2-2021

SPARC: Statistical Performance Analysis With Relevance Conclusions

Justin C. Tullos

Scott R. Graham

Jeremy D. Jordan

Pranav R. Patel

IEEE Open Journal of the
Computer Society

Received 31 January 2021; accepted 17 February 2021. Date of publication 19 February 2021;
date of current version 25 March 2021. The review of this article was arranged by Associate Editor N. K. Thulasiraman.

Digital Object Identifier 10.1109/OJCS.2021.3060658

SPARC: Statistical Performance Analysis With Relevance Conclusions

JUSTIN C. TULLOS [1] (Student Member, IEEE), SCOTT R. GRAHAM [1] (Senior Member, IEEE), JEREMY D. JORDAN[2], AND PRANAV R. PATEL [3] (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Air Force Institute of Technology, WPAFB, Ohio 45434 USA
[2]Department of Mathematics, Air Force Institute of Technology, WPAFB, Ohio 45434 USA
[3]Sensors Directorate of the Air Force Research Laboratory, WPAFB, Ohio 45434 USA

CORRESPONDING AUTHOR: JUSTIN C. TULLOS (e-mail: jctullos@gmail.com)

ABSTRACT The performance of one computer relative to another is traditionally characterized through benchmarking, a practice occasionally deficient in statistical rigor. The performance is often trivialized through simplified measures, such as the approach of central tendency, but doing so risks a loss of perspective of the variability and non-determinism of modern computer systems. Authentic performance evaluations are derived from statistical methods that accurately interpret and assess data. Methods that currently exist within performance comparison frameworks are limited in efficacy, statistical inference is either overtly simplified or altogether avoided. A prevalent criticism from computer performance literature suggests that the results from difference hypothesis testing lack substance. To address this problem, we propose a new framework, SPARC, that pioneers a synthesis of difference and equivalence hypothesis testing to provide relevant conclusions. It is a union of three key components: (i) identifying either superiority or similarity through difference and equivalence hypotheses (ii) scalable methodology (based on the number of benchmarks), and (iii) a conditional feedback loop from test outcomes that produces informative conclusions of relevance, equivalence, trivial, or indeterminant. We present an experimental analysis characterizing the performance of a trio of RISC-V open-source processors to evaluate SPARC and its efficacy compared to similar frameworks.

INDEX TERMS Performance benchmarking, RISC-V, relevance testing, statistical analysis.

I. INTRODUCTION

Benchmarking is a conventional practice in the computing domain for assessing a computer's performance relative to another. A standard set of representative programs are executed, covering a wide range of functionality, in order to capture performance metrics. But, the resulting metrics often lack sufficient statistical rigor for extensive analysis. A geometric mean, arithmetic mean, or performance ratio is reported and accepted at face value without indication of the sample distribution or a confidence level. It promotes misleading performance evaluations that permeate throughout the computing industry. Suffice to say, measures of central tendency have appropriate uses, but in some circumstances, thorough statistical analysis is needed for meaningful performance evaluation.

The Hierarchical Performance Testing (HPT) framework in [1], VarCatcher framework in [2], and methodology in [3] highlight the complexity of conducting a robust analysis and the lack of statistical rigor surrounding traditional computer performance comparisons. While [1] relies on difference hypothesis testing with non-parametric statistics, [2] and [3] cite the lack of relevant information at the conclusion of hypothesis testing as motivation for their respective custom frameworks. The challenge is developing a methodology that relies on fundamental statistical inference common across fields of research, is simple to customize based on user requirements, and provides results relevant to the performance, rather than a custom software framework. To achieve this, we address the limitations of HPT with respect to hypothesis testing and model a new framework that improves the efficacy of their method.

This paper forces a clear distinction between two ideas that are often pooled incorrectly in hypothesis testing: statistical significance and practical relevance. Significance is the ability of our statistical test to detect an effect size [4] and is

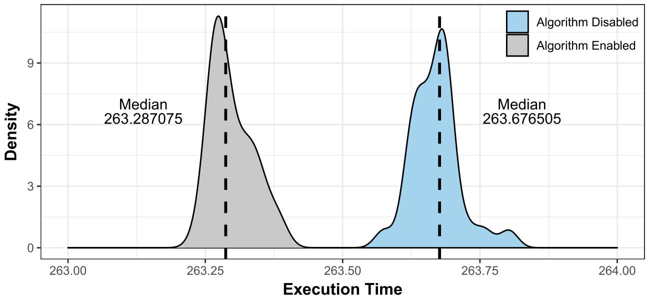

FIGURE 1. Comparing two distributions of execution time with an algorithm enabled and disabled.

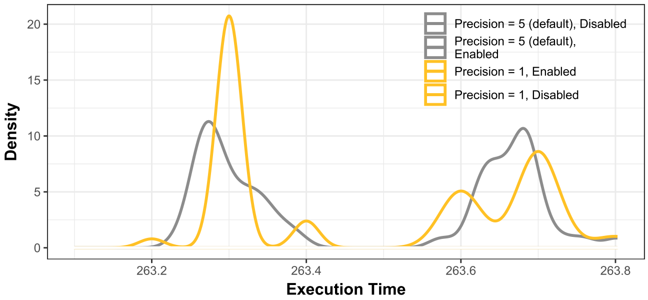

FIGURE 2. Comparing two distributions of execution time with the effects of differing decimal precision.

correlated to the type of test used. In difference hypothesis testing [5], failing to reject H_0 (i.e. lack of significance) does not imply lack of effect. Conversely, a difference hypothesis test that rejects H_0 (i.e. detects significance) expresses nothing about the practical relevance of the result. A statistical framework that only uses difference hypothesis testing is limited to identifying changes and does not address conditions of equivalence, or similarity between population samples within a margin.

To illustrate the limitation of difference tests, consider an exploratory data study to evaluate the performance of a security algorithm and its impacts to the Rocket RISC-V processor. We instantiated Rocket on a field programmable gate array (FPGA) and collected performance metrics. The experiment was performed with and without the algorithm enabled, 30 times each, and execution times were recorded. For statistical analysis, we used the HPT framework on the results to assess suitability towards the larger RISC-V processor performance evaluation conducted later in the text. We provide a density plot of two performance score distributions in Fig. 1. Hereinafter, we limit decimal precision to three digits with rounding for display purposes only, actual experiment calculations are conducted without rounding. The difference hypothesis test resulted in a statistical significance between the two distributions as shown in the figure. The median execution time of the program with the algorithm enabled as compared to disabled is 263.287 seconds and disabled is 263.677 seconds, or a percentage difference of 0.148% between them. But, the practical relevance of 0.1479% in our application was minuscule. We would have concluded the two execution times as approximately equivalent. Primarily conducting a difference hypothesis test excluded a condition in which both distributions would be considered equivalent.

This analysis highlights another key limitation of difference tests: the constructed null hypothesis is illogical and a difference is always detected with sufficient samples [6], [7]. In our analysis, the null and alternate hypotheses tested either a 0 difference between the two continuous response variables, or a difference detected, respectively. The test is structured given H_0 being true and if the probability distribution of our test statistic is low, then H_0 is rejected. But, this structured argument for point or exact null is a fallacy and has been debated for decades [6], [7], [8], [9], [10], [11]. The probability that a continuous random variable assumes any specific value is zero [12]. Likewise, with sufficient population samples the test will always detect a difference [13].

In embedded system performance evaluation, inadvertent data manipulation often occurs either due to rounding, or with an insufficient context of data output. Both can lead to incorrectly assuming that two response variables are equal or that the difference between them is 0 and affect the study. Returning to the example experiment, observe the original density plot with and without the algorithm enabled in Fig. 2. We graph the response variable density, execution time, which defaults to decimal precision of 5 based on the output software code. We overlaid the plots with a modified response variable density, by deliberately rounding data to decimal precision of 1. As shown in the figure, the characterization of our data distribution has altered significantly. Notably, the illustration fails to capture how altered data can proliferate into a difference hypothesis test. The differences to the *n*th decimal that once characterized our data are filtered out along with any insightful conclusions that could be derived. While it might seem obvious, we highlight the issue after encountering it in computer performance analysis research within the field. The aforementioned criticisms are not limited solely to computer performance evaluations, but to the field of null difference hypothesis testing in general.

