# A Linux in Unikernel Clothing

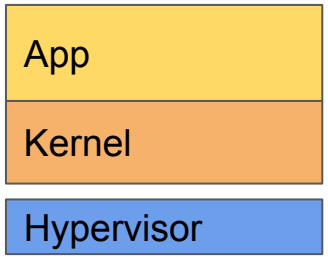
**Hsuan-Chi Kuo**[+], Dan Williams[*], Ricardo Koller[*] and Sibin Mohan[+]

[+]University of Illinois at Urbana-Champaign
[*]IBM Research

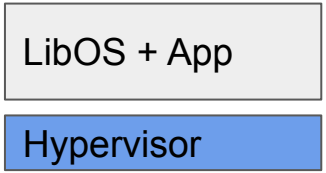
**Lupine**

IBM **Research**

# Unikernels are great




- Heavy
- Inefficient

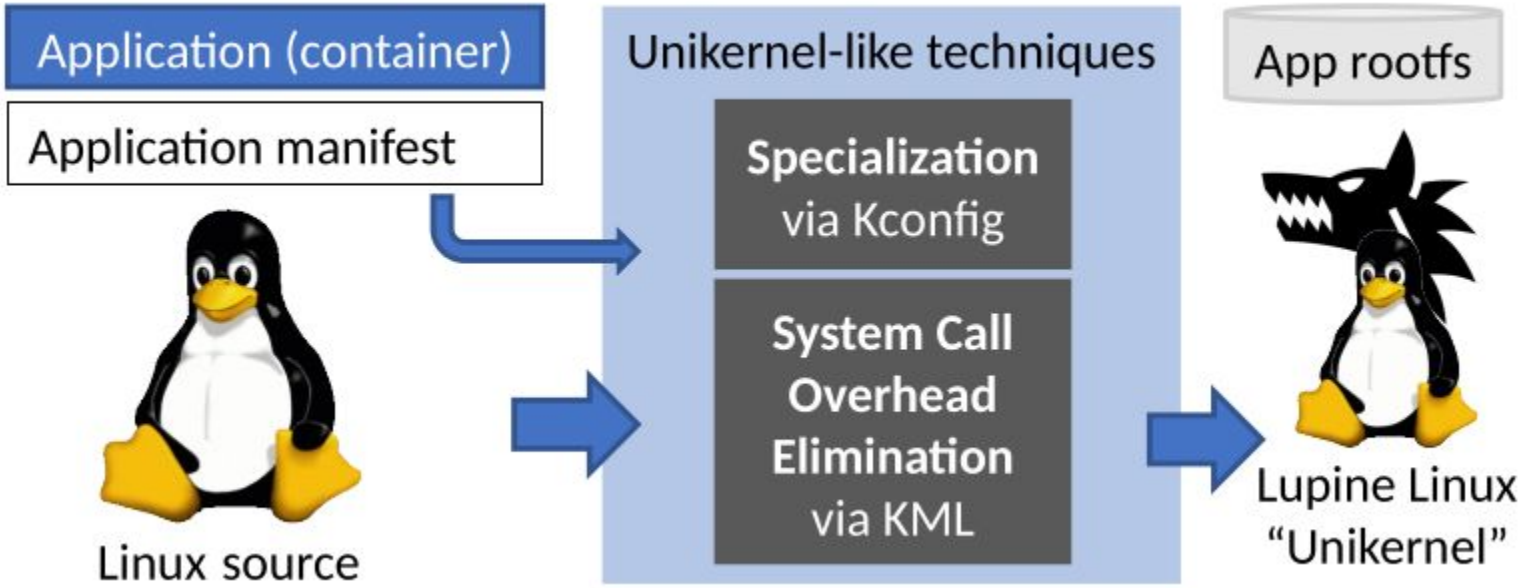



- Small kernel size
- Fast boot time
- Improved performance
- Better security

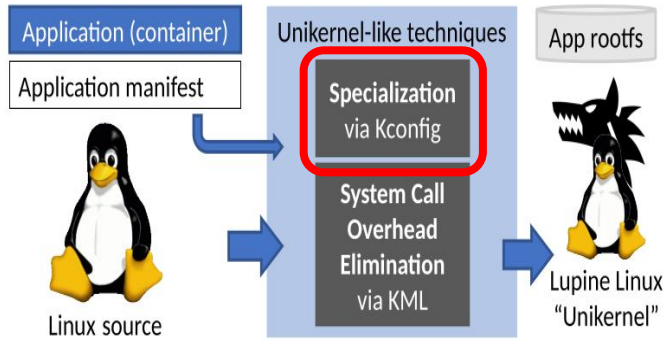
**BUT**: Unikernels lack full Linux Support

