Energy-Efficient Time Series Analysis using Transprecision Computing

Ivan Fernandez   Ricardo Quislant
Eladio Gutierrez   Oscar Plata

Department of Computer Architecture
University of Malaga (Spain)
ivanfv@uma.es

IEEE 32nd SBAC-PAD, September 2020

About me

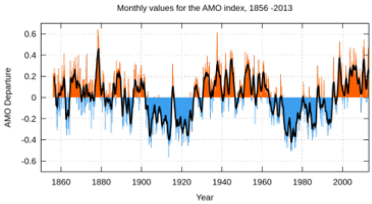
- 3^{rd}**-year PhD Student** at University of Malaga (Spain)
- **Advisors**: Oscar Plata and Eladio Gutierrez
- Research topic: **Acceleration of time series analysis**

Table of Contents

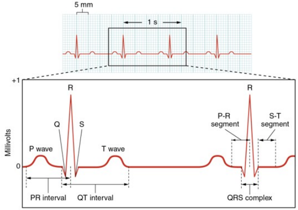
Table of Contents

Time Series Analysis

Time series analysis has a huge interest in many fields

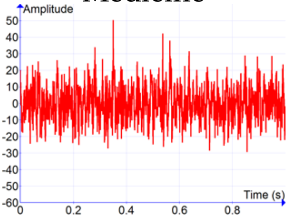


Climate change



Medicine

Economics

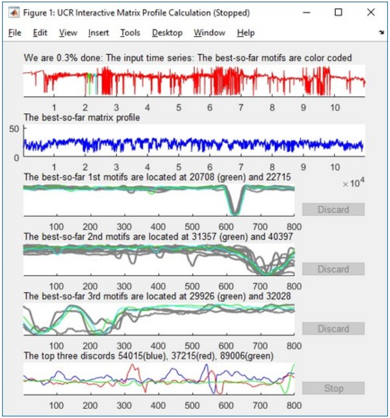


Signal processing

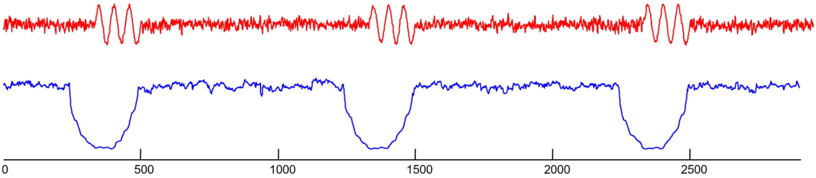
Matrix Profile

https://www.cs.ucr.edu/~eamonn/MatrixProfile.html

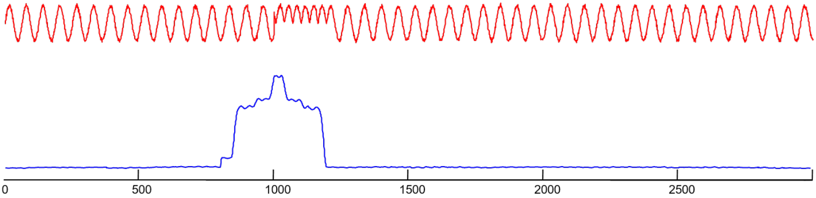
- Open source tool intended for **motif** (similarity) and **discord** (anomaly) discovery
- Implemented in several languages: C++, Python, CUDA, R, MATLAB
- No need for similarity threshold



Motifs and Discords



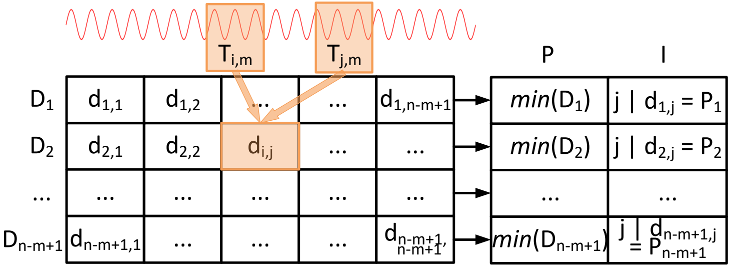
Synthetic **similarity** (motif) example