To address the limitations illustrated above, we propose an improved statistical framework called SPARC. This appears to be the first computer performance analysis approach to combine difference and equivalence hypotheses tests and use the results to form four conclusions [14], [15] relevant to a computer performance evaluation under study. The main contributions of this paper are summarized below.

- Proposed a non-parametric framework, permitting analysis under distribution-free statistics tests, and developed with a straightforward procedure for implementation. Difference tests are conducted with a Wilcoxon Signed-Rank Test for paired computer performance observations for detecting statistically significant distributions. Subsequently, equivalence within a median tolerance

is assessed for distributions statistically significant but practically irrelevant.
- Developed a methodology inspired by HPT [1]. Minimized the false positive error rate using a multiple hypotheses error correction. It provides scalability based on the number of benchmark programs executed without inflating the error rate.
- Implemented SPARC framework enhances analysis with a conditional feedback loop that discriminates between overpowered or underpowered performance evaluations.
- Evaluated the new methodology with a performance evaluation consisting of a trio of RISC-V softcore processors instantiated on a field programmable gate array (FPGA).

The remainder of this paper is organized as follows. Section II summarizes related work and introduces the motivating HPT framework. Section III provides key fundamentals of statistical analysis with respect to difference and equivalence hypothesis testing. In-depth procedures are listed for constructing an equivalence margin and to conduct analysis with the framework. It also addresses limitations of non-parametric statistics in error correction and sample size estimation. In Section IV, an experiment is conducted between RISC-V softcore processors and the performance comparison analyzed with SPARC. Section V concludes the text with final remarks and future work.

II. RELATED WORK

This section summarizes the Hierarchical Performance Testing (HPT) framework methodology published in [1], which provides a statistical analysis framework for comparing the performance between two computers. Additionally, we review research that models the distribution of computer performance data through clustering. The following section will then establish our methodology inspired by the HPT framework.

A. BENCHMARKS

There are several benchmark programs specifically developed to evaluate a computer system's performance, often bundled together as a suite of applications. The System Performance Evaluation Corporation (SPEC) [16] is a popular example of a benchmark suite that can be compiled and executed on a variety of computer architectures. To compare two systems, one merely needs to execute a given benchmark on each system, after which the execution times can then be appropriately compared.

In some cases, benchmark programs may be very specialized in order to test specific functionality of a system under test (SUT); examples include testing floating-point operations or integer multiplication. After the SUT completes a benchmark, performance metrics are reported as time-based or throughput. Often, they are developed as separate software applications rather than originating from a sole benchmark suite such as SPEC. Over time, users consolidate the applications into a suite that is suitable for their requirements. An example is the benchmark suite, RV8, compiled for the RISC-V instruction set architecture used later in the text.

B. HPT FRAMEWORK

In [1], the authors developed the non-parametric HPT framework to promote statistically sound computer performance evaluations. The framework is a methodology using difference hypothesis tests to compare benchmark suite results between two computers to determine superiority. The authors reveal common errors made with respect to parametric and non-parametric statistics while conducting performance evaluations.

Chen *et al.* illustrates the improper use of parametric statistical tests, such as the t-test, on non-normally distributed computer performance data. If the data collected from a computer benchmark is not properly characterized prior to statistical analysis, it could be incorrectly assumed to be parametric instead of non-parametric. Without appropriate verification tests, an assumption of the underlying distribution of the data may contribute to a faulty analysis and misled conclusion of the comparison. They evaluate a SPEC benchmark suite comparison that displayed a skewed non-normal distribution using the t-test which resulted in transforming the data to normality. The t-test concluded the under performing computer was superior, demonstrating the deficiency in assuming a distribution.

The Central Limit Theorem (CLT) is often used to characterize distributions as approximately normal given a large sample size [4]. Frequently, a minimum sample size of 30 or more is referenced in statistics to employ the CLT. Although, this was disputed for computer performance distributions in [1] with an experiment consisting of 32 000 benchmark performance scores. The analysis reveals that a sample size of approximately 500 observations still deviated from normality, but could be sufficient to utilize the CLT. Executing a number of benchmarks within a suite, 500 times each, appears inefficient based on the inconsistency of the data.

Many sources of variability and non-determinism exist within a computer system and the complex layers of interactions they are comprised of, discussed in [1]. Further, published performance evaluations routinely omit confidence intervals (CI), which provides a measure of the randomness of a variable and accuracy estimate of observed data [12].

Performance evaluations often report a collection of mean completion times or relative speedups and declare one to be superior, with little, if any, documentation of statistical methods used in the comparison. While the mean completion time or speedup serves a purpose as a visual exploration of data, incorporating additional statistics provides insight into the origination, or population, of the sampled data. Such insight is fundamental in determining the accuracy of observations and conclusion. Excluding statistical analysis undermines the original intent behind the performance comparison.

Thus, the authors in [1] developed the non-parametric framework to promote statistically sound computer performance evaluations. The HPT framework is a methodology using hypothesis tests to compare benchmark suite results between two computers to determine superiority. The significance level, α, is chosen prior to conducting the hypothesis tests; standard suggestion is 0.05 for one-tailed or 0.10 for two-tailed hypothesis tests.

In order to analyze the performance between two computers on a suite of benchmarks, a series of steps, which comprise the HPT framework were outlined by [1]. We provide the following abridged procedure for reference, and build upon it later in the text. Suppose a benchmark suite is used that contains n benchmarks and each benchmark is repeated m times. Matrices $C_A = [a_{i,j}]_{n\times m}$ and $C_B = [b_{i,j}]_{n\times m}$ must be constructed for both computers; rows represent the nth benchmark and columns represent the mth benchmark repeat of performance scores [1].

For each benchmark, a null (H_0) and alternative (H_1) hypotheses are tested for significance using a Wilcoxon Rank-Sum Test. If the results show statistical significance, reject H_0 that both computers are equivalent; else fail to reject H_0. After the Wilcoxon Rank-Sum Test is complete for all n benchmarks, assign to a new column the score representing difference in medians on significant results for each benchmark or assign a 0 for insignificant results.

Concluding HPT is a comprehensive hypotheses test consisting of H_0 of general equivalent performance or H_1 general superior performance [1]. A Wilcoxon Signed-Rank Test is completed on the difference in median performance scores to either reject H_0 or fail to reject H_0 at the significance level.

C. CLUSTER METHOD TO DESCRIBE UNDERLYING PERFORMANCE SCORE DISTRIBUTIONS

In [17], the authors established a clustering method to model distributions of computer performance metrics. Observed data from benchmarking was non-parametric and density plots indicated bimodal and multimodal distributions. They surmised that the non-parametric distributions are a Gaussian mixture, a combination of multiple Gaussian distributions of clustered multivariate data. The clustering method determines population estimation parameters that could be used with more powerful parametric statistical tests over non-parametric.

III. PROPOSED FRAMEWORK METHODOLOGY

In this section, we introduce our SPARC method, which incorporates equivalence tests and family-wise error correction associated with multiple hypotheses tests. It reduces Type I errors and supports various conclusions for relevant and practical results of a performance evaluation.

A. ELEMENTS OF RELEVANT STATISTICAL PERFORMANCE EVALUATION

A key element in any statistical experiment, including benchmarking, is designing the experiment such that results provide valid statistical information required for analysis. Design of experiments [18] is a field of study dedicated to this aim. Our methodology focuses primarily on non-parametric statistical tests after benchmarking data has been collected and assumes the experiment uses an appropriate design. But we address three essential elements for consideration prior to conducting any data collection: 1) standardized hypotheses notation; 2) family-wise error correction; and 3) sample size estimation. Error correction and sample size estimation are implicitly linked when considering multiple benchmarks for statistical analysis; the number of samples affects the significance of a statistical analysis and the significance is affected by the overall error rate for the evaluation.

1) STANDARDIZED HYPOTHESES NOTATION

Before we formally present the rationale behind equivalence tests, we provide a standardized hypotheses notation used throughout the rest of this paper. In the introduction, we discussed limitations of an analysis that uses difference hypotheses which motivated the addition of equivalence tests. First, we introduce the term positivist theory derived from [19], to describe difference hypothesis tests. That is, the null hypothesis of a difference test H_0 is often defined as the lack of an effect or no difference between effects and is tested against an alternative hypothesis H_1 of significant effect or difference [5]. Positivist theory simply denotes H_0 and H_1 hypotheses of difference tests as H_0^+ and H_1^+, incorporating the $+$ symbol to reflect testing for a significant effect. Likewise, we introduce the term negativist theory [19], to describe equivalence hypotheses that test for a lack of effect (i.e. equivalence). Negativist theory defines the equivalence hypotheses H_0 and H_1, as H_0^- and H_1^-. We use the positivist and negativist theory notations H_0^+, H_1^+, H_0^-, and H_1^- in this text to differentiate between difference and equivalence hypotheses.

