Paper review Title: *Performance and Scalability of Lightweight Multi-Kernel based Operating Systems,* **B. Gerofi, R. Riesen, et. al, In Proceedings of the 2018 IEEE International Parallel and Distributed Processing Symposium, May 21-25, 2018, pp. 116-125. P8**

Reviewer: swayanshu shanti pragnya

1. Summarize the (at most) 3 key main ideas.

The key ideas are as follows:

1. Light weight multi kernel operating system in HPC helps in leverages the processing time in processors. Light weight kernel (LWK) provides high performance for supporting tools compare to Linux.
2. Demonstration and evaluation of the results by Light weight multi kernels in a very large scale implication.

3.Performance improvisation is resulted by median evaluation.

2. State the main contribution of the paper

1. The paper gives the idea about the actual requirement of a multikernel.

2. The comparison between Linux and LWK leads to show the scalability of LWK over LINUX

3. The architectural comparison gives more clarity about the proposal of LWK and McKernel

3. Critique the main contribution

> *3.a. Rate the significance of the paper on a scale of 5 (breakthrough), 4 (significant contribution), 3 (modest contribution), 2 (incremental contribution), 1 (no contribution or negative contribution). Explain your rating in a sentence or two.*

> The concept of Multi kernel based operating system is really unique and it's published recently, which clearly notify the concept of Arrakis OS which was proposed in 2014.

> The evolution from Arrakis to multi kernel based OS is truly a remarkable research.

> The significance include Large scale performance which can be achieved by MultiKernel OS. So this is **significant contribution** for further research.

3. b. Rate how convincing is the methodology: do the claims and conclusions follow from the experiments? Are the assumptions realistic? Are the experiments well designed? Are there different experiments that would be more convincing? Are there other alternatives the authors should have considered? (And, of course, is the paper free of methodological errors.)

1. The software environment they experimented is Linux kernel version 3.10.0-327.22.2 and out of 64 CPU cores, only 4 cores were assigned for OS activities.

2. They deployed mOS LWK 3.10.0 which is again a Linux kernel. Few operations for supporting application like LAMMPS scaling experiment that results 100% in Linux but 121% in mOS for regular heap management is bit confusing.

3. The idea of mcOS, LWK is absolutely clear but the methods of using different cores using different software can be little hazy.

4. Only 2 type of scaling method which includes Lulesh2.0 and LAMMPS need to be explained more clearly.

They could have taken EOS or a particular program for the experiments in large scale not in small scale which they did.

4. c. What is the most important limitation of the approach?

1. The excessive modification sometime breaks Linux compatibility.

2. Replacing the memory management system and by eliminating daemon process can create problems for schedulers that may diverge the LINUX tree of patches.

3. Demonstration of Multi kernel ability of performance isolation was measured in small scale.

4. Rate the writing in the paper on a scale of 5 (great) to 1 (muddled), and justify your ranking. Did you have to re-read sections? Were algorithms clearly explained? Did the paper have a logical flow?

I will rate the paper as 3 because,

1. I re-read the methodology and performance evaluation section as the scaling techniques they used for ex- LAMMPS is bit confusing at the beginning.

2. The performance and whatever abstract they wrote, throughout the paper, they followed the path which means logical flow is there.

3. Using abbreviations without proper explanation like CCS-QCD, MCDRAM, MiniFE are of conflict.

5. *Answer one of the following three questions (whichever is most relevant for this paper):*

1. What lessons should system researchers and builders take away from this work?

Question 1 answers:

1. The concept of allocating functionalities to different cores using different components of operating system.

2. Different ways of scaling.

3. Conceptual and architectural view for mOS

4. Comparison of mOS and mcKernel against the linux baseline.

These points can be extended for further research.