Statistical Analysis of Multicore CPUs Operation in Scaled Voltage Conditions

Manolis Kaliorakis, Athanasios Chatzidimitriou, George Papadimitriou and Dimitris Gizopoulos

Abstract—Designers try to reduce the voltage margins of CPU chips to gain energy without sacrificing reliable operation. Statistical analysis methods are appealing to predict the safe operational margins at the system level as they do not induce extra area overheads and they can be applied during manufacturing or after the chips' release to the market. In this study, we present a comprehensive statistical analysis of the behavior of ARMv8 64-bit cores that are part of the enterprise 8-core X-Gene 2 micro-server family when they operate in scaled voltage conditions. Our prediction schemes that use real hardware counters as input are based on linear regression models with several feature selection techniques that aim to predict the safe voltage margins of any given workload when the cores operate in scaled conditions. Our findings show that our model is able to accurately predict the safe margins providing up to 20.28% power savings.

Index Terms—Hardware reliability, Energy-aware systems, Statistical methods

1 Introduction

PROCESS and dynamic variations in supply voltage and temperature can affect the correct operation of microprocessors. Thus, the designers choose to insert conservative guardbands in the operation voltage to protect the chips, despite the cost in terms of energy. Several hardware techniques associated with significant area and design overheads have been proposed to eliminate the conservative guardbands [1]. Recently, to avoid the extra overheads, several system-level approaches have been proposed to predict the safe operation limits of the processors. For example, the authors in [2] [3] propose an approach to predict the large voltage noise droops. In the same concept, [4] [5] propose a firmware implemented approach to predict the lowest safe voltage operation value based on the observation of corrected errors manifested on caches of an Intel Itanium processor. The energy gain in these studies comes from the variations of the V_{min} when the same workload runs on different cores (*core-to-core* variation) or different workloads run on the same core (*workload-to-workload* variation).

Towards this direction, a recent work [6] provides extensive characterization of the safe voltage operation margins of the ARMv8-based 8-core X-Gene 2 CPU chip; significant *core-to-core* and *workload-to-workload* variations running workloads from the SPEC CPU2006 suite [7] are reported. The most significant variations were observed between the most robust and the most sensitive cores of the chip (those with the lowest and the highest V_{min} on average). The robust core has lower V_{min} than the sensitive core for all workloads, with a difference ranging between 10mV and 35mV. Also, the variation of the V_{min} when different workloads are executed on the same core is remarkable for both cores (with range 885mV-915mV for the sensitive and 865mV-885mV for the robust core). Finally, between the two cores the highest observed V_{min} is 915mV and the lowest is 865mV that is translated into 12.83% and 22.10% power savings compared to the case of using the pessimistic nominal operation limit of 980mV.

Furthermore, [6] discusses the possibility to develop models to predict the V_{min} and the Severity (a metric that reflects the behavior of the cores below their safe V_{min} and before the occurrence of any catastrophic error) of the ARMv8 cores of the chip using as inputs the hardware performance counters. The models of [6] were able to efficiently predict only the Severity but not the V_{min} of the cores.

In this paper, we enrich our prediction models based on linear regression and accompanied with different features selection techniques to accurately predict *both* the V_{min} and the Severity. This is the first detailed study that compares several prediction schemes and identifies the correlation of the hardware counters with the V_{min} and the Severity using commercial ARMv8 cores. The models we deliver predict with excellent accuracy (only 0.51%) both the V_{min} and the Severity of any workload at scaled voltage conditions. Our findings can be used as inputs by a software implemented mitigation mechanism leading to power gains from 11.87% up to 20.28% depending on the aggressiveness of the prediction scheme.

2 X-Gene 2 Characterization Findings

In this section, we present the major findings presented in [6] that guided our decisions to target not only the V_{min} but also the Severity in the proposed prediction scheme. The chip consists of four pairs of ARMv8 cores. Each core has its own 32KB parity protected L1 instruction and data caches, while each pair of cores share an ECC protected unified 256KB L2 cache. All cores are in the same voltage domain, which can be changed with a granularity of 5mV beginning from the nominal 980mV. For this study, all the cores operate at 2.4 GHz, while we change only the voltage of the cores.

