Unikernels Motivations, Benefits and Issues: A Multivocal Literature Review

NABIL EL IOINI, University of Nottingham Malaysia, Malaysia
AYOUB EL MAJJODI, University of Bergen, Norway
DAVID HASTBACKA, Tampere University, Finland
TOMAS CERNY
Davide Taibi (University of Oulu)

Oct. 17, 2023
Ludwigsburg, Germany
Context

• Edge Computing Require Lightweight containers

• Unikernels can be a potential solution
Unikernels

Highly-specialized single-address space, immutable and lightweight images.

Linking an application only with its necessary libraries at compile-time

All the services, from device drivers to schedulers moved to the network stack

Two categories

• Language-based: Tied to single technology

• POSIX-like: single address space and a single privilege level
VM, Containers and Unikernels

Virtual Machines

Containers

Unikernels
Goal and RQs

• What are the motivations for the adoption of Unikernels?

• What benefits are achieved by using Unikernels?

• What are the major issues of Unikernels
Multivocal Literature Review

Method

Analysis of Grey Literature and Peer-Reviewed Literature

Search String:
unikernel* AND (motivations OR benefits OR problem* OR issue* OR “operating system”).

- **PR**: ACM digital Library, IEEEXplore Digital Library, Scopus, Springer link
- **GL**: Google Search, Twitter, Search, Reddit Search, Medium Search, LinkedIn Search, Quora, Hacker News Algolia Search
Multivocal Literature Review

Method

- Snowballing, Quality Assessment of GL, Inter-rater reliability, Open/Selective Coding...

Results:

- 590 initial sources
  - Inclusion/Exclusion:
    - 528 excluded
    - 62 included
      - 40 (64.51 %) peer-reviewed-conference papers
      - 22 (35.49 %) grey literature
## Unikernels

### Results

<table>
<thead>
<tr>
<th>Unikernel Framework</th>
<th>Targets</th>
<th>Programming Languages</th>
<th>Project Status</th>
<th># Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hermitux</td>
<td>Xen, KVM</td>
<td>C, C++, Fortan, Python</td>
<td>4 active contributors</td>
<td>2</td>
</tr>
<tr>
<td>Lupine-linux</td>
<td>KVM</td>
<td>language independent</td>
<td>4 active contributors</td>
<td>1</td>
</tr>
<tr>
<td>Rumprn</td>
<td>Xen, KVM</td>
<td>C, C++, Java, Go, JavaScript, Node, Python, Ruby</td>
<td>last commit was on May 11, 2020, 25 contributors</td>
<td>16</td>
</tr>
<tr>
<td>IncludeOS</td>
<td>KVM, ESXi, OpenStack</td>
<td>C++</td>
<td>last commit was on May 11, 2020, 60 contributors</td>
<td>10</td>
</tr>
<tr>
<td>MirageOS</td>
<td>KVM, Xen</td>
<td>OCaml</td>
<td>last commit was on December, 2020, 52 contributor</td>
<td>16</td>
</tr>
<tr>
<td>OSv</td>
<td>VirtualBox, ESXi, KVM</td>
<td>Java, C, C++, Node</td>
<td>very active project with 103 contributor and +44 releases</td>
<td>11</td>
</tr>
<tr>
<td>RustyHermit</td>
<td>KVM</td>
<td>C, C++, Fortran, Go</td>
<td>active 9 contributors</td>
<td>1</td>
</tr>
<tr>
<td>Hermitcore</td>
<td>KVM</td>
<td>C, C++, Fortan, Go</td>
<td>active 10 contributors</td>
<td>5</td>
</tr>
<tr>
<td>ClickOS</td>
<td>Xen</td>
<td>C++</td>
<td>supported by NEC</td>
<td>3</td>
</tr>
<tr>
<td>MiniOS</td>
<td>Xen</td>
<td>C++</td>
<td>supported by XEN project, active project</td>
<td>2</td>
</tr>
<tr>
<td>Ling</td>
<td>Xen</td>
<td>Erlang</td>
<td>has not been updated since 2015</td>
<td>1</td>
</tr>
<tr>
<td>HaLVM</td>
<td>Xen</td>
<td>Haskell</td>
<td>has not been active since 2018</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4. Unikernel Frameworks
Motivations

Results

• Security
• Performance
• Supporting Technology
• Resource Optimization
• Service Modularity
• Service Isolation
• Reduced Costs
• Personal Motivations
Benefits

Results

- Performance ↑
- Resource Optimization ↑
- Security ↓
- Service Isolation
- Deployment
- Supporting Technologies
- Reduced Costs
- Service Modularity
Issues

Results

• Maturity

• Technologies
  • Lack of Multi-Processing
  • Development Process
  • Vendor Lock-in

• Compatibility

• Management

• Security

• Resource Utilization
  • Throughput is slightly higher than containers due to the lack of a userspace copy
  • Transmission performance is lower due to higher CPU usage
Open Questions

• Security threats are not clear

• Not clear when unikernels should be used

• Unikernels management not clear
  • Do they need a separate management layer?
  • Do we need a K8 like platform?
Conclusions

- Promising technology
- Multiple implementations
- Still in a early development stage
- Not easy to use and manage