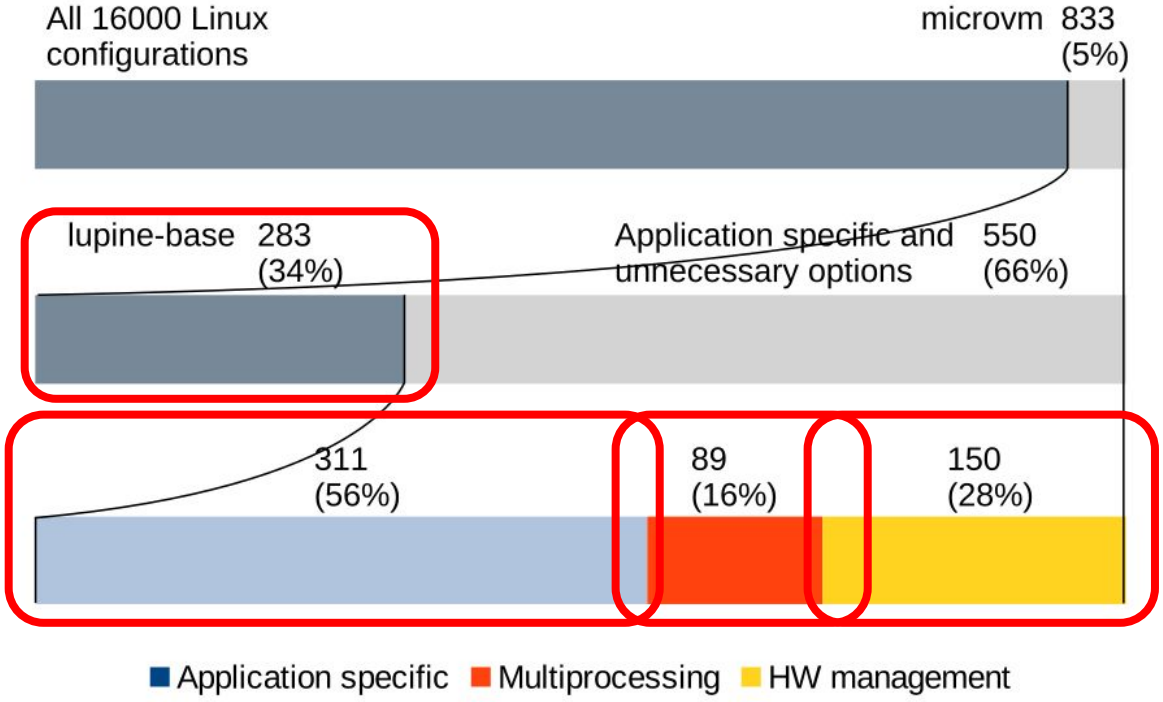
- Hermitux: supports only 97 system calls
- OSv:
  - Fork() , execve() are not supported
  - Special files are not supported such as /proc
  - Signal mechanism is not complete
- Rumprun: only 37 curated applications
- Community is too small to keep it rolling