To identify the safe voltage operation margins of all cores we ran all the workloads from SPEC CPU2006 suite with different input datasets (40 different programs in total), starting from the nominal value (980mV) and reducing the supply voltage in 5mV steps. Each experiment was repeated 10 times. We classify the effects of our experiments in six categories: (i) normal operation (**NO**), (ii) Silent Data Corruption (**SDC**), (iii) Corrected Error (**CE**), (iv) detected but Uncorrected Error (**UE**), (v) Application Crash (**AC**), and (vi) System Crash (**SC**).

The first important finding is the three different regions of operation consistently observed for all the experiments according to the effects of each voltage step reduction: (1) **Safe region** (the experiments run to the end having a normal operation; the last point of this region defines the V_{min}), (2) **Unsafe region** (the experiments generated an abnormal behavior but not a system crash), (3) **Crash region** (the experiments lead to system crashes at least in some executions). The divergences of these regions between the cores are remarkable. For instance, the width of the Unsafe region for the

• *The authors are with the Computer Architecture Lab, University of Athens, Greece, Email: {manoliskal, achatz, georgepap, dgizop}@di.uoa.gr.*

most sensitive and the most robust core is 40mV-85mV and 65mV-105mV, respectively. Since in the Unsafe region the system is still alive, for certain machine and application combinations, operation at some high voltage values of this region can be affordable [6]: the CEs are always harmless, while there are systems that can also handle the UEs when they are detected if a rollback mechanism to previous checkpoint is implemented. Moreover, there are some applications that can tolerate the SDCs such as approximate computing algorithms, video streaming, image processing algorithms and security oriented applications (e.g. jammer attacks detectors). An aggressive predictor that exceeds the operation below the safe V_{min} and before any catastrophic for the system error occurs in the Unsafe region can increase the power gains.

The second important finding is that the silent data corruptions appear at higher voltage levels than corrected errors alone. Consequently, due to the occurrence of SDCs first, it is not possible to guide the voltage prediction based only on the manifested errors that was the case of the studies on Intel Itanium CPUs [4] [5].

The Severity metric as was defined in [6] can be used by an aggressive predictor scheme that targets operation below the safe V_{min} within the Unsafe region. According to [6], the Severity ranges from 0 to 19, where 0 denotes normal operation. The system that can tolerate SDCs, CEs and UEs can have Severity up to 7 units. In Table 1, we indicatively present the power savings of all cores for one workload (*bwaves*) in the case that a prediction mechanism targets the V_{min} value or follows a more aggressive policy targeting the Severity. In this example, a more aggressive prediction can lead from 1.10 up to 4.47 percentile units more power savings compared to the conservative one that targets the V_{min}.

TABLE 1
POWER SAVINGS TARGETING V_{MIN} OR SEVERITY FOR BWAVES

	Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7
V_{min}	14.72%	14.72%	14.72%	14.72%	18.45%	18.45%	16.59%	17.52%
Severity	18.01%	18.01%	15.82%	18.01%	22.92%	21.81%	21.01%	20.87%

3 STATISTICAL ANALYSIS METHODOLOGY

For our analysis, we used the linear regression model that can be easily calculated on hardware in contrast to the more complex non-linear models, as it is able to provide high accuracy with relatively small population of independent variables. The regression techniques calculate a function to predict a value of the dependent variable from a set of independent variables that are called *features*. Assuming a set of $x_1, x_2, x_3, \ldots, x_N$ independent variables and y the dependent variable, our analysis is based on the Ordinary Least Squares (OLS) model that calculates a set of weights β (one for each variable x) and an error term e, according to the formula:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + e_i \qquad (1)$$

In (1), y_i is the ith response value (i.e. V_{min} or Severity values), x_{ji} is the jth feature (e.g. hardware counter values) evaluated at the ith observation, and e_i is the ith statistical error. The goal of the model is to deliver the optimal values of the coefficients $\beta_1, \beta_2, \beta_3, \ldots, \beta_k$ so as to minimize the sum of the squares of the differences between the predicted and the observed responses of the test dataset.

We used as inputs for our models the output from the characterization phase (V_{min} and Severity values) and the performance counters from the entire execution of the workloads in nominal voltage conditions using the *perf* tool [8]. For the implementation of all our models, we used the *sklearn* python libraries [9].

