Benchmarking unikernels with Distributed Map Reduce

Akilan Selvacoumar[1], Robert Stewart[1], Hans-Wolfgang Loidl[1], and Ryad Soobhany[1]

[1]Heriot-Watt University, UK

Abstract

Unikernels are specialised, single-address-space machine images constructed by using library operating systems. This paper benchmarks unikernels on distributed systems, specifically benchmarking distributed map reduce applications on unikernels. The results will be compared against the same application running on a docker container and in a monolithic OS across five performance metrics.

Research Questions and Goals

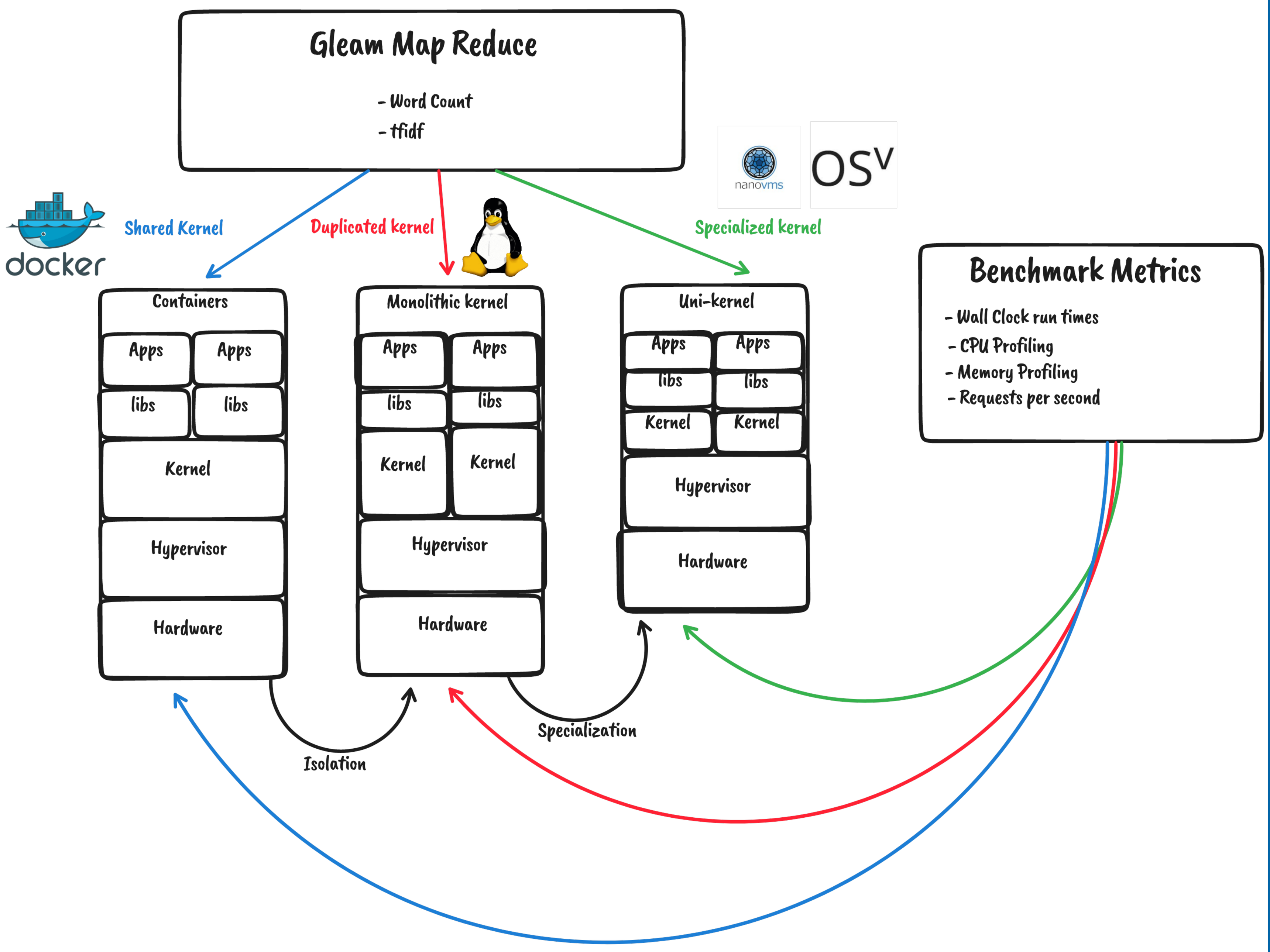
Research Questions:

- (RQ1) Would unikernels running map reduce applications *out-perform* monolithic OS or containers in terms of *wall-clock runtimes, CPU profiling and memory profiling*?
- (RQ2) In a distributed map reduce environment, if RQ1 is successfully proven in terms of speed-ups based on the metrics measured, what would be the *scalability and elasticity* of map-reduce applications on unikernels within Cloud infrastructures ?

Expirement Goals:

- (G1) Get Uni-kernels to build a map reduce application.
- (G2) Spawn multiple Uni-kernel images and run the distributed map reduce application.
- (G3) Building a open source implementation to measure performance on a map reduce application on Uni-kernels.