# Can Linux behave like a unikernel?

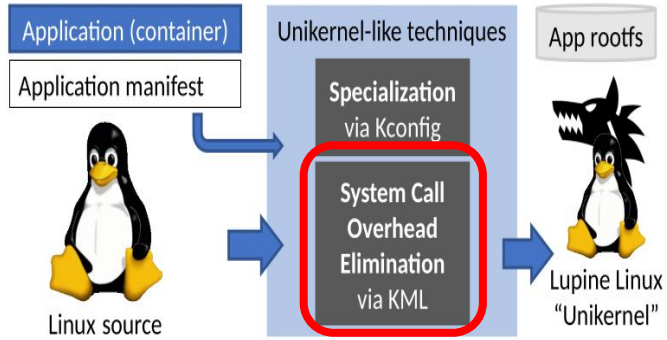


# Lupine Linux



# Lupine Linux

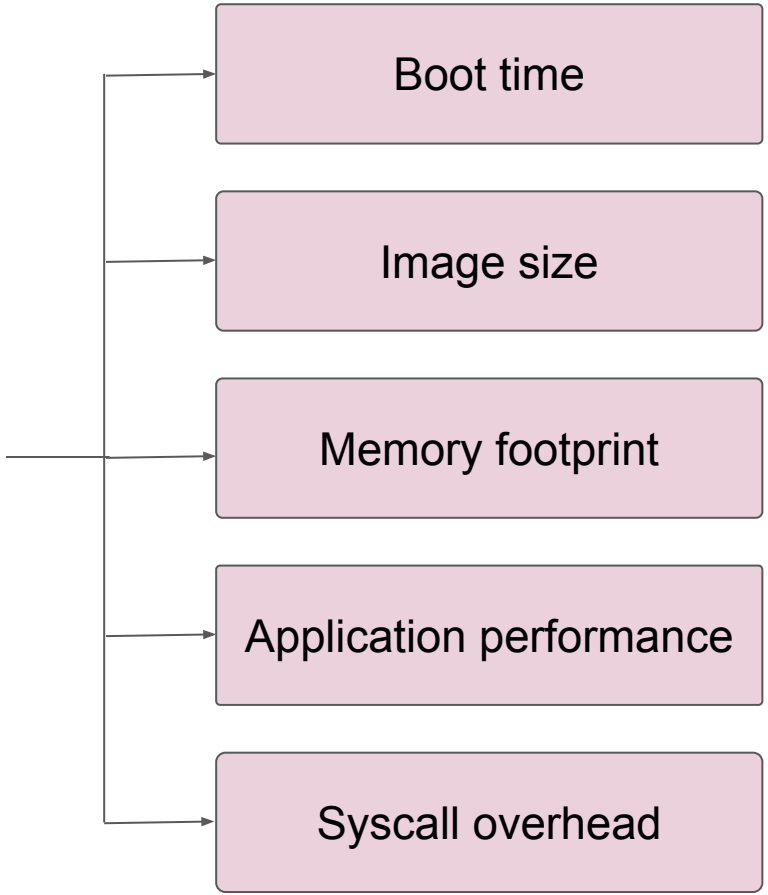



- Kernel mode Linux (KML)
  - Enables normal user process to run in kernel mode
  - Processes can still use system services such as paging and scheduling
  - App calls kernel routines directly without privilege transition costs

- Minimal patch to libc
  - Replace syscall instruction to call
  - The address of the called function is exported by the patched KML kernel using the vsyscall
  - No application changes/recompilation required

