iJournals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 11 Issue 11 November 2023

Identification of Underperforming Supercomputer Operating System Kernel Function Using Statistical Techniques and Distance Measures; and Generalizations to Bioengineering Domains

Author: Parthasarathy Srinivasan

Affiliation: **Oracle Corporation**

E-mail:ospark2477@gmail.com

DOI: 10.26821/IJSHRE.11.11.2023.111101

ABSTRACT

Computer Operating Systems have traditionally included scheduling components for effective use of machine resources, especially the CPU. These scheduling components exhibit stochastic behavior in the timing of the instruction schedule. This timing can be measured and analyzed for variations and departures from known and ideal component characteristics. This analysis is beneficial for comparing and identifying the categorical group (in this case the particular Operating System Call) which is deviant from known behavior and therefore the potential bottleneck in the overall system performance with respect to the instruction schedule timing.

Keywords: Operating System, kernel, Coherence, System call, Supercomputer, CPU, instruction, scheduler.

1. INTRODUCTION

With the above understanding in context, in this investigation, the author has analyzed the timing data gathered, using three statistical techniques:-

1) Find the variance of each of the categorical groups (O/S calls)as an indicator of system stability and identify the high variant bottleneck.
2) Perform an ANOVA to confirm or contraindicate the identification done above.
3) Compute divergence from standard device characteristics using well known distance metrics such as Kullback–Leibler divergence. These methods can be generalized to work upon data from Bioengineering domains and devices to identify trends in pharmacology for example.

METHODOLOGY

The experiment set up for the above purpose depended on the In Memory Data Grid (Oracle) Coherence technology, which was chosen as the application of interest, to study the effect of the variation in performance due to the O/S system calls.Coherence is a clustered map of Key Value pairs stored in member JVMs and served to clients with the help of the services built into it. Particularly, the ability to delete (remove) key value pairs from “partitions “within the container JVMs, depends internally on OS system calls which interact with the memory locations where the key value pairs are stored. The overall time taken by the deletion job was different when certain (kernel) parameters impacting the O/S call functionality were applied as against the time taken when these parameters were not applied. Attempt is made to identify the amount of the above variation statistically based upon the lines outlined in the abstract above (variance, KL divergence).The timing data of the system calls was collected using the strace utility.

KULLBACK–LEIBLER DIVERGENCE CODE

```
import matplotlib.pyplot as plt import numpy as np
from scipy.stats import norm


def kl_divergence(p, mu1, sigma1, mu2, sigma2):
return (1.0 / len(p)) * sum(np.log(norm.pdf(p[i], mu1, sigma1)) - np.log(norm.pdf(p[i],mu2, sigma2)) for i in range(len(p)))

def kl_divergence_cf(mu1, sigma1, mu2, sigma2):
return (np.log(sigma2/sigma1) + (np.power(sigma1, 2) + (np.power((mu1-mu2), 2)))/(2*np.power(sigma2, 2)) - 0.5)

def kl_divergence_js(p, q):
return (1.0 / len(p)) * sum(np.log(p[i]) - np.log(norm.pdf(q[i], mu2, sigma2)) for i in range(len(p)))

######## Sampling ########

mu1, sigma1 = 6.2731707317073E-5, 2.3220706697088E-5 # mean and standard deviation p = np.random.normal(mu1, sigma1, 10000)



mu2, sigma2 = 0.000062702, 0.000021 # mean and standard deviation

q = np.random.normal(mu2, sigma2, 10000)

######## Close Form - KL Divergence ########



print("Close Form kl Kullback–Leibler divergence _div : " + str(kl_divergence_cf(mu1, sigma1, mu2, sigma2)))

######## MC Sampling - KL Divergence ########



print("Monte Carlo Estimation kl_div : " + str(kl_divergence(p.tolist(), mu1, sigma1, mu2, sigma2)))

######## Plotting ########



count, bins, ignored = plt.hist(p, 30, density = True) plt.plot(bins, 1/(sigma1 * np.sqrt(2 * np.pi)) *
np.exp(- (bins - mu1)**2 / (2 * sigma1**2) ), linewidth = 2, color = 'r')

count, bins, ignored = plt.hist(q, 30, density=True) plt.plot(bins, 1/(sigma2 * np.sqrt(2 * np.pi)) *
np.exp( - (bins - mu2)**2 / (2 * sigma2**2) ), linewidth = 2, color = 'r')
plt.show()
```