Synthetic **anomaly** (discord) example

Profile Calculation

The profile P and its index I are calculated on the fly using a matrix:



Those (Euclidean) distance values are calculated using the following:

$$d_{i,j} = \sqrt{2m\left(1 - \frac{Q_{i,j} - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)}$$

where $Q_{i,j}$ is the dot product between the subsequences of length m, and μ_i, μ_i, σ_i σ_j are their respective means and standard deviations

Motivation

- Matrix Profile is usually **based on double-precision floating point arithmetic**

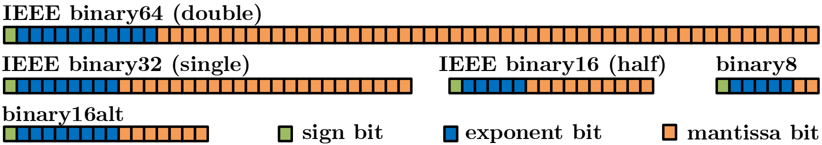
Analysing a time series of 130,000 elements using a subsequence length of 1,000 elements requires	2.4 Billion subtractions (-) 2.7 Billion multiplications (*) 2.9 Billion divisions (/) 2.8 Billion multiply-accumulations (FMA) + 2.8 Billion comparisons (<) **13.6 Billion operations !!!**

Observation

The number of operations increases **exponentially** with the time series length and the subsequence size, being a **major contributor ($\approx$ 50%) to energy consumption** in modern computing platforms

Motivation

- **Transprecision Computing** has emerged as a promising technique to boost energy efficiency and performance by reducing the number of bits in floating point operations
- There are already transprecision-enabled FPUs that can be implemented in RISC-V, FPGAs or ASICs (e.g., **FPNew***)



Available data types in FPNew. 32-bit, 16-bit and 8-bit operations can be performed in a SIMD approach

* https://github.com/pulp-platform/fpnew

Motivation

- **Problem**: Time series analysis, and particularly Matrix Profile, calculations involves a huge number of floating-point operations
- **Goal**: Enable energy-efficient and high performance time series analysis suitable for low power and embedded devices
- **Key Idea**: Use a transprecision computing approach, adjusting the precision to the needs of the application
- **Key Mechanism**: Study the accuracy of Matrix Profile algorithms using a transprecision emulation library. Provide guidelines to computer architects to design devices with required precision
- **Results**: Energy consumption reduced up to $3.3\times$ with respect to double precision approaches

Table of Contents

Transprecision Matrix Profile

- In this work, we develop two **transprecision benckmarks** of Matrix Profile, based on state-of-the-art implementations: SCRIMP and SCAMP

- We use **FlexFloat*** transprecision emulation library to enable arbitrary exponent and mantissa combinations in the calculations

- Both benchmarks are **freely available to community**, providing accuracy exploration for time series applications

SCRIMPFF SCAMPFF

* https://github.com/oprecomp/flexfloat

SCRIMPff

Our `SCRIMPff` benchmark is based on a parallel and vectorized implementation of SCRIMP. It is developed using a **configurable mixed precision** approach and its computation comprises the following steps:

1. Calculation of the dot product using *high* precision
2. Calculation of the Euclidean distance using *low* precision
3. Update the distance profile and index using *low* precision if needed

SCAMPff

`SCAMPff` follows a similar computation scheme than `SCRIMPff`, but **replaces the dot product with a mean-centered-sum-of-products** in order to reduce the floating-point rounding errors and the number of operations required:

$$df_i = \frac{T_{i+m-1} - T_{i-1}}{2}$$

$$dg_i = T_{i+m-1} - \mu_i + T_{i-1} - \mu_{i-1}$$

Additionaly, `SCAMPff` uses the Pearson correlation coefficient that can be computed in fewer operations and it is more robust than the Euclidean distance used by `SCRIMPff`:

$$D_{i,j} = \sqrt{2m(1 - P_{i,j})}$$

Table of Contents

Top-K Accuracy Metric

- Time series motifs and discords have been used for more than 15 years in the field of data mining for their capacity to find time series **subsequences with special significance**
- We provide defitions to these special subsequences and **propose a metric to measure the accuracy** in the detection of motifs and discords from two time series subsequences