2) MULTIPLE HYPOTHESIS ERROR CORRECTION

In the following, let $X = x_{i,1}, x_{i,2}, \ldots, x_{i,m}$ and $Y = y_{i,1}, y_{i,2}, \ldots, y_{i,m}$ for $i = 1, 2, \ldots, n$ denote independent samples of performance scores from Computer X and Computer Y on the nth benchmark, respectively. Each hypothesis test performed in a multiple evaluation experiment increases the probability of rejecting H_0 when H_0 is true (Type I error) defined as the Family-Wise Error Rate (FWER) [20]. In other words, in a family of comparisons that are related the false positive error rate increases [20]. The worst-case FWER for n total benchmarks tested at an α_n is:

$$FWER \leq 1 - (1 - \alpha_n)^{\beta+1}, \tag{1}$$

where β is the number of benchmark tests plus an additional overall hypothesis of general performance.

Using an appropriate error correction method, we can control the family-wise error in the performance evaluation while still providing statistically significant results [20]. Each hypothesis test used to analyze a benchmark increases the FWER and requires correction. There are two methods we introduce here, the Bonferroni Correction [21] and Holm-Bonferroni Correction [22]. Each α correction method has its advantages and disadvantages that should be considered depending on a study's requirements. In the RISC-V evaluation later in the text, we use the Bonferroni Correction.

There are two benefits for using the Bonferroni Correction. First, it is a simple correction applied to every test in our study and, second, it allows calculating confidence intervals across benchmark comparisons [22]. It is widely used but has also been criticized as overcorrecting α to reduce Type I errors and subsequently reducing the probability of detecting any significance [20]. The method to calculate an error corrected α_{New} is as follows:

$$\alpha_{New} = \frac{\alpha_{Old}}{(n+1)}, \tag{2}$$

where n is the total number of benchmarks planned plus the overall hypothesis test and α_{Old} is the overall requested alpha (0.05 for one-tailed, 0.10 for two-tailed tests).

After α_{New} is calculated, the p-value of each benchmark hypothesis test is compared with α_{New} to either reject H_0 or fail to reject H_0:

$$p_n \leq \alpha_{New}, \tag{3}$$

where p_n represents the p-value of the nth benchmark.

An alternative method that does not overcorrect α is the Holm-Bonferroni Correction [22]. The method corrects sequentially, calculating α_{New} for each p-value comparison. While it provides stronger statistical power compared to Bonferroni, there is added complexity to determine confidence intervals based on a changing α_{New}. We present the procedure as it pertains to our framework as an option if confidence intervals are not required. Let p_n be denoted as the p-value calculated after conducting the Wilcoxon Signed-Rank Test, for the nth benchmark. Sort in ascending order such that $p_1 < p_2 < \cdots < p_i$ for $i = 1, 2, \ldots, n$. Assign α_{New} based on ranks of the test until the first non-significant result is found (failed to reject H_0) and the correction is complete. Any further benchmark hypothesis tests are non-significant. The equation for this procedure is as follows:

$$p_n < \frac{\alpha_{new}}{i+1-n} \tag{4}$$

3) SAMPLE SIZE

In computer performance evaluations, determining the proper sample size is a fine balance between under or over sampling for a proper test. The significance (p-value) of each benchmark analysis is correlated with the sample size [23]. If an insufficient number of samples are collected from a benchmark, there is risk of an underpowered test (i.e. not providing a significant result due to a low p-value). If an over abundance of samples are collected, then the risk is an overpowered study that inefficiently used resources.

There are multiple ways to calculate sample sizes for a t-test statistic based on an effect size estimate, such as Cohen's D [24], if the underlying distribution is known or assumed. However, for non-parametric statistic tests we make no assumptions on the underlying distribution. The methods in [25], [26], however, illustrate how an estimated sample size can be determined for the Wilcoxon Signed-Rank Test if assumptions are made on the effect size and an unbiased estimator through a resampling process. We conclude there is merit in applying the techniques to a computer performance evaluation to reduce the number of benchmark repeats or increase power of the tests. At the same time, execution times are often non-deterministic which suggests resampling observations with prior data could affect the outcome or provide inaccurate sample size estimations. While there are no clear methods available for sample size estimation suitable for our framework, two of the resulting outcomes will report if a benchmark test was underpowered or overpowered.

B. EQUIVALENCE TESTING

Instead of testing the significance that performance scores from two computers are different, we introduce an approach called equivalence testing [27], [28]. In difference testing, we attempt to prove the alternative hypothesis H_1^+ of a significant statistical difference. If we fail to reject the null H_0^+ of no difference, we can only conclude there was a lack of evidence to reject H_0^+. We cannot conclude equivalence because it was not tested. By adding equivalence hypotheses tests to the framework, we have additional information to make inferences of a performance evaluation.

Equivalence testing is often found in clinical settings to assess whether the effect of two treatments or medications are within a predefined equivalence margin [27]. The burden of proof for equivalence resides in the alternative hypothesis H_1^-. An equivalence margin $[-\delta, \delta]$ establishes the range in which two variables contained within are considered practically equivalent at δ. In our context, the equivalence margin renders two statistically significant but practically irrelevant performance score distributions as equivalent if the response variable is within the predefined $[-\delta, \delta]$ interval.

One widely used method for equivalence testing is the Two One-Sided Tests (TOST) procedure in [29]. Choosing an appropriate equivalence margin δ for the TOST is paramount to a performance evaluation [27]; selecting a δ which is too stringent risks excluding practically equivalent performance scores and selecting a δ which is too broad risks false equivalence. [27] proposed either using past studies or pilot studies to establish a δ, but we consider this unsuitable for computer performance evaluations as preexisting data is either lacking or includes components specific to a system. Instead, we suggest setting it tailored to the evaluation depending on the motivation and context of the study. As an alternative, an equivalence $\delta = 5\%$ of the speedup ratio can be used between two computers on a benchmark with an equivalence margin of [0.95, 1.05]. The speedup ratio S is defined as:

$$S = \frac{Execution\_Time_{Old}}{Execution\_Time_{New}} \tag{5}$$

C. COMBINING DIFFERENCE AND EQUIVALENCE HYPOTHESES

Combining hypotheses tests for difference and equivalence leads to practical and relevant conclusions not possible individually. Hypothesis testing for difference supports conclusions for statistical significance but lacks conditions for practical irrelevance or equivalence. Conversely, equivalence testing supports conclusions on equivalent distributions but lacks conditions for substantial performance differences that are of interest. Therefore, the prevailing solution is a combination of difference and equivalence testing for practical relevance [14], [15].

This following outlines the procedures in our framework for combining the two types of tests for a relevant performance evaluation. Our method changes the Mann-Whitney (Wilcoxon Rank-Sum) Test in [1] to a Wilcoxon Signed-Rank Test for paired observations. Although, with minor alterations, our procedures can still be applied to the Mann-Whitney Test. The RISC-V processors evaluated in the next section necessitated a paired non-parametric test.

Suppose we are evaluating two computer's performance on a benchmark suite consisting of n benchmarks, each repeated m-times. Let (x_i, y_i) be the ith pair for $i = 1, 2, \dots, m$ for Computer X and Computer Y of m observations on the nth benchmark. Construct matrices $B_n = [x_{i,1}, y_{i,2}, r_{i,3}]_{m\times 3}$ for $n = 1, 2, \dots, n$ for n benchmarks. Let r_i denote the pairwise ratio x_i/y_i for $i = 1, 2, \dots, m$ and $M_{X/Y}$ denote the median pairwise ratio of performance. We use the Wilcoxon Signed-Rank Test under the assumption that the ratios r_i are continuous and symmetric around a common median $\theta = 1$ [1]. Difference hypotheses for the two-tailed Wilcoxon Signed-Rank Test are defined as:

- H_0^+: the median performance score ratio $M_{X/Y}$ of Computer X, Computer Y on the nth benchmark is symmetric around $\theta = 1$ (corresponding with no location shift from the benchmarks)
- H_1^+: the median performance score ratio $M_{X/Y}$ of Computer X, Computer Y on the nth benchmark is not symmetric around $\theta = 1$

Conduct a Wilcoxon Signed-Rank Test with an α corrected for family-wise error to either reject H_0^+ or fail to reject H_0^+. For brevity, we omit the procedure as it is readily available online or in statistics textbooks. However, we illustrate the procedure in detail for equivalence within a margin.