2. GRAPH PLOTS

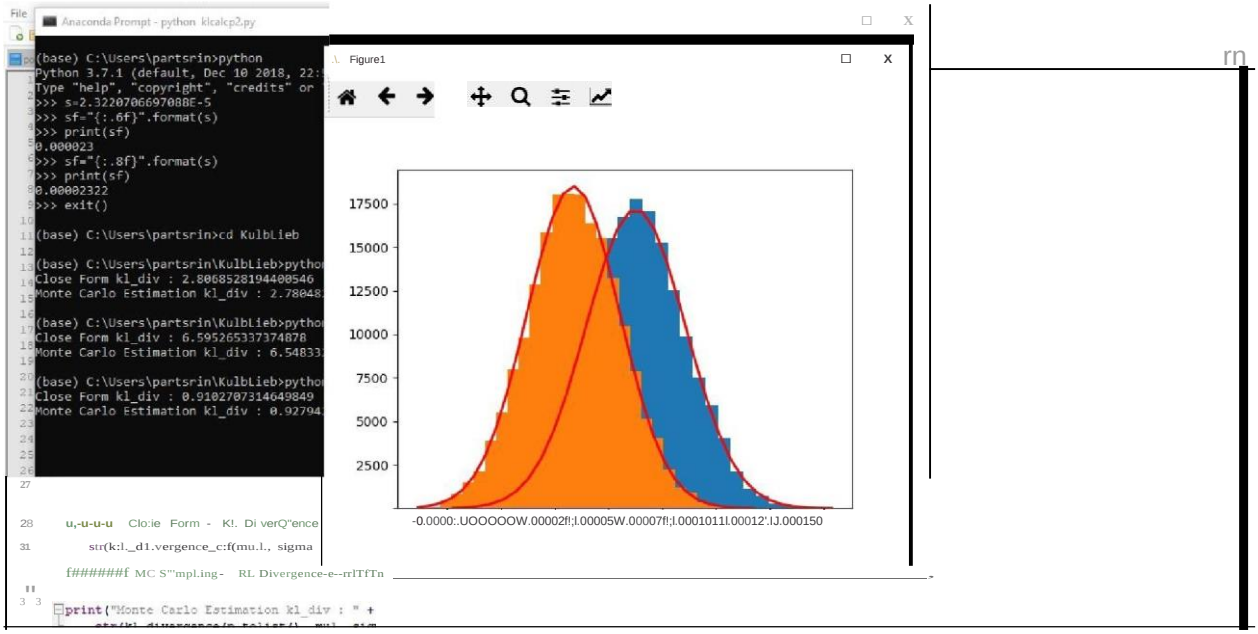

Fig 1: Comparative plot of KL divergence in timings of deviant system call with that of non-deviant system call (before applying kernel parameters) .

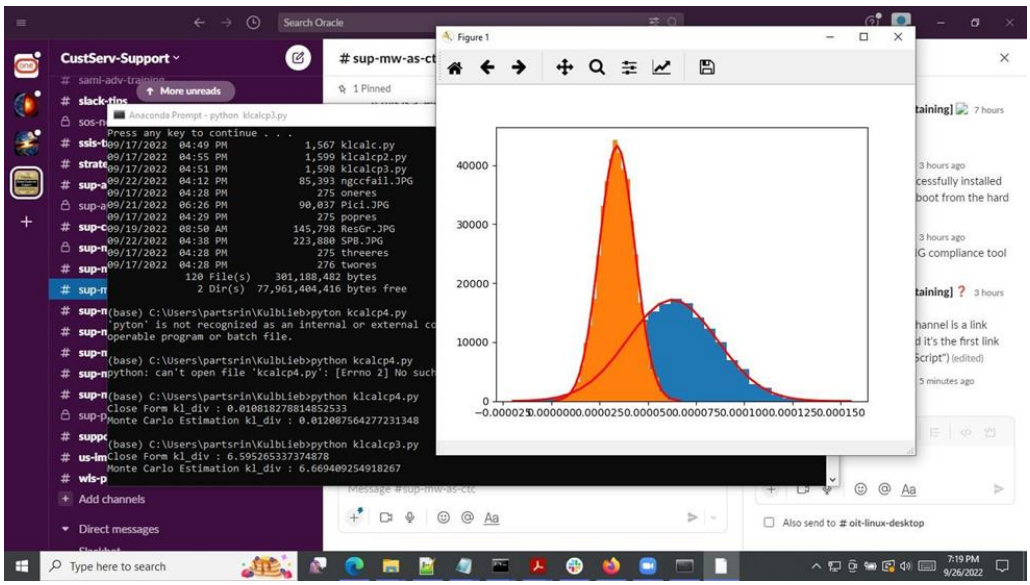

Fig 2: Comparative plot of KL divergence in timings of deviant system call with that of aggregate timings for all system calls (before applying kernel parameters).