Research Methods



- ***Benchmark application***
  - *Word count*: Word count computes the number of occurrences of each word in a dataset and was selected as it is familiar to most users running map reduce application.
  - *tf-idf (term frequency–inverse document frequency)*: Statistical measure that evaluates how relevant a word is to a document in a collection of documents.
- ***Comparators***
  - *Unikernel*
  - *Monolithic OS*
  - *Docker Container*
- ***Benchmark Metrics***
  - *Boot-up time*
  - *Requests per second*
  - *CPU profiling*
  - *Memory profiling*

Tree Structure of the Gleam Uni-kernel image

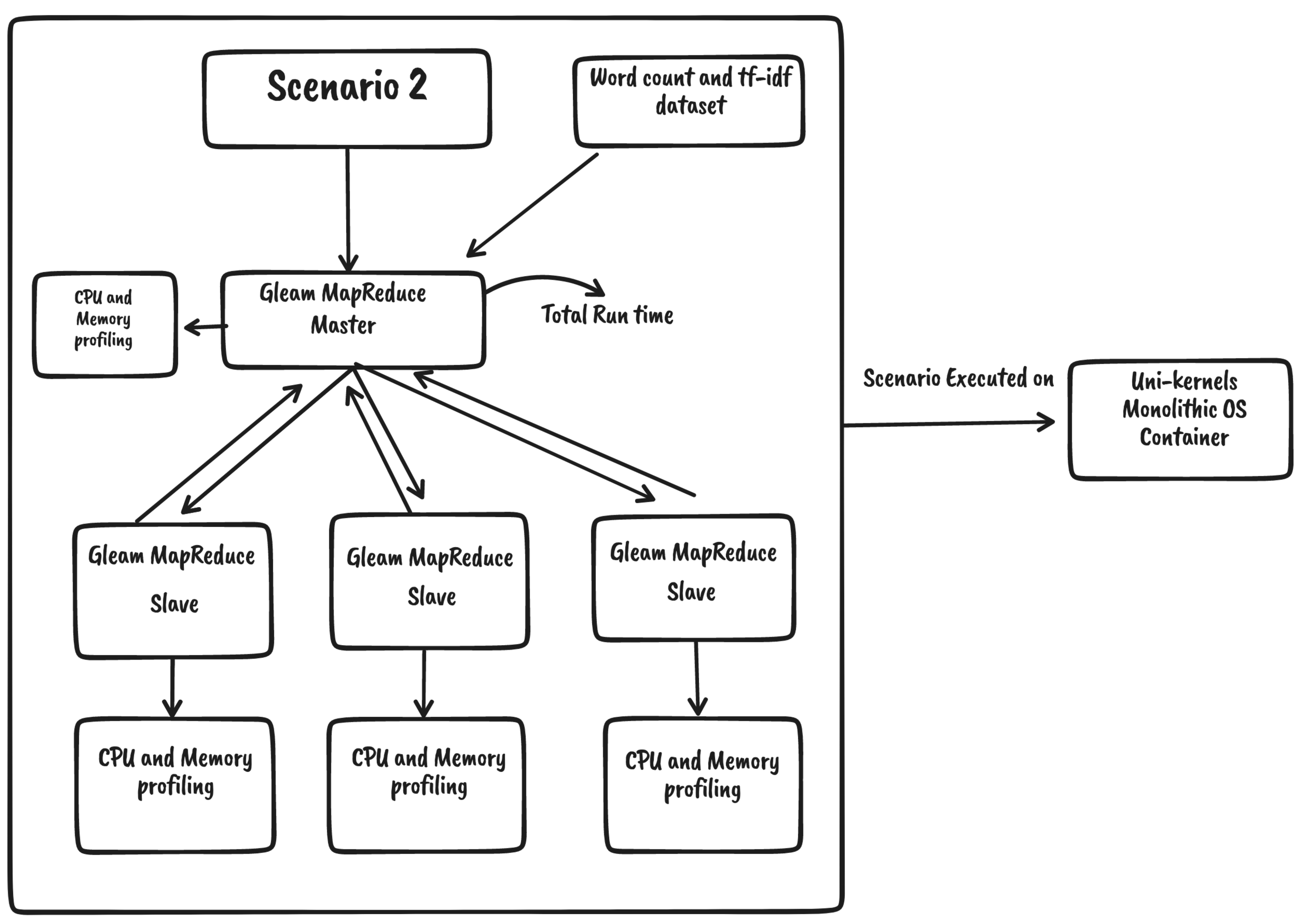
```
/
|   etc
|   |   ssl
|   |   |   certs
|   |   |   |   ca-certificates.crt
|   |   resolv.conf
|   |   passwd
|   (gleam-master or gleam slave)
|   proc
|   |   sys
|   |   |   kernel
|   |   |   |   hostname
|   |   self
|   |   |   exe -> (/gleam-master or /gleam-slave)
|   testing
|   lib64
|   |   ld-linux-x86-64.so.2
|   lib
|   |   x86_64-linux-gnu
|   |   |   libnss_dns.so.2
|   |   |   libpthread.so.0
|   |   |   libc.so.6
```

Image size for map reduce on Uni-kernels

	Binary size	Total NanoVM Image Size
Gleam Master	14.3 MB	30.8 MB
Gleam Slave	13.8 MB	30.2 MB

Experimental Setup

- *Scenario 1*: The *word count* and *tfidf* be excuted on a single node. This would answer question RQ1. The hardware for scenario 1 would be an Intel(R) Core(TM) i7-1065G7 with 4 cores allocated and with 16 GB memory.
- *Scenario 2*: This scenario benchmarks map reduce on distributed environment with a single master node and multiple slave nodes. The following scenario answers RQ2. The cluster configuration would be 1 master and 3 slaves nodes. The hardware specifications for the following experiments would be the 2.3 GHz Intel® XEON ® E5-2673 v4 for the Gleam master and slave nodes. Each of these nodes would be provided with 3.5 GB of memory.



Conclusion

The empirical evidence gained from these measurements, running two different scenarios, will be used to answer the three research questions stated in the introduction of the paper. The empirical data from running the experiments on unikernels will provide a better understanding on how unikernels perform on a distributed memory environments.