The X-Gene 2 provides 101 hardware counters that report all the major events concerning the memory hierarchy (accesses and misses in all cache levels, TLB and page walks levels, prefetches, etc.), the core pipeline (flushes, mispredictions, etc.) and the system (bus accesses, etc.). Before training the linear regression model and in order to eliminate and select the most appropriate counters for it, we used three different *feature selection techniques*:

1) F_regression: This approach is based on F-distribution and computes the correlation of each independent feature X[:, i] with the dependent variable y, according to the following formula:

$$\rho_i = \frac{\big(X[:,i] - mean(X[:,i])\big) * (y - mean(y))}{std(X[:,i]) * std(y)} \qquad (2)$$

Then, for each feature, the *F-value* is computed according to eq. (3), where n is the population of y value. Finally, using the *F-value*, the associated *p-value* of each feature is determined [10].

$$F_i = \frac{{\rho_i}^2}{1 - {\rho_i}^2} * (n-1) \qquad (3)$$

Subsequently, the *F-values* and *p-values* of all features are sorted and finally, the best k features with the best values are selected and used as input from the linear regression model. Note that the lower the *p-value* and the higher the *F-value*, the more efficient the model becomes when we include this feature in the model.

2) Recursive Feature Elimination (*RFE*): The goal of *RFE* is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features, and weights are assigned to them according to their correlation coefficient. Then, the least important features are pruned from the current set of features and new weights are assigned to the remaining features. This procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

3) Polynomial Feature Transformation (*Polynomial*): To evaluate the correlation between two or more features, we implemented polynomial feature transformation that generates a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to a specified degree. For example, if an input sample of features is two-dimensional and consists of two features $[x_1, x_2]$, the new polynomial feature matrix of degree-2 will be $[1, x_1, x_2, x_1^2, x_1x_2, x_2^2]$. In this study, we implemented the polynomial feature transformation for all the cases targeting either the V_{min} or the Severity with a degree of correlation equal to 2. Our experiments revealed that a degree of correlation greater than 2 cannot provide better accuracy to our prediction models. After this transformation, we use either *f_regression* or *RFE* feature selection to select the best features for our linear regression models.

The evaluation of our models' accuracy targeting either the V_{min} or the Severity, was done by: (a) using the coefficient of determination (R^2) that assesses how well a model explains and predicts the future outcomes; also, it is indicative of the level of explained variability in the dataset. The larger the values of R^2, the better fit the model provides, while the best fit exists when R^2 is equal to 1, (b) using the Root Mean Square Error (*RMSE*) that represents the deviation between the predicted and the observed values (the smaller the *RMSE* the more accurate the model is), (c) comparing our

models with the baseline (naïve) model, which is the average of the target values (V_{min} or Severity) of the training dataset.

Depending on the target (the V_{min} or the Severity), we use different *samples* that correspond to information vectors of the values of the dependent and the independent variables that were used to train and test the models. All the samples used to predict the V_{min} consist of the performance counters from the entire execution of the workloads that were normalized per kilo committed architectural instructions, while the samples used to predict the Severity also include the information concerning the voltage values of each voltage reduction step from the characterization phase. To avoid biasing the results, all the collected values of each performance counter were normalized according to their minimum and maximum observed values, getting a final value with range [0, 1].

For all the experiments, we split the samples into 80% training and 20% test datasets, while we also cross validated our models with different combinations of train and test datasets coming from different workloads (8000K combinations for each experiment). The total population of samples that were used to predict the V_{min} and the Severity is 320 and 800 respectively.

4 Experimental Results

We implemented our linear regression models with the three different feature selection algorithms targeting both the V_{min} and the Severity in the eight cores of the X-Gene 2 running all the workloads from SPEC CPU2006 suite with all their inputs (40 programs in total). Due to space limitations, we present only the most representative and interesting cases of our analysis on the most robust (Core 4) and the most sensitive (Core 0) cores. For all the cases, we evaluate the accuracy when we select from one up to ten most important features to feed our linear model. Finally, for the population of features with the highest accuracy (the lowest *RMSE*), we present the final prediction model equations according to (1).

