container
  - MySQL
  - Web Server

OS

Hypervisor
Unikernel: Library Operating Systems for the Cloud

Anil Madhavapeddy, Richard Mortier, Charalampos Rotsos, David Scott, Ralraj Singh, Thomas Gazagnaire, Steven Smith, Steven Hand, and Jon Crowcroft

University of Cambridge, University of Nottingham, Citrix Systems Ltd, OCamlPro SAS

In Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems pg. 461-472.
UNIKERNEL = EXOKERNEL + CONTAINERS

- Run one application per virtual machine
- One process per application
- Everything compiled into a VM image
- Do not compile unused code

Unikernel, Figure 1
UNIKERNEL

- Run directly on top of standard hypervisor
- Can run multiple unikernels on the same hypervisor
MIRAGE

- Produces unikernels
- Compiles OCaml code to Xen VM image
- 4 main components
  - Text + Data segment
  - Foreign Grants
  - Minor Heap
  - Major Heap
TEXT AND DATA

- OCaml Runtime
- PVBoot
  - Initializes VM
**HEAP**

- Minor Heap
  - Short lived values in VM
  - Fast

- Major Heap
  - Long lived values
FOREIGN GRANTS

- Used for VM communication
- Write data to a grant table
- Exchange table between VM address spaces
Mirage unikernel improvements result in better performance than having multiple cores.
EXOKERNEL VERSUS UNIKERNEL

- **Exokernel**
  - Fast and customizable
  - All applications on same system
  - Poor isolation

- **Unikernel**
  - Fast and customizable
  - Single application per system
  - Better isolation
ACKNOWLEDGEMENTS

‣ Thanks to Hakim for helping me prepare for this presentation!


‣ Container-based Operating System Virtualization: A Scalable, High performance Alternative to Hypervisors

PROTECTED CONTROL TRANSFER

- Implementation of interprocess communication
  - Put the messages in the receiver process’s context
- Asynchronous: Rest of sender process's time slice goes to receiver
- Synchronous: All future time slices go to receiver process
- 7x faster than best reported implementation

<table>
<thead>
<tr>
<th>OS</th>
<th>Machine</th>
<th>MHz</th>
<th>Transfer cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aegis</td>
<td>DEC2100</td>
<td>12.5MHz</td>
<td>2.9</td>
</tr>
<tr>
<td>Aegis</td>
<td>DEC3100</td>
<td>16.67MHz</td>
<td>2.2</td>
</tr>
<tr>
<td>Aegis</td>
<td>DEC5000</td>
<td>25MHz</td>
<td>1.4</td>
</tr>
<tr>
<td>L3</td>
<td>486</td>
<td>50MHz</td>
<td>9.3 (normalized)</td>
</tr>
<tr>
<td>Machine</td>
<td>OS</td>
<td>Procedure call</td>
<td>Syscall (getpid)</td>
</tr>
<tr>
<td>-----------</td>
<td>-----</td>
<td>----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>DEC2100</td>
<td>Ultrix</td>
<td>0.57</td>
<td>32.2</td>
</tr>
<tr>
<td>DEC2100</td>
<td>Aegis</td>
<td>0.56</td>
<td>3.2 / 4.7</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Ultrix</td>
<td>0.42</td>
<td>33.7</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Aegis</td>
<td>0.42</td>
<td>2.9 / 3.5</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Ultrix</td>
<td>0.28</td>
<td>21.3</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Aegis</td>
<td>0.28</td>
<td>1.6 / 2.3</td>
</tr>
</tbody>
</table>

Exokernel, Table 5
<table>
<thead>
<tr>
<th>Machine</th>
<th>OS</th>
<th>pipe</th>
<th>pipe’</th>
<th>shm</th>
<th>lrpc</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100</td>
<td>Ultrix</td>
<td>326.0</td>
<td>n/a</td>
<td>187.0</td>
<td>n/a</td>
</tr>
<tr>
<td>DEC2100</td>
<td>ExOS</td>
<td>30.9</td>
<td>24.8</td>
<td>12.4</td>
<td>13.9</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Ultrix</td>
<td>243.0</td>
<td>n/a</td>
<td>139.0</td>
<td>n/a</td>
</tr>
<tr>
<td>DEC3100</td>
<td>ExOS</td>
<td>22.6</td>
<td>18.6</td>
<td>9.3</td>
<td>10.4</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Ultrix</td>
<td>199.0</td>
<td>n/a</td>
<td>118.0</td>
<td>n/a</td>
</tr>
<tr>
<td>DEC5000</td>
<td>ExOS</td>
<td>14.2</td>
<td>10.7</td>
<td>5.7</td>
<td>6.3</td>
</tr>
</tbody>
</table>
# Time to Perform VM Operations