We utilize the non-parametric TOST Wilcoxon Signed-Rank Test for equivalence procedure in [30] with a median ratio [31] δ chosen *a priori*. Two one-sided tests are conducted to determine if the performance score distributions within the margin $[-\delta, \delta]$ are equivalent. Since we use a ratio performance, our equivalence margin becomes $[1 - \delta, 1 + \delta]$. Both tests must reject the null for equivalence to be established [15]. The upper bound equivalence $1 + \delta$ and lower bound equivalence $1 - \delta$ signed ranks are computed and tested separately.

The upper bound equivalence, δ, null (H_{01}^-) and alternative (H_{11}^-) hypotheses are defined as follows:

- H_{01}^-: the performance score ratio distribution x_i/y_i on the nth benchmark is greater than or equal to the upper bound equivalence $1 + \delta$
- H_{11}^-: the performance score ratio distribution x_i/y_i on the nth benchmark is less than the upper bound equivalence $1 + \delta$

The lower bound equivalence, $1 - \delta$, null (H_{02}^-) and alternative (H_{12}^-) hypotheses are defined as follows:

- H_{02}^-: the performance score ratio $M_{X/Y}$ on the nth benchmark is less than or equal to the lower bound equivalence $1 - \delta$
- H_{12}^-: the performance score ratio $M_{X/Y}$ on the nth benchmark is greater than the lower bound equivalence $1 - \delta$

Let $f_i = (x_i/y_i) - (1 + \delta)$ for $i = 1, 2, \dots, m$ denote the pairwise ratio for the mth observation minus upper bound $1 + \delta$ for Computer X, Computer Y on the nth benchmark. Let ψ_i denote the sign indicator of f_i as:

$$\psi_i = \begin{cases} 0, & f_i > 1 \\ -1, & f_i < 1 \end{cases} \tag{6}$$

Rank R_i for $i = 1, 2, \dots, m$ the absolute values $|f_1|, \dots, |f_i|$ in ascending order. The product $R_i\phi_i$ denotes the signed rank of f_i. The test statistic, W^-, for $1 + \delta$ is the sum of absolute values of negative ranks defined as:

$$W^- = \sum_{i=1}^{m} R_i\phi_i, \qquad i = 1, 2, \dots, m; \tag{7}$$

where m denotes the number of m benchmark observations.

Similarly, let $g_i = (x_i/y_i) - (1 - \delta)$ for $i = 1, 2, \dots, m$ denote the pairwise ratio for the mth observation minus lower bound $1 - \delta$ for Computer X, Computer Y on the nth benchmark. Let ψ_i denote the sign indicator of g_i as:

$$\psi_i = \begin{cases} 1, & g_i > 1 \\ 0, & g_i < 1 \end{cases} \tag{8}$$

Rank R_i for $i = 1, 2, \dots, m$ the absolute values $|g_1|, \dots, |g_i|$ in ascending order. The product $R_i\phi_i$ denotes the signed rank of g_i. The test statistic, W^+, for $1 - \delta$ is the sum of absolute values of positive ranks defined as:

$$W^+ = \sum_{i=1}^{m} R_i\phi_i, \qquad i = 1, 2, \dots, m; \tag{9}$$

where m denotes the number of m benchmark observations.

If $(m < 6)$, determine the exact p-value from Wilcoxon Signed-Rank Test tables for a one-sided test with α separately for both W^- and W^+.

If $(m \geq 6)$, the rank distribution is approximately normal [32]. Therefore, calculate the z-score as follows:

TABLE I. Relevance Conclusions

	Difference Test	Equivalence Tests	
Conclusion	Rej H_0^+?	Rej H_{01}^-?	Rej H_{02}^-?
Indeterminant	No	No	No
Trivial Difference	Yes	Yes	Yes
Relevant Difference	Yes	No	No
Equivalence	No	Yes	Yes

$$z_1 = \frac{W^- - \frac{M(M+1)}{4}}{\sqrt{\frac{M(M+1)(2M+1)}{24}}} \tag{10}$$

$$z_2 = \frac{W^+ - \frac{M(M+1)}{4}}{\sqrt{\frac{M(M+1)(2M+1)}{24}}} \tag{11}$$

Reject H_{01}^- if $z_1 \geq z_{1-\alpha}$ indicating $M_{X/Y}$ is within δ. If we fail to reject H_{01}^-, we do not calculate z_2 because equivalence does not hold. Conversely, if H_{01}^- is rejected then we proceed with calculating z_2. Finally, reject H_{02}^- if $z_2 \geq z_{1-\alpha}$ indicating $M_{X/Y}$ is within $-\delta$.

The outcomes from the non-parametric TOST of equivalence test and difference test are utilized together to determine a conclusion on the performance comparison between Computer X and Computer Y on the nth benchmark. In Table I and below, we list four conclusions: trivial difference, indeterminant, relevant difference, and equivalence [14], [15]:

- **Indeterminant**: fail to reject H_0^+ and (H_{01}^- or H_{02}^-). Indicating additional benchmark samples are needed for the evaluation.
- **Trivial difference**: reject H_0^+ and (H_{01}^- and H_{02}^-). Performance distributions were statistically significant but practically irrelevant.
- **Relevant difference**: reject H_0^+ but fail to reject (H_{01}^- or H_{02}^-). Performance difference on a benchmark that was outside the equivalence margin specified.
- **Equivalence**: fail to reject H_0^+ but reject (H_{01}^- and H_{02}^-). Performance scores come from the same distribution.

After all n benchmarks have one of the four relevance testing outcomes presented above, an optional Wilcoxon Signed-Rank Test for difference can be conducted depending on the results. In the case that all test's outcomes are equivalence, trivial difference, or a mixture of both, then the relevance testing is completed. Tests with all indeterminant benchmark results would likely require either additional samples or experimental design changes. For any other test outcome cases, an optional test can still be conducted with procedures detailed further in the text. We provide recommendations for publishing the results following the optional test procedure.

We can employ the optional Wilcoxon Signed-Rank Test for difference to determine an overall general performance comparison on the benchmark suite. Let $R_i = M_{X/Y}$ for $i = 1, 2, \ldots, n$ denote the median ratio on the nth benchmark. For benchmarks not concluded as relevant difference, assign $R_i = 0$; exclude it from the tests and reduce n, the number of benchmarks in the sample size, to $n = n - 1$ [12]. The test is excluded because the assumption of continuous variables under the null in a Wilcoxon Signed-Rank Test does not hold. The original family-wise error corrected α calculated prior to the evaluation remains unchanged to account for multiple hypotheses tests. Using the same procedures in the text above, the difference hypotheses for general performance comparison for a one-tailed Wilcoxon Signed-Rank Test are defined as:

- H_0^+: the benchmark suite performance score ratios of Computer X, Computer Y are symmetric around $\theta = 1$ (corresponding with no location shift from the benchmark suite)
- H_1^+: the benchmark suite performance score ratios of Computer X, Computer Y are symmetric around theta $\theta > 1$ (or $\theta < 1$)

Our framework provides outcomes that are practical and relevant to the study or performance comparison under consideration. We demonstrate the procedures with an evaluation of three RISC-V processors in the next section. Finally, we suggest writing a conclusion that includes the number of tests, outcomes (indeterminant, trivial difference, relevant difference, or equivalence), p-values, effect size in terms of location shift, α or confidence level, equivalence margin $[-\delta, \delta]$, and justification for the equivalence margin for performance evaluations.

IV. RISC-V EVALUATION

In this section, we evaluate three RISC-V softcore processors on an FPGA with SPARC and evaluate the analysis in comparison to HPT. The experimental configuration, captured performance metrics, and test assumptions are discussed prior to the evaluation.