Predicting V_{min} of the Most Sensitive Core

In Fig. 1, we illustrate the accuracy results (in terms of *RMSE* measured in *mV*) of all our models when we target the V_{min} of the most sensitive core. The black dotted line presents the accuracy of the baseline (naïve) model which is the average values of the training dataset for prediction, while each set of bars represents the accuracy of the different models by using different population of selected features for the prediction. By using more than 4 features, all the methods are less accurate than the baseline model. Moreover, the linear regression model using only *RFE* is less accurate compared to the baseline model for all the cases. The linear regression that is accompanied by polynomial transformation with *f_regression* and up to 4 selected features gives better accuracy than the other methods for all the cases. The best accuracy (*RMSE* equal to 5.0108mV) was observed after using the polynomial transformation with *f_regression* selection and only 4 selected polynomial features. Table 2 presents the final prediction model according to equation (1). Moreover, the R^2 that was measured for this prediction model is high (close to 0.75) indicating a good fit of the model.

Our model finally delivers very high accuracy compared to the real values of our experiments (with only 5mV inaccuracy that corresponds to a single voltage step reduction supported by the machine). The potential power savings of using this scenario of prediction and adding a very small guardband of only 5mV (equal to our predicted inaccuracy) are 11.87% compared to the case of using the very pessimistic nominal voltage limit.

In general, the most important features that are finally used by the models described in Section 4 can be indicators of large voltage droops (i.e. BTB mispredictions, decode stalls, exceptions, branches) and timing errors in the pipeline of the X-Gene 2 CPUs (i.e. integer, FP operations). Therefore, they are intuitively well-correlated to the V_{min} values of the cores.

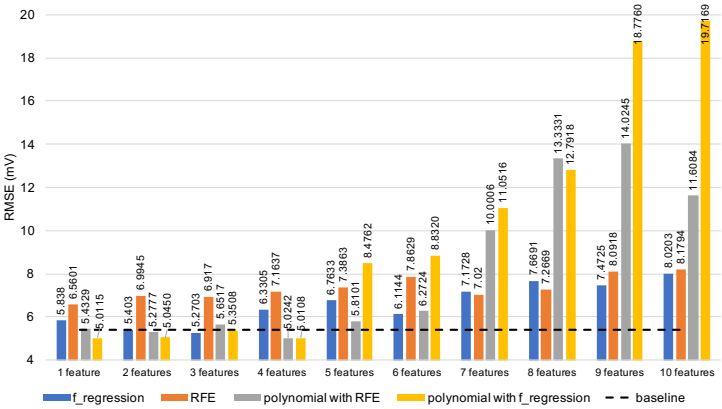

Fig. 1. Accuracy of predicting the V_{min} of the most sensitive core.

TABLE 2
V_{MIN} Prediction Model of the Most Sensitive Core

Hardware counters used by the model	Prediction Model (with 4 polynomial features)
L2 data prefetch request (*L2_pref*)	897.90 + (23.01∗ *L2_pref*) + (54.20 ∗ *BTB_miss* ∗ *FP*) – (7.15 ∗ *INT* ∗ *L2_pref*) – (58.84 ∗ *FP* ∗ *Indirect_br*)
BTB mispredictions (*BTB_miss*)	
Floating point operation (*FP*)	
Integer data processing (*INT*)	
Indirect braches (*Indirect_br*)	

Predicting V_{min} of the Most Robust Core

The results of the accuracy of the different prediction models that target the V_{min} of the most robust core are illustrated in Fig. 2. All the models with more than 4 or less than 3 features are less accurate than the baseline model. The best accuracy is observed for linear regression after applying polynomial transformation with *f_regression* and using only 3 polynomial features with *RMSE* equal to 5.0922mV (this model is presented in Table 3). The R^2 is again high (0.70), while the potential power savings of this model are 17.52% compared to the case of using the nominal value.

Predicting Severity of the Most Sensitive Core

Fig. 3 presents the results of the accuracy (in terms of *RMSE* that is measured in *Severity units* as was defined in [6]) when we target the Severity of the most sensitive core. Both the linear regression with a simple *f_regression* feature selection mechanism or the model in which we firstly apply a polynomial transformation and then we select the best features with *f_regression* significantly outperform the baseline model for all the cases. On the other hand, the simple *RFE* feature selection and the model with the polynomial transformation with *RFE* feature selection are less accurate than the baseline model for all our experiments. The best accuracy (with *RMSE* equal to 2.7223 Severity units) is observed for the model that uses *f_regression* feature selection with 3 features (this model is presented in Table 4). The R^2 for this model is very high (equal to 0.92) indicating the high efficiency of our model. The potential power savings of using this aggressive prediction scheme that targets the Severity including the measured error are 16.59% compared to the case of using the nominal voltage value for protection. These savings correspond to 39.76% more power savings instead of using the conservative prediction targeting the V_{min}.