<table>
<thead>
<tr>
<th>Machine</th>
<th>OS</th>
<th>dirty</th>
<th>prot1</th>
<th>prot100</th>
<th>unprot100</th>
<th>trap</th>
<th>appel1</th>
<th>appel2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100</td>
<td>Ultrix</td>
<td>n/a</td>
<td>51.6</td>
<td>175.0</td>
<td>175.0</td>
<td>240.0</td>
<td>383.0</td>
<td>335.0</td>
</tr>
<tr>
<td>DEC2100</td>
<td>ExOS</td>
<td>17.5</td>
<td>32.5</td>
<td>213.0</td>
<td>275.0</td>
<td>13.9</td>
<td>74.4</td>
<td>45.9</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Ultrix</td>
<td>n/a</td>
<td>39.0</td>
<td>133.0</td>
<td>133.0</td>
<td>185.0</td>
<td>302.0</td>
<td>267.0</td>
</tr>
<tr>
<td>DEC3100</td>
<td>ExOS</td>
<td>13.1</td>
<td>24.4</td>
<td>156.0</td>
<td>206.0</td>
<td>10.1</td>
<td>55.0</td>
<td>34.0</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Ultrix</td>
<td>n/a</td>
<td>32.0</td>
<td>102.0</td>
<td>102.0</td>
<td>161.0</td>
<td>262.0</td>
<td>232.0</td>
</tr>
<tr>
<td>DEC5000</td>
<td>ExOS</td>
<td>9.8</td>
<td>16.9</td>
<td>109.0</td>
<td>143.0</td>
<td>4.8</td>
<td>34.0</td>
<td>22.0</td>
</tr>
</tbody>
</table>
## Time to Perform VM Operations (Two Different Page-Tables)

<table>
<thead>
<tr>
<th>Machine</th>
<th>Method</th>
<th>dirty</th>
<th>prot1</th>
<th>prot100</th>
<th>unprot100</th>
<th>trap</th>
<th>appel1</th>
<th>appel2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100</td>
<td>Original page-table</td>
<td>17.5</td>
<td>32.5</td>
<td>213.</td>
<td>275.</td>
<td>13.9</td>
<td>74.4</td>
<td>45.9</td>
</tr>
<tr>
<td>DEC2100</td>
<td>Inverted page-table</td>
<td>8.0</td>
<td>23.1</td>
<td>253.</td>
<td>325.</td>
<td>13.9</td>
<td>54.4</td>
<td>38.8</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Original page-table</td>
<td>13.1</td>
<td>24.4</td>
<td>156.</td>
<td>206.</td>
<td>10.1</td>
<td>55.0</td>
<td>34.0</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Inverted page-table</td>
<td>5.9</td>
<td>17.7</td>
<td>189.</td>
<td>243.</td>
<td>10.1</td>
<td>40.4</td>
<td>28.9</td>
</tr>
</tbody>
</table>

Table 3, Exokernel
<table>
<thead>
<tr>
<th>Machine</th>
<th>OS</th>
<th>Roundtrip latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC5000/125</td>
<td>ExOS/ASH</td>
<td>259</td>
</tr>
<tr>
<td>DEC5000/125</td>
<td>ExOS</td>
<td>320</td>
</tr>
<tr>
<td>DEC5000/125</td>
<td>Ultrix</td>
<td>3400</td>
</tr>
<tr>
<td>DEC5000/200</td>
<td>Ultrix/FRPC</td>
<td>340</td>
</tr>
</tbody>
</table>