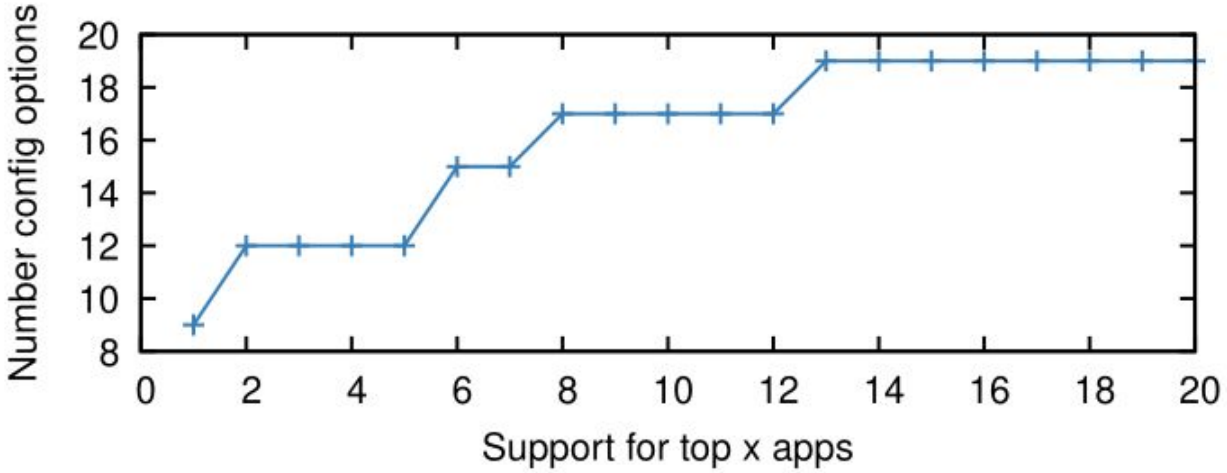
# Evaluation Metrics

Based on: **Unikernel benefits**



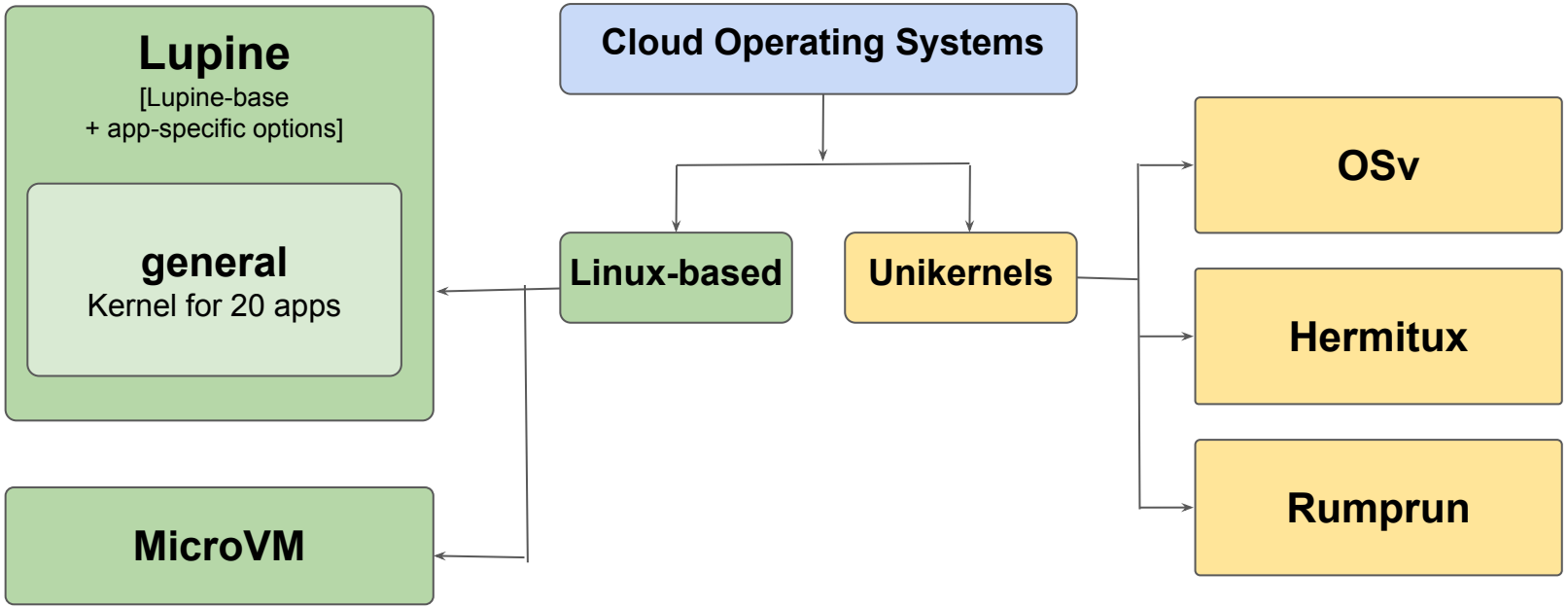
# Configuration diversity

- 20 top apps on Docker hub (83% of all downloads)
- Only 19 configuration options required to run all 20 applications: **lupine-general**