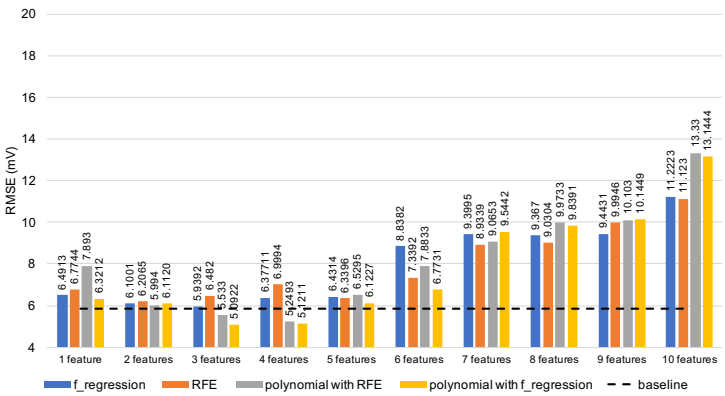

Fig. 2. Accuracy of predicting the V_{min} of the most robust core.

TABLE 3
V_{MIN} PREDICTION MODEL OF THE MOST ROBUST CORE

Hardware counters used by the model	Prediction Model (with 3 polynomial features)
L2 data prefetch request (*L2_pref*)	898.24 + (27.36 ∗ *L2_pref*) + (61.24 ∗ *BTB_miss* ∗ *FP*) – (8.96 ∗ *INT* ∗ *L2_pref*)
BTB mispredictions (*BTB_miss*)	
Floating point operation (*FP*)	
Integer data processing (*INT*)	

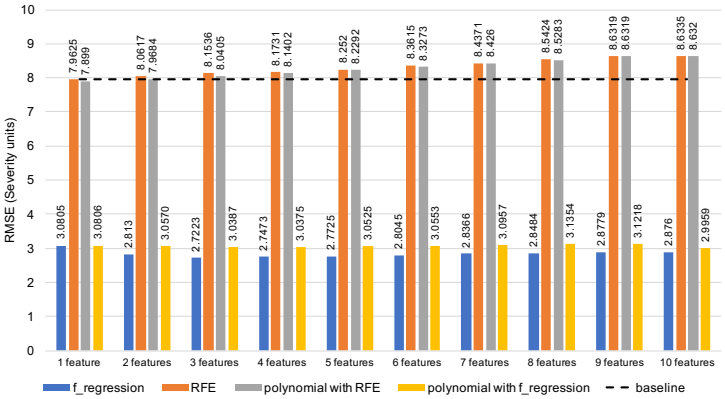

Fig. 3. Accuracy of predicting the Severity of the most sensitive core.

TABLE 4
SEVERITY PREDICTION MODEL OF THE MOST SENSITIVE CORE

Hardware counters used by the model	Prediction Model (with 3 features)
Voltage (*Volt*)	20.35 + (3.66 ∗ *Volt*) – (2.34 ∗ *L1D_tlb_write*) – (22.53 ∗ *dec_stalls*)
L1 data TLB write (*L1D_tlb_write*)	
Decode stalls (*dec_stalls*)	

Predicting Severity of the Most Robust Core

Finally, Fig. 4 illustrates the results of the accuracy of the models that predict the Severity values of the most robust core of the chip. All the models apart from the case of polynomial transformation with *RFE* feature selection outperform again the baseline prediction (the difference is about 5.2 Severity units). The best accuracy in terms of *RMSE* is equal to 2.5 Severity units and it is observed for the linear regression model with *RFE* feature selection using 6 features (this model is presented in Table 5). The R^2 for this model is again very high (equal to 0.91) indicating the high efficiency of the prediction model. Including the error, the potential power savings are 20.28% instead of using the nominal voltage value, while these gains correspond to 15.75% more power savings compared to the conservative case of using a scheme that targets the V_{min}.