A. EXPERIMENT SETUP

TABLE II. Softcore Processor FPGA Configurations

	Core				L1 Cache (KB)		
Processor	Type	Width (bits)	Pipeline Stages	Clock Rate	Inst	Data	DDR3 Size (GB)
Rocket	In-order	64	6	50	16	32	1
Ariane	In-order	64	5	50	16	32	1
Shakti	In-order	64	5	50	16	32	1

Our evaluation consists of three RISC-V softcore processors instantiated on a FPGA and a benchmark suite containing eight benchmarks to validate our methodology. The RISC-V processors Shakti [33], Ariane [34], and Rocket [35] are open-source softcore IP designs implemented in hardware descriptive languages for synthesis on an FPGA. Each processor has its own system-on-a-chip design integrated within the build that includes, but is not limited to: L1 and L2 cache, DDR3 memory controller, and universal asynchronous receiver-transmitter. We instantiated them in a Xilinx Virtex-7

TABLE III. Benchmark Descriptions

Benchmark	Description
AES	Encrypt, decrypt and compare 30 MiB of data
Bigint	Compute 23^{111121} and count base 10 digits
Dhrystone	Synthetic integer workload
Miniz	Compress, decompress and compare 8 MiB of data
Norx	Encrypt, decrypt and compare 30 MiB of data
Primes	Calculate largest prime number below 33333333
Qsort	Sort array containing 50 million items
SHA512	Calculate SHA-512 hash of 64 MiB of data

XC7VX485 T on a FPGA VC707 Evaluation Kit, using the Xilinx 2018.3 Vivado Design Suite. Both Ariane and Rocket had VC707 build configurations available, but Shakti required customization to port an existing FPGA build generation to the VC707. Shakti was customized by adding peripherals present on Rocket or Ariane, but absent from the Shakti build and did not affect the datapath of the processor. We implemented the softcores to operate at 50 MHz clock rate and the system configurations are listed in Table II.

We evaluated the processors with the benchmark suite, RV8, consisting of eight common benchmarks compiled for the RISC-V ISA and list their descriptions in Table III.

For each processor, Vivado synthesizes the system-on-a-chip design and generates an FPGA-specific bitstream, which it loads to the VC707. We compile the operating system, Linux version 5.3, as the main execution environment for the software and use a script to batch execute each benchmark 30 times to capture the number of clock cycles to complete it. The performance metric, number of clock cycles, is our response variable. In the experiment, we chose a sample size of 30 for each benchmark to examine suitability of its distribution for applying the Central Limit Theorem in our analysis.

B. PERFORMANCE METRIC MEASUREMENT

Within each benchmark, we inserted code to capture the clock cycles with inline assembly through a RISC-V specific pseudo-instruction, rdcycle [36]. The code executes at program start and program completion to calculate the number of clock cycles and then divided by the clock rate to derive the execution time L as follows:

$$L = \frac{Cycles_{End} - Cycles_{Start}}{Clock_{Rate}} \tag{12}$$

L is used to calculate the speedup ratio S between pairwise comparisons defined as:

$$S = \frac{L_A}{L_B} \tag{13}$$

We use the speed ratio to abstract out units of time and identify performance shifts that occur between processor comparisons.

C. SPARC FRAMEWORK SPECIFIC

To show correct application of our equivalence tests and conclusions, we define a wide $[1 - \delta, 1 + \delta]$ equivalence margin for analysis. Specifically, we use $\delta = 0.50$ and $[0.50, 1.50]$ as the primary equivalence margin for our softcore processor performance comparison. The bounds are purposefully large to illustrate the effect of equivalence tests and relevance outcomes on an analysis.

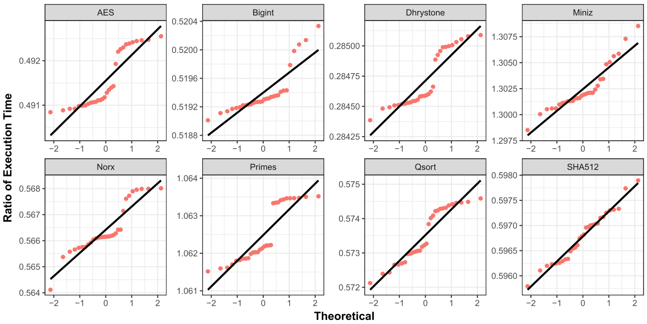

FIGURE 3. Rocket with Ariane quantile-quantile plots for each benchmark. Data points are the ratio, Rocket to Ariane, compared to a theoretical normal distribution line.

For our two primary RISC-V evaluations, there are a total of 34 hypotheses tests $2(8 + 8 + 1)$ conducted. There are two pairwise comparisons, Rocket to Ariane and Rocket to Shakti. A comparison between Ariane and Shakti was omitted here for space considerations. For the pairwise comparison Rocket to Ariane, we conduct 8 difference hypotheses tests for location shifts plus 8 equivalence hypotheses tests plus 1 for the overall analysis. The tests are repeated for the second pairwise comparison of Rocket to Shakti. Therefore, we set the overall evaluation error $\alpha = 0.05$ which translates to a $FWER \leq 0.82\,518$ using (1). We use the Bonferroni Correction method (2) to control the $FWER$ but still allow $(1 - \alpha/m)$ confidence intervals calculated. The error corrected $\alpha_{New} = 0.0\,014\,706$ which is compared to each benchmark test p_i to reject H_0 or fail to reject H_0.

D. RISC-V PERFORMANCE EVALUATIONS WITH SPARC FRAMEWORK

We present results to evaluate the efficacy of using the new methodology for performance comparisons beginning with Rocket to Ariane. In Fig. 3, we illustrate quantile-quantile plots for each benchmark with data points as paired-observation ratios against a theoretical normal distribution line. Visually, the plots for AES, Bigint, Norx, and Primes indicate non-normal distributions not suitable for parametric tests. The sharp curved data points around the normal line on AES and Norx are due to heavy tails and the large gap in data points on Bigint and Primes suggest bimodal distributions. In Fig. 4, benchmark densities are plotted and affirm multimodal distributions. We conducted Shapiro-Wilk Tests [37] and Kolmogorov-Smirnov Tests [38] for normality to affirm our visual analysis listed in Table VI and Table VII, respectively. We test each benchmark distribution against the Shapiro-Wilk and Kolmogorov-Smirnov H_0^+ that the distribution is normal; rejecting H_0^+ signifies the distribution is not normal. The Shapiro-Wilk Tests found Dhrystone and

TABLE IV. SPARC Framework Results for Difference and Equivalence At [0.50, 1.50] in Rocket to Ariane Comparison Tests

				H_0^+		H_{01}^-		H_{02}^-				
Benchmark	M_X (sec)	M_Y (sec)	$M_{X/Y}$	z	p	z_1	p_1	z_2	p_2	Rej H_0^+ at α?	Rej H_0^- at α?	Relevance
AES	129.516	263.677	0.491	4.772	1.825e-6	-4.792	0.992	4.772	9.127e-7	Yes	No	Rel diff
Bigint	203.471	391.805	0.519	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Dhrystone	75.660	265.857	0.285	4.772	1.825e-6	-4.792	0.992	4.772	9.127e-7	Yes	No	Rel diff
Miniz	505.566	388.321	1.302	-4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Norx	73.644	130.069	0.566	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Primes	257.019	242.012	1.062	-4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Qsort	129.927	242.012	0.573	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
SHA512	81.029	135.736	0.597	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff

TABLE V. SPARC Framework Results for Difference and Equivalence At [0.50, 1.50] Rocket to Shakti Comparison Tests

				H_0^+		H_{01}^-		H_{02}^-				
Benchmark	M_X (sec)	M_Y (sec)	$M_{X/Y}$	z	p	z_1	p_1	z_2	p_2	Rej H_0^+ at α?	Rej H_0^- at α?	Relevance
AES	129.516	299.904	0.432	4.772	1.825e-6	-4.792	0.992	4.772	9.127e-7	Yes	No	Rel diff
Bigint	203.471	170.041	1.197	-4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Dhrystone	75.660	129.989	0.582	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Miniz	505.566	473.909	1.067	-4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Norx	73.644	87.899	0.838	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Primes	257.019	448.945	0.573	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
Qsort	129.927	170.825	0.761	4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff
SHA512	81.029	77.208	1.049	-4.772	1.825e-6	4.772	9.127e-7	4.772	9.127e-7	Yes	Yes	Triv diff

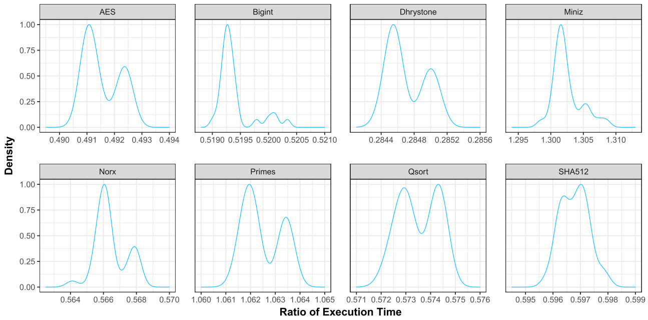

FIGURE 4. Rocket to Ariane density plots for each benchmark.