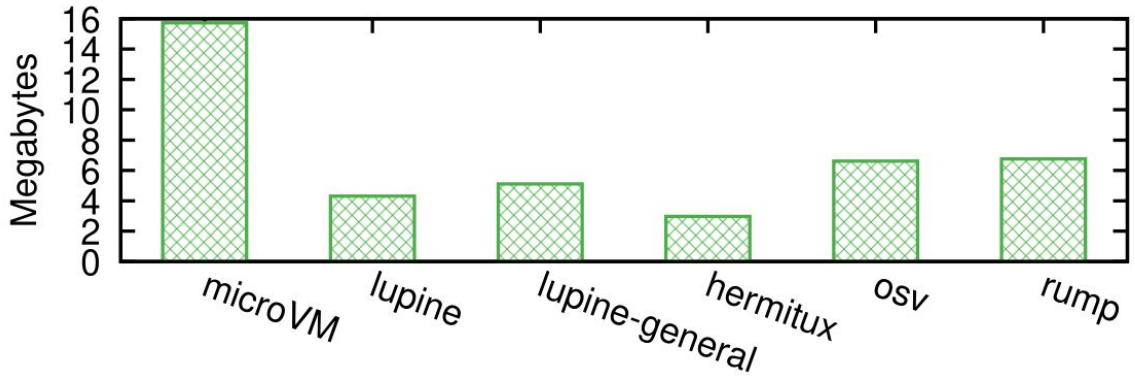

| Name | Downloads | Description | # Options atop *lupine-base* |
|---|---|---|---|
| nginx | 1.7 | Web server | 13 |
| postgres | 1.6 | Database | 10 |
| httpd | 1.4 | Web server | 13 |
| node | 1.2 | Language runtime | 5 |
| redis | 1.2 | Key-value store | 10 |
| mongo | 1.2 | NOSQL database | 11 |
| mysql | 1.2 | Database | 9 |
| traefik | 1.1 | Edge router | 8 |
| memcached | 0.9 | Key-value store | 10 |
| hello-world | 0.9 | C program “hello” | 0 |
| mariadb | 0.8 | Database | 13 |
| golang | 0.6 | Language runtime | 0 |
| python | 0.5 | Language runtime | 0 |
| openjdk | 0.5 | Language runtime | 0 |
| rabbitmq | 0.5 | Message broker | 12 |
| php | 0.4 | Language runtime | 0 |
| wordpress | 0.4 | PHP/mysql blog tool | 9 |
| haproxy | 0.4 | Load balancer | 8 |
| influxdb | 0.3 | Time series database | 11 |
| elasticsearch | 0.3 | Search engine | 12 |

# Evaluation - Comparison configurations



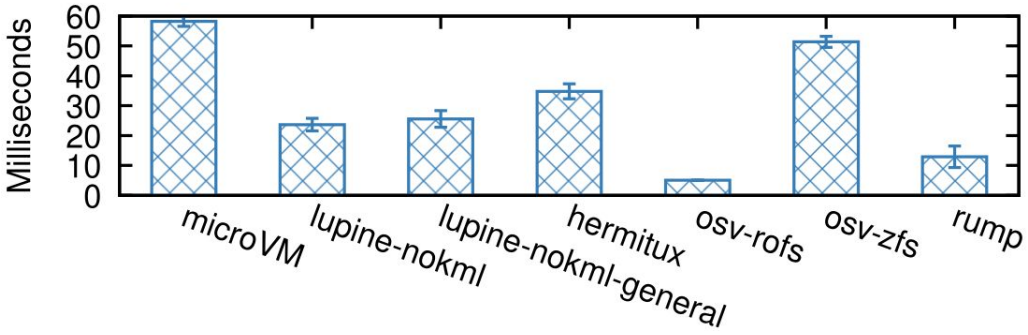
# Evaluation - Image size

- Configuration is effective
- 4 MB
- 27% of microvm
- Even lupine-general produces smaller images than Rump, OSv



# Evaluation - Boot time

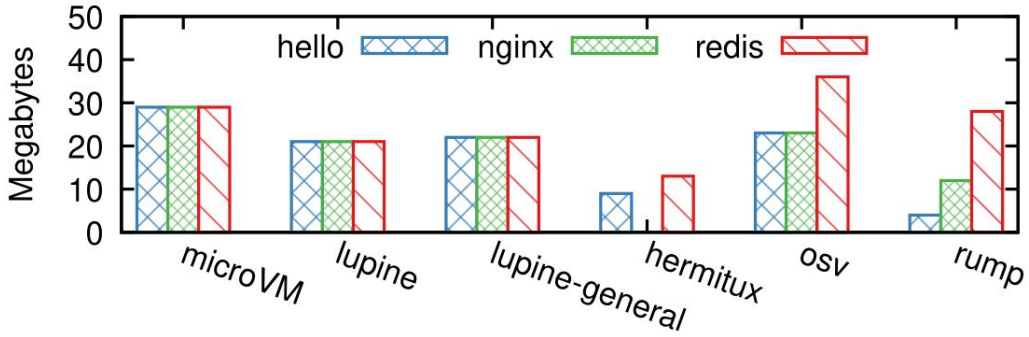
- OSv boot heavily depends on FS choice
- Lupine boot time without KML*
- Even lupine-general boots faster than Hermitux, OSv