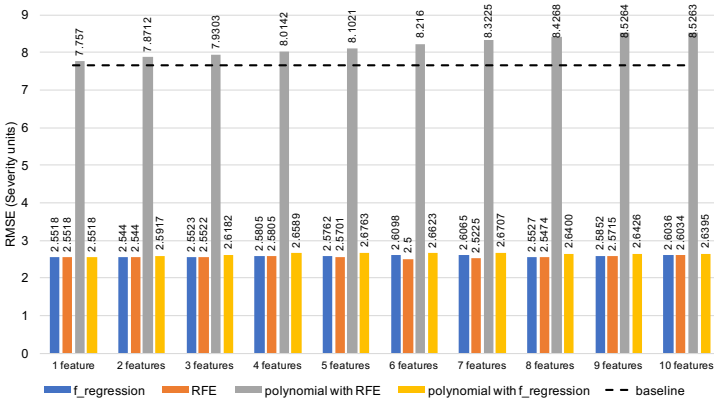

Fig. 4. Accuracy of predicting the Severity of the most robust core.

TABLE 5
SEVERITY PREDICTION MODEL OF THE MOST ROBUST CORE

Hardware counters used by the model	Prediction Model (with 6 features)
Voltage (*Volt*)	18.94 + (47.80 ∗ *Volt*) – (20.56 ∗ *excpt_taken*) + (7.09 ∗ *L1I_miss*) – (3.92 ∗ *cond_br*) – (29.50 ∗ *L1D_tlb_write*) – (22.11 ∗ *dec_stalls*)
Exceptions taken (*excpt_taken*)	
L1 instruction cache miss (*L1I_miss*)	
Conditional branches (*cond_br*)	
L1 data TLB write (*L1D_tlb_write*)	
Decode stalls (*dec_stalls*)	

5 CONCLUSION

We presented a comprehensive statistical analysis using linear regression with different feature selection methods to accurately predict the safe voltage operation margins of the cores of the enterprise X-Gene 2 micro-server ARMv8-based CPU running different workloads from the SPEC CPU2006 suite. The power savings targeting the V_{min} can be up to 17.52% compared to the case of using the nominal voltage value or up to 20.28% when we use a more aggressive scheme that targets the Severity instead of the V_{min}.

REFERENCES

[1] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. IEEE/ACM Int. Symp. on Microarchitecture*, 2003.

[2] V. J. Reddi, M. S. Gupta, G. H. Holloway, G.-Y. Wei, M. D. Smith, and D. M. Brooks, "Voltage emergency prediction: Using signatures to reduce operating margins," in *Proc. IEEE Int. Conf. on High-Performance Computer Architecture*, 2009.

[3] M. S. Gupta, V. J. Reddi, G. Holloway, G.-Y. Wai, and D. M. Brooks, "An event-guided approach to reducing voltage noise in processors," in *Proc. IEEE/ACM Design, Automation & Test in Europe Conf.*, 2009.

[4] A. Bacha, and R. Teodorescu, "Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors," in *Proc. IEEE/ACM International Symposium on Computer Architecture*, 2013.

[5] A. Bacha, and R. Teodorescu, "Using ECC feedback to guide voltage speculation in low-voltage processors," in *Proc. IEEE/ACM Int. Symp. on Microarchitecture*, 2014.

[6] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, P. Lawthers, and S. Das, "Harnessing voltage margins for energy efficiency in multicore CPUs," in *Proc. IEEE/ACM Int. Symp. on Microarchitecture*, 2017.

[7] J. L. Henning, "SPEC CPU2006 benchmark descriptions," in *ACM SIGARCH Comp. Architecture News*, vol. 34, pp. 1-17, Sept. 2006.

[8] Perf: Linux Profiling with Performance Counters. Retrieved 2017 from https://perf.wiki.kernel.org/index.php/Main_Page.

[9] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," in *Machine Learning Research*, vol. 12, pp. 2825-2830, Oct. 2011.

[10] J. R. Lackritz, "Exact p Values for F and t Tests", in *The American Statistician*, Vol. 38, No. 4, Nov. 1984, pp. 312-314.