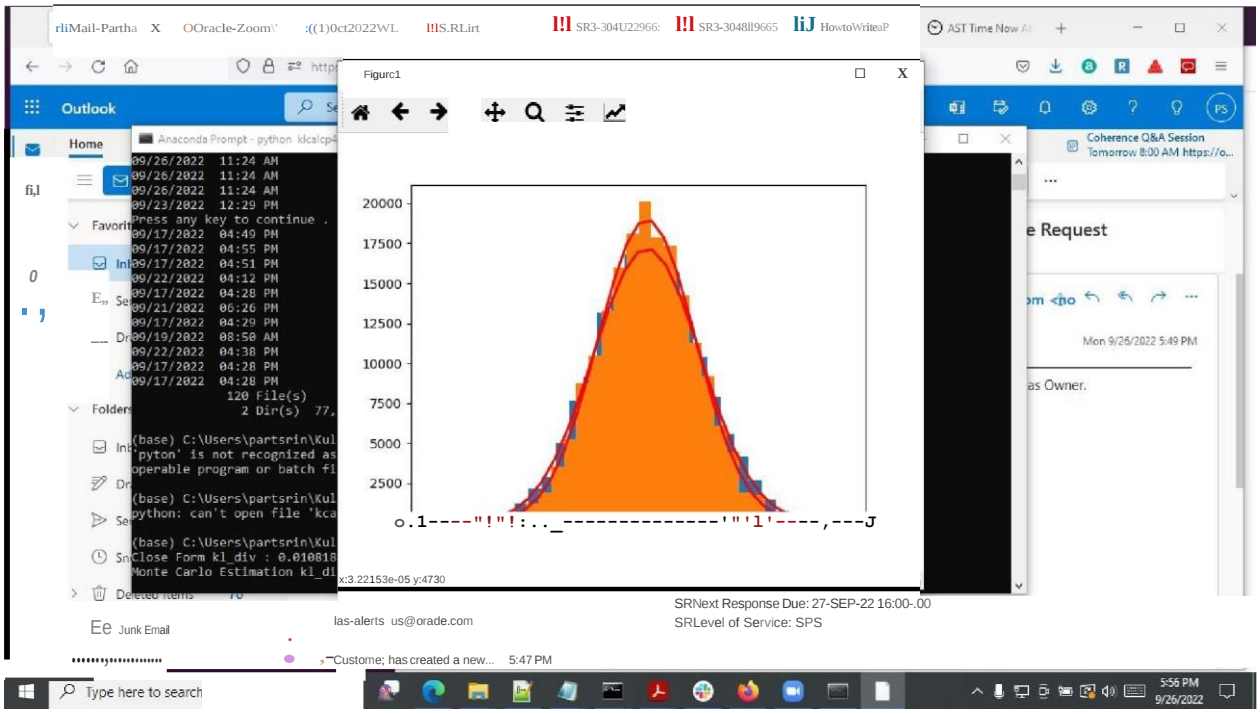

Fig 3: Comparative plot of KL divergence in timings of deviant system call with that of aggregate timings for all system calls (after applying kernel parameters).

3. RESULTS

The comparison of the KL Divergence between the "population data" (aggregate timing data of all the system calls put together) with that of one particular system call (F_ION_READ) showed that the F_ION_READ system call exhibited more divergence from the population when the kernel tuning parameters were not applied as compared to when those parameters were applied.

4. CONCLUSION

The experiment performed validated the statistical methodology proposed to identify underperforming system calls.

5. REFERENCES

[1]. Thushan Ganegedara, "Intuitive Guide to Understanding KL Divergence", in Towards Data Science, 2018

[2]. Kulback and Leibler, "On Information and Sufficiency", *in* Annals of Mathematical Statistics. **22** (1): 79–86, 1951

DISCLAIMER

The methodology, content, opinions, results and conclusions expressed/mentioned in this article is the sole works of the Author. Oracle Corporation or any of the business affiliates of Oracle Corporation and the Author (Parthasarathy Srinivasan) have no liability legal or otherwise arising from the use of the material in this article and/or any derivative works. As such the material is intended/provided only for academic use and without any warranty of any sorts.