*KML incompatibility with CONFIG_PARAVIRT

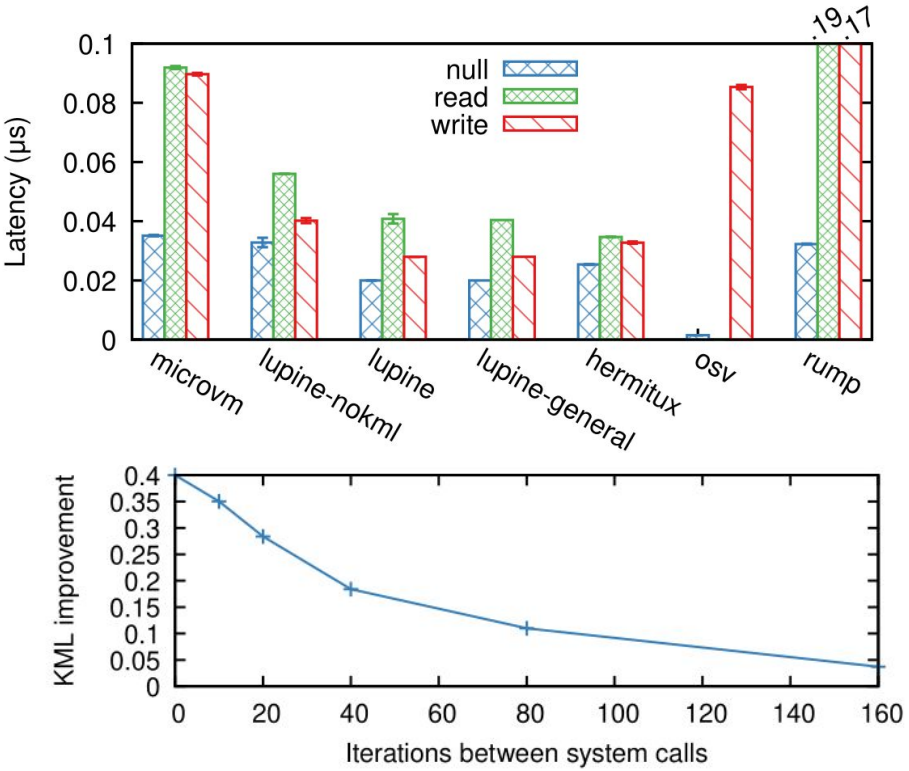
# Evaluation - Memory footprint

- Repeatedly tested app with decreasing memory allotment
- Choice of apps limited by unikernels
- No variation in lupine: lazy loading makes binary size irrelevant



# Evaluation - System call overheads

- Lmbench
- 56% improvement over microvm from specialization
- Additional 40% from KML
- KML benefit vanishes quickly in more realistic workloads



# Evaluation - Application performance

- Throughput normalized to microVM
- Application choice limited by unikernels
- Lupine outperforms microVM by up to 33%
- Lupine-general does not sacrifice performance
- Linux implementation is highly optimized

| Name | redis-get | redis-set | nginx-conn | nginx-sess |
|---|---|---|---|---|
| microVM | 1.00 | 1.00 | 1.00 | 1.00 |
| lupine-general | 1.19 | 1.20 | 1.29 | 1.15 |
| **lupine** | **1.21** | **1.22** | **1.33** | **1.14** |
| lupine-tiny | 1.15 | 1.16 | 1.23 | 1.11 |
| lupine-nokml | 1.20 | 1.21 | 1.29 | 1.16 |
| lupine-nokml-tiny | 1.13 | 1.13 | 1.21 | 1.12 |
| hermitux | .66 | .67 | | |
| osv | | | .87 | .53 |
| rump | .99 | .99 | 1.25 | .53 |

**Table 4.** Application performance normalized to MicroVM (Note: higher value is better).

# Lupine achieves unikernel benefits

| | |
|---|---|
| Image size | **4MB image size** |
| Boot time | **23ms boot time** |
| Application performance | **Up to 33% higher throughput** |

Best of all, it is still a Linux.

# Takeaways

- **Specialization is important:**
  - 73% smaller image size, 59% faster boot time, 28% lower memory footprint and 33% higher throughput than the state-of-the-art VM
- **Specialization per application may not be:**
  - 19 options (lupine-general) cover at least 83% of downloaded apps with at most 4% reduction in performance
- **System call overhead elimination may not be:**
  - only 4% improvement for macro-benchmark, unlike 40% for microbenchmarks
- **Lupine avoids common pitfalls**: has support for unmodified Linux applications, optimized implementation

# Thank you!

Hsuan-Chi Kuo hckuo2@illinois.edu
Dan Williams djwilla@us.ibm.com
Ricardo Koller kollerr@us.ibm.com
Sibin Mohan sibin@illinois.edu

https://synercys.github.io/projects/lupine

Lupine

IBM **Research**

Office of Naval Research
Science & Technology

C3SR