TABLE VI. Shapiro-Wilk Tests for Normality

	Rocket to Ariane			Rocket to Shakti		
Benchmark	W	p	Rej H_0^+?	W	p	Rej H_0^+?
AES	0.846	5.148e-4	Yes	0.905	1.124e-2	Yes
Bigint	0.818	1.418e-4	Yes	0.931	5.302e-2	No
Dhrystone	0.941	9.603e-2	No	0.267	3.76e-11	Yes
Miniz	0.861	1.056e-3	Yes	0.691	1.16e-6	Yes
Norx	0.870	1.701e-3	Yes	0.501	5.425e-9	Yes
Primes	0.791	4.486e-5	Yes	0.778	2.69e-5	Yes
Qsort	0.910	4.774e-2	Yes	0.732	4.637e-6	Yes
SHA512	0.961	0.335	No	0.938	8.163e-2	No

TABLE VII. Kolmogorov-Smirnov Tests for Normality

	Rocket to Ariane			Rocket to Shakti		
Benchmark	D	p	Rej H_0^+?	D	p	Rej H_0^+?
AES	0.221	0.091	Yes	0.195	0.178	No
Bigint	0.293	0.009	Yes	0.103	0.878	No
Dhrystone	0.252	0.037	Yes	0.431	1.4e-5	Yes
Miniz	0.228	0.074	Yes	0.318	0.003	Yes
Norx	0.232	0.067	Yes	0.453	3.83e-6	Yes
Primes	0.271	0.020	Yes	0.304	0.006	Yes
Qsort	0.180	0.255	No	0.249	0.040	Yes
SHA512	0.113	0.797	No	0.145	0.507	No

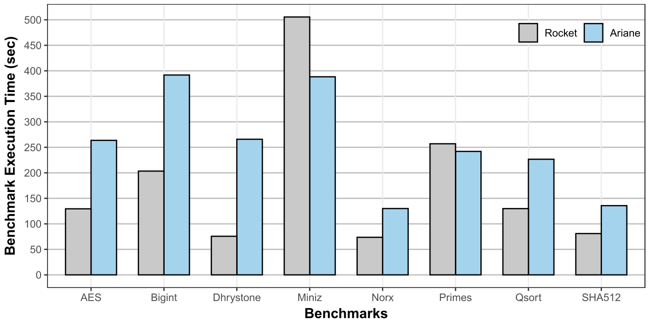

FIGURE 5. Median Rocket and median Ariane bar graph for each benchmark.

SHA512 are the only two normal distributions of the Rocket to Ariane evaluation. Whereas the Kolmogorov-Smirnov Tests found Qsort and SHA512 are non-normal. Therefore, we cannot rely on the Central Limit Theorem despite a larger sample size.

Proceeding with the new relevance framework, we conducted difference and equivalence hypotheses tests on the speedup ratio and the results are listed in Table IV. The median speedup ratio M_X/M_Y for Rocket (M_X) and Ariane (M_Y) shows a speedup to a faster time if greater than 1 and a slowdown if less than 1. The ideal ratio is 1 if the processors were equal in median execution time. Fig. 5 presents a bar graph plotting the median execution times listed in Table IV of

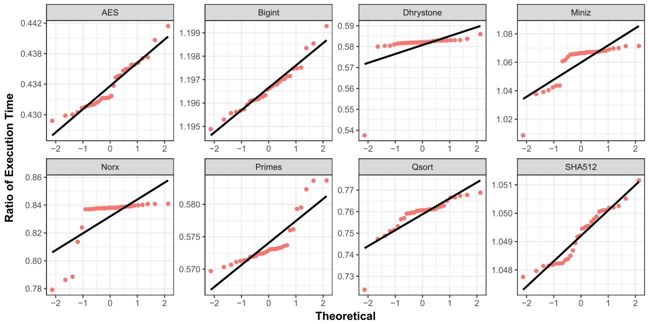

FIGURE 6. Rocket with Shakti quantile-quantile plots for each benchmark. Data points are the ratio, Rocket to Shakti, compared to a theoretical normal distribution line.

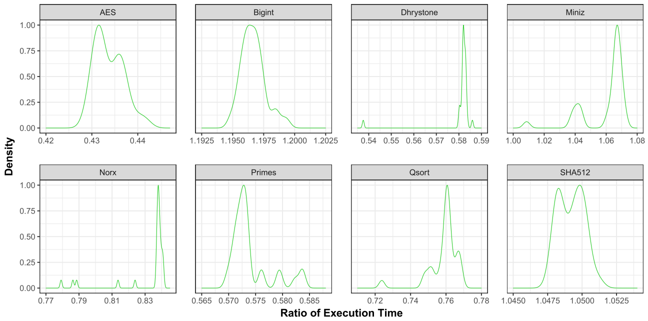

FIGURE 7. Rocket to Shakti density plots for each benchmark.

Rocket and Ariane within each benchmark. Each benchmark difference test rejected H_0^+, or that the performance score distributions are symmetric around $\theta = 1$. In other words, there was a distribution location shift of the median speedup ratio. Further, the tests of equivalence at $\delta = 0.50$ rejected H_0^- in all but the AES and Dhrystone benchmarks. The hypothesis test results, together with the four possible relevance choices from Section III-C, allow us to conclude that there is a relevant difference in median performance of the speedup ratio between the Rocket and Ariane RISC-V processors in 2 benchmarks, and a trivial difference in 6 benchmarks. We recommended previously in the text that the effect sizes should be listed, either in the evaluation conclusion, or as we listed in Table IV.

In the performance evaluation of Rocket to Shakti, we present quantile-quantile plots in Fig. 6. The plots show non-normal distributions in all benchmarks except for Bigint and SHA512. In contrast to quantile-quantile plots in Fig. 3, the distributions in Dhrystone, Norx, Miniz, and Qsort are highly skewed left and include heavy tails. The heavy tail in Dhrystone is caused by an outlier data point at −65 seconds. Similarly, outlier data points in Norx result in a heavy tail distribution and signify parametric tests could be affected if they were used. Further, density plots in Fig. 7 illustrate non-normal benchmark distributions. Results from Shapiro-Wilk Tests and Kolmogorov-Smirnov Tests for normality listed in Table VI and Table VII confirm that all benchmarks are non-normal except for AES, Bigint and SHA512.

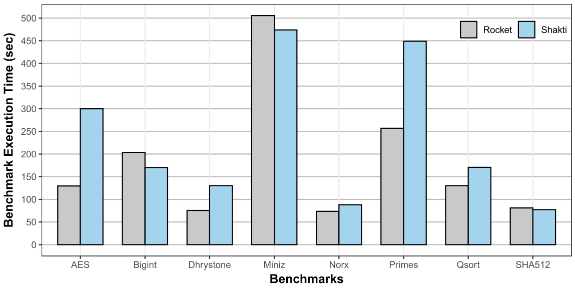

FIGURE 8. Median Rocket and median Shakti bar graph for each benchmark.

We could perhaps employ a different statistical analysis, examining the outlying data points to determine if they can be removed and then testing for normality again. This would require altering the α correction again, accounting for the additional hypotheses tests, and also adding justification for outlier data point removal. But if the process was successful and produced normal distributions, then parametric statistical tests could have been performed. We refrained from employing this technique because of the extensive time and experience required to distinguish between data points that are outliers versus data points that indicate a problem with the experimental design. Instead, SPARC was designed to test population medians with consideration that outlier data points are not removed.