Exokernel, Table 11
EXOKERNEL: SEPARATE PROTECTION FROM MANAGEMENT

- Tracking ownership of resources
- Protect bindings and resource usage
  - Ex. Accessing memory not accessible to application
- Revoking resource privileges

Exokernel, Figure 1
EXOS APPLICATION-LEVEL STRIDE SCHEDULER

- Each process given a fixed proportion of resources
- Implemented counter program

Exokernel, Figure 3
TIME TO PERFORM LRPC EXTENSIONS

<table>
<thead>
<tr>
<th>Machine</th>
<th>lrpc</th>
<th>tlrpc</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100</td>
<td>13.9</td>
<td>8.6</td>
</tr>
<tr>
<td>DEC3100</td>
<td>10.4</td>
<td>6.4</td>
</tr>
<tr>
<td>DEC5000</td>
<td>6.3</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Table 12
### Time to Classify TCP/IP Headers

<table>
<thead>
<tr>
<th>Filter</th>
<th>Classification Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPF</td>
<td>35.0</td>
</tr>
<tr>
<td>PATHFINDER</td>
<td>19.0</td>
</tr>
<tr>
<td>DPF</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Exokernel, Table 7
## Time to Perform 150x150 Matrix Multiplication

<table>
<thead>
<tr>
<th>Machine</th>
<th>OS</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100</td>
<td>Ultrix</td>
<td>7.1</td>
</tr>
<tr>
<td>DEC2100</td>
<td>ExOS</td>
<td>7.0</td>
</tr>
<tr>
<td>DEC3100</td>
<td>Ultrix</td>
<td>5.2</td>
</tr>
<tr>
<td>DEC3100</td>
<td>ExOS</td>
<td>5.2</td>
</tr>
<tr>
<td>DEC5000</td>
<td>Ultrix</td>
<td>3.8</td>
</tr>
<tr>
<td>DEC5000</td>
<td>ExOS</td>
<td>3.7</td>
</tr>
</tbody>
</table>
## Sample of AEGIS’s Call Interface

<table>
<thead>
<tr>
<th>System call</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yield</td>
<td>Yield processor to named process</td>
</tr>
<tr>
<td>Scall</td>
<td>Synchronous protected control transfer</td>
</tr>
<tr>
<td>Acall</td>
<td>Asynchronous protected control transfer</td>
</tr>
<tr>
<td>Alloc</td>
<td>Allocation of resources (e.g., physical page)</td>
</tr>
<tr>
<td>Dealloc</td>
<td>Deallocation of resources</td>
</tr>
</tbody>
</table>

Table 2
### Machine Configuration for Exokernel Experiments

<table>
<thead>
<tr>
<th>Machine</th>
<th>Processor</th>
<th>SPEC rating</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC2100 (12.5 MHz)</td>
<td>R2000</td>
<td>8.7 SPECint89</td>
<td>~ 11</td>
</tr>
<tr>
<td>DEC3100 (16.67 MHz)</td>
<td>R3000</td>
<td>11.8 SPECint89</td>
<td>~ 15</td>
</tr>
<tr>
<td>DEC5000/125 (25 MHz)</td>
<td>R3000</td>
<td>16.1 SPECint92</td>
<td>~ 25</td>
</tr>
</tbody>
</table>

Table 1
## Sample of Aegis’s Primitive Operations

<table>
<thead>
<tr>
<th>Primitive operations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLBwr</td>
<td>Insert mapping into TLB</td>
</tr>
<tr>
<td>FPUmod</td>
<td>Enable/disable FPU</td>
</tr>
<tr>
<td>CIDswitch</td>
<td>Install context identifier</td>
</tr>
<tr>
<td>TLBvadelete</td>
<td>Delete virtual address from TLB</td>
</tr>
</tbody>
</table>

Table 3
EXOKERNEL

- Provides interface to interact with hardware
- Physical resources managed at application-level
- Most of the system implemented using a Library Operating System
- Separate resource protection from resource management
SECURE BINDINGS

- Used by LibOSes to access resources
- Decouples authorization from use of resource
- Performs authorization at bind time
- Hardware can implement protection checks quickly
LIBRARY OPERATING SYSTEM (LIBOS)

- Use hardware interface
- Implement rest of the operating system
- Implementation can be specialized to meet needs of applications