Instead of removing any outlier data points though, we present results from the difference and equivalence hypotheses tests performed on the median speedup ratio of execution times for Rocket (M_X) and Shakti (M_Y) in Table V. The bar graph in Fig. 8 plots median execution times for Rocket and Shakti within each benchmark comparison. Again, the difference tests rejected H_0^+, indicating a distribution location shift of median speedup ratio. Alternatively, the tests of equivalence at $\delta = 0.50$ rejected H_0^- in all benchmarks except for AES. Here, we can conclude that there is a relevant difference in the median speedup ratio performance between Rocket and Shakti on 1 out of 8 benchmarks and a trivial difference in the other 7. We also conclude from the effect sizes in Table V as the speedup ratio of median performance, Rocket only had a relevant difference of faster median speedup ratio over Shakti in 1 of the 8 benchmarks.

In addition to the benchmark tests, we conducted a final Wilcoxon Signed-Rank Test for an overall relevant difference between Rocket to Ariane, and Rocket to Shakti, in Table VIII. Each test previously found trivial differences between Rocket and Ariane on 6 benchmarks, therefore we reduced the sample size by 6 since this test is only concerned with relevant differences. In the Rocket to Ariane general performance comparison, we fail to reject H_0^+, which indicates that there is not enough evidence to support a conclusion of a relevant difference in performance between Rocket and Ariane. A similar test was conducted for the Rocket to Shakti general performance comparison, with a similar outcome. In the previous tests, 1 resulted in a relevant difference between

TABLE VIII. SPARC General Performance Results for Both Comparisons

	$M_{X/Y}$	
Benchmark	Ariane	Shakti
AES	0.491	0.432
Bigint	Triv Diff	Triv Diff
Dhrystone	0.285	Triv Diff
Miniz	Triv Diff	Triv Diff
Norx	Triv Diff	Triv Diff
Primes	Triv Diff	Triv Diff
Qsort	Triv Diff	Triv Diff
SHA512	Triv Diff	Triv diff
Results H_0^+	p	Rej at $\alpha = 0.10$?
Ariane	0.50	No
Shakti	1.00	No

Rocket and Shakti for AES. Therefore, we remove any trivial difference tests from consideration as stated in Section III-C and reduced the sample size by 7. Out of 8 benchmarks, there was only a relevant difference in performance between Rocket and Shakti in 1 benchmark, and subsequently the test fails to reject H_0^+. The results are not unexpected. It is reasonable to assume that two processors with similar levels of performance would require more than 1 benchmark to reach a conclusion. The insight gained from the test of general performance between Rocket to Shakti is that we failed to reject H_0^+ of equal performance and a follow-on experiment with additional benchmarks would be required for further determination.

E. FRAMEWORK IN COMPARISON TO HPT

In order to compare the efficacy of SPARC to the HPT [1] framework, we consider some differences with respect to the benchmark statistics tests. As noted in Section IV-C, the observations are pairwise between processors and more appropriate for the Wilcoxon Signed-Rank Test used in SPARC. In HPT, the Wilcoxon Rank-Sum Test is usable on pairwise comparisons, but some information common to both populations is lost. Tests suitable for a difference of observations, likely remove variability shared between the two observations. In contrast to the Wilcoxon Rank-Sum Test, which compares two independent observations.

The test statistic is another key difference between SPARC and HPT. In SPARC, we specifically identify the speedup ratio between processors as the test statistic, whereas HPT designates an unspecified performance score. Again, the key disparity derives from using Wilcoxon Signed-Rank Test or Wilcoxon Rank-Sum Test and how each framework classifies response variables as paired or unpaired. We perform the HPT tests according to the procedures in [1], but use a paired Wilcoxon Rank-Sum Test with the speedup ratio in order to provide a standardized comparison between frameworks.

In Section III-A, family-wise error was discussed, in addition to possible risks to a study if α is not corrected. Determining if the data tested is within a family, and therefore affected by FWER, can be subjective. To illustrate the probability of making a Type I error, the FWER for SPARC and HPT is illustrated in Fig. 9. While our evaluation only used 8 benchmarks, without an α correction the FWER is 36.98% with HPT. But, the HPT framework lacks discussion on multiple hypothesis testing, nor does it discuss methods to correct α. This paper considers the omission as accidental and we purposefully discussed FWER in the SPARC procedures to remove ambiguity.

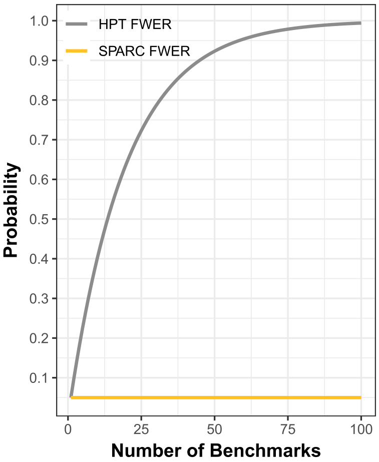

FIGURE 9. Family-Wise Error Rate of HPT and SPARC frameworks.

TABLE IX. HPT Framework Results for Wilcoxon Rank-Sum Tests in Both Comparisons

	Rocket to Ariane		Rocket to Shakti	
Benchmark	p	Rej H_0^+ at α?	p	Rej H_0^+ at α?
AES	1.863e-9	Yes	1.863e-9	Yes
Bigint	1.863e-9	Yes	1.863e-9	Yes
Dhrystone	1.863e-9	Yes	1.863e-9	Yes
Miniz	1.863e-9	Yes	1.863e-9	Yes
Norx	1.863e-9	Yes	1.863e-9	Yes
Primes	1.863e-9	Yes	1.863e-9	Yes
Qsort	1.863e-9	Yes	1.863e-9	Yes
SHA512	1.863e-9	Yes	1.863e-9	Yes

For HPT, two-tailed Wilcoxon Rank-Sum Tests were performed for each benchmark to determine whether Rocket or Ariane has a difference in median speedup ratio performance, listed in Table IX. The median speedup ratios are unchanged from Table V, therefore we only list the test results. For each test, H_0^+ was rejected at $\alpha = 0.10$ indicating a difference in benchmark speedup ratio performance between the two processors. Similarly, we performed the same tests for Rocket to Shakti with results listed in Table IX. For each benchmark, H_0^+ was rejected indicating a difference in speedup ratio performance between Rocket and Shakti.

Finally, a two-tailed Wilcoxon Signed-Rank Test is performed as HPT's general performance comparison across all benchmarks. The test was conducted twice, on Rocket to Ariane and Rocket to Shakti, listed in Table X. On both general

TABLE X. HPT General Performance Results for Both Comparisons

	$M_{X/Y}$	
Benchmark	Ariane	Shakti
AES	0.491	0.432
Bigint	0.519	1.197
Dhrystone	0.285	0.582
Miniz	1.302	1.067
Norx	0.566	0.838
Primes	1.062	0.573
Qsort	0.573	0.761
SHA512	0.597	1.049
Results H_0^+	p	Rej at $\alpha = 0.10$?
Ariane	0.014	Yes
Shakti	0.014	Yes

performance tests, HPT rejected H_0^+ which indicated a difference in performance.

In comparison to SPARC, the individual benchmark results by HPT illustrate the difference between each framework's concluding information. Specifically, in HPT each benchmark H_0^+ was rejected compared to the 6 trivial difference results for Rocket to Ariane and 7 trivial difference results for Rocket to Shakti in SPARC. As indicated by the follow-on equivalence tests, the difference in performance was within the $[0.50, 1.50]$ margin and subsequently each benchmark removed from the general performance comparison. The SPARC framework provides a method to establish an equivalency margin that the study has defined as similarly performing systems, compared to only detecting a difference in HPT. SPARC concluded there was only a trivial difference in performance in the majority of the benchmarks for both comparisons and resulted in a lack of evidence that supported a difference in performance.

Further, a difference will always be detected in HPT for evaluations similar to the example discussed in the introduction and illustrated in Fig. 3. The SPARC framework excels in conditions of similar performance or equivalence, and we are able to use the additional insights from SPARC to influence follow-on experimental design.

V. CONCLUSION

In this paper, the statistical framework SPARC is proposed for a scalable and distribution-free performance evaluation of computers. SPARC identifies superiority or equivalence with hypotheses tests for each benchmark that conditionally result in four relevance conclusions. Through the application of an error correction method in SPARC, error inflation is reduced in multiple benchmark scenarios. Our performance comparison of three RISC-V softcore processor's performance on an FPGA indicated the efficacy of SPARC in relation to a similar framework. The additional insight from relevance conclusions enhances the study results and refines discussion for further experimentation if required.

ACKNOWLEDGMENTS

The views expressed in this document are those of the authors and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense, or the United States Government. This document has been approved for public release; distribution unlimited, case #88ABW-2020-3839.

REFERENCES

[1] T. Chen, Y. Chen, Q. Guo, O. Temam, Y. Wu, and W. Hu, "Statistical performance comparisons of computers," in *Proc. - Int. Symp. High-Performance Comput. Architecture*, 2012, pp. 399–410.

[2] W. Zhang *et al.*, "Varcatcher: A framework for tackling performance variability of parallel workloads on multi-core," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1215–1228, Apr. 2017.

[3] T. Kalibera and R. Jones, "Rigorous benchmarking in reasonable time," in *Proc. Int. Symp. Memory Manage.*, 2013, pp. 63–74.

[4] A. Field, J. Miles, and Z. Field, *Discovering Statistics Using R*. Sage Publications, 2012.

[5] R. A. Fisher, *Statistical Methods, Experimental Design, and Scientific Inference*. London, U.K.: Oxford Univ. Press, 1990.

[6] F. J. Boster, "On making progress in communication science," *Hum. Commun. Res.*, vol. 28, no. 4, pp. 473–490, Oct. 2002.

[7] T. R. Levine, R. Weber, C. Hullett, H. S. Park, and L. L. M. Lindsey, "A critical assessment of null hypothesis significance testing in quantitative communication research," *Hum. Commun. Res.*, vol. 34, no. 2, pp. 171–187, Apr. 2008.

[8] J. Cohen, "Things I have learned (so far)," in *Proc. 98th Annu. Conv. Amer. Psychol. Assoc.*, Boston, MA, USA, 1992.

[9] J. Cohen, "The earth is round (p < .05)," *Amer. Psychologist*, vol. 49, no. 12, p. 997, 1994.

[10] D. H. Johnson, "The insignificance of statistical significance testing," *J. Wildlife Manage.*, pp. 763–772, 1999.

[11] P. E. Meehl, "Theoretical risks and tabular asterisks: Sir karl, sir ronald, and the slow progress of soft psychology," *J. Consulting Clin. Psychol.*, vol. 46, no. 4, p. 806, 1978.

[12] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. Wiley, 2013.

[13] P. Meehl, "14 What Social Scientists Don't Understand," *Metatheory in Social Science: Pluralisms and Subjectivities*, p. 315, 1986.

[14] A. Dinno, "Comment on 'the effect of same-sex marriage laws on different-sex marriage: Evidence from the Netherlands'," *Demography*, vol. 51, no. 6, pp. 2343–7, Dec. 2014.

[15] W. W. Tryon and C. Lewis, "An inferential confidence interval method of establishing statistical equivalence that corrects tryon's (2001) reduction factor," *Psychol. Methods*, vol. 13, no. 3, pp. 272–277, 2008.

[16] J. Bucek, K.-D. Lange, and J. v. Kistowski, "SPEC CPU2017: Next-generation compute benchmark," in *Proc. Companion ACM/SPEC Int. Conf. Performance Eng.*, New York, NY, USA, Apr. 2018, pp. 41–42.

[17] J. Worms and S. Touati, "Parametric and non-parametric statistics for program performance analysis and comparison," p. 70, 2017. [Online]. Available: https://hal.inria.fr/hal-01286112

[18] R. A. Fisher, *The Design of Experiments*. Edinburgh, U.K.: Oliver & Boyd, 1960.

[19] D. P. Reagle and H. Vinod, "Inference for negativist theory using numerically computed rejection regions," *Comput. Statist. Data Anal.*, vol. 42, no. 3, pp. 491–512, 2003.

[20] D. Szucs and J. Ioannidis, "When null hypothesis significance testing is unsuitable for research: A reassessment," *Front. Human Neurosci.*, vol. 11, p. 390, Aug. 2017.

[21] C. Bonferroni, "Teoria Statistica Delle Classi E Calcolo Delle Probabilita," *Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, vol. 8, pp. 3–62, 1936.

[22] N. J. Salkind, "Holm's sequential bonferroni procedure," in *Encyclopedia of Research Design*. SAGE Publications, Inc., 2018, pp. 1–8.

[23] S. Chakraborti, B. Hong, and M. A. Van De Wlel, "A note on sample size determination for a nonparametric test of location," *Technometrics*, vol. 48, no. 1, pp. 88–94, Feb. 2006.

[24] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Routledge, May 2013. [Online]. Available: https://www.taylorfrancis.com/books/9780203771587

[25] G. Shieh, S.-L. Jan, and R. H. Randles, "Power and sample size determinations for the wilcoxon signed-rank test," *J. Stat. Comput. Simul.*, vol. 77, no. 8, pp. 717–724, Aug. 2007.

[26] G. E. Noether, "Sample size determination for some common nonparametric tests," *J. Amer. Stat. Assoc.*, vol. 82, no. 398, pp. 645–647, Jun. 1987.

[27] E. Walker and A. S. Nowacki, "Understanding equivalence and noninferiority testing," *J. General Internal Med.*, vol. 26, no. 2, pp. 192–6, Feb. 2011.

[28] S. Wellek, *Testing Statistical Hypotheses of Equivalence*. Chapman & Hall/CRC Press, 2003.

[29] D. J. Schuirmann, "A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability," *J. Pharmacokinetics Biopharmaceutics*, vol. 15, no. 6, pp. 657–680, Dec. 1987.

[30] C. A. Mara and R. A. Cribbie, "Paired-samples tests of equivalence," *Commun. Statist. - Simul. Comput.*, vol. 41, no. 10, pp. 1928–1943, Nov. 2012.

[31] S. Feng, Q. Liang, R. D. Kinser, K. Newland, and R. Guilbaud, "Testing equivalence between two laboratories or two methods using paired-sample analysis and interval hypothesis testing," *Anal. Bioanalytical Chem.*, vol. 385, no. 5, pp. 975–981, 2006.

[32] C. A. Bellera, M. Julien, and J. A. Hanley, "Normal approximations to the distributions of the wilcoxon statistics: Accurate to what n? graphical insights," *J. Statist. Educ.*, vol. 18, no. 2, Jul. 2010.

[33] A. Menon, S. Murugan, C. Rebeiro, N. Gala, and K. Veezhinathan, "Shakti-t: A risc-v processor with light weight security extensions," in *Proc. Hardware Architectural Support Secur. Privacy*, New York, NY, USA, 2017.

[34] F. Zaruba and L. Benini, "The cost of application-class processing: Energy and performance analysis of a linux-ready 1.7-GHz 64-bit RISC-V core in 22-nm FDSOI technology," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 11, pp. 2629–2640, Nov. 2019.

[35] K. Asanovic *et al.*, "The rocket chip generator," EECS Dept. Univ. California, Berkeley, Tech. Rep. UCB/EECS-2016-17, 2016.

[36] A. Waterman and K. Asanovic, "The Risc-V Instruction Set Manual, Volume Ii: Privileged Architecture, v1. 12," 2019.

[37] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 1965.

[38] D. A. Darling, "The kolmogorov-smirnov, cramer-von mises tests," *The Ann. Math. Statist.*, vol. 28, no. 4, pp. 823–838, 1957.

JUSTIN C. TULLOS (Student Member, IEEE) received the B.S. degree in computer engineering from the University of Illinois at Chicago, Chicago, IL, USA, in 2011. He is currently working toward the M.S. degree in electrical engineering from the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, USA. His research interests include secure boot architecture, performance analysis of softcore processors, and FPGA hardware synthesis.

SCOTT R. GRAHAM (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2004. He is currently an Associate Professor of computer engineering with the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, USA. His main research interests include the security of cyber physical systems, looking at the interaction of computer architecture, networks, and security.

JEREMY D. JORDAN received the Ph.D. degree in operations research from the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, USA, in 2012. He led the Big Data Analytics and Operations Research Program for the Air Force Office of Scientific Research in Europe. He is currently an Assistant Professor of Statistics with the Air Force Institute of Technology. His research interests include network optimization, decision analysis, and uncertainty in networks.

PRANAV R. PATEL (Member, IEEE) received the Ph.D. degree in electrical engineering from the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, USA, in 2020. He is currently a Senior Electronics Engineer with the Air Force Research Laboratory, Wright-Patterson Air Force Base. His main research interests include RF and digital design, digital signal processing, data communication, and computer architecture.