Definitions I

Definition

The **motif** M_1 of a time series T is the unordered pair of subsequences $\{T_{i,m}, T_{j,m}\}$ which is the most similar among all possible pairs:

$$M_1 = \{T_{i,m}, T_{j,m}\} \iff dist(T_{i,m}, T_{j,m}) \leq dist(T_{u,m}, T_{v,m})$$

$$\forall i, j, u, v;\ i \neq j, u \neq v.$$

Definition

The **Top-K motifs** $M_{1,K}$ of a time series T is the set of the first K motifs:

$$M_{1,K} = \begin{cases} M_K \cup M_{1,K-1}, & K > 1 \\ M_1, & K = 1 \end{cases}$$

being M_K the motif (M_1) of the time series $T \setminus M_{1,K-1}, \forall K > 1$.

Definitions II

Definition

The **discord** D_1 of a time series T is the unordered pair of subsequences $\{T_{i,m}, T_{j,m}\}$ which is the most dissimilar among all possible pairs:

$$D_1 = \{T_{i,m}, T_{j,m}\} \iff dist(T_{i,m}, T_{j,m}) \geq dist(T_{u,m}, T_{v,m})$$

$$\forall i, j, u, v;\; i \neq j, u \neq v.$$

Definition

The **Top-K discords** $D_{1,K}$ of a time series T is the set of the first K discords:

$$D_{1,K} = \begin{cases} D_K \cup D_{1,K-1}, & K > 1 \\ D_1, & K = 1 \end{cases}$$

being D_K the discord (D_1) of the time series $T \setminus D_{1,K-1}, \forall K > 1$.

Table of Contents

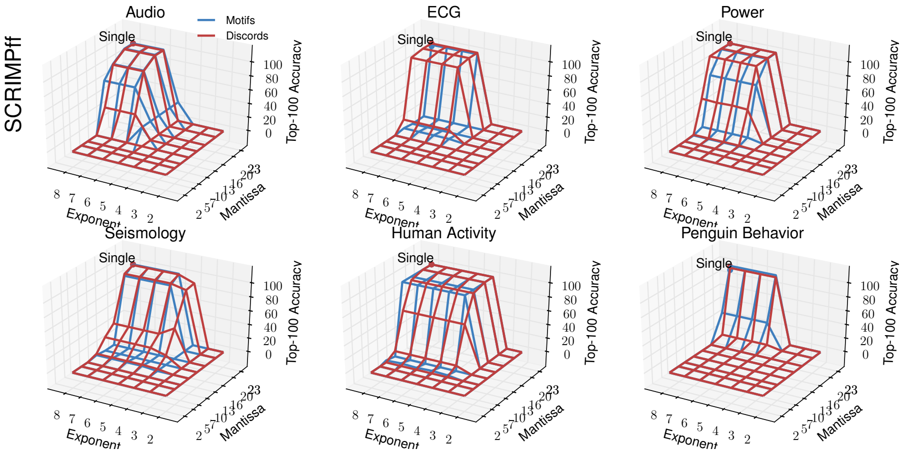
Evaluation Methodology

- We use an **Intel Xeon Phi** 7210 to perform our experiments
- We compute both SCRIMP and SCAMP using **double and single precision as reference**
- We use **FlexFloat library** to explore precisions lower than single
- We provide energy results using an available **transprecision FPU** and the FlexFloat operation breakdown statistics

Time series	**n**	**m**	**Max**	**Min**	**Scale**
Audio	20234	200 (2s)	6.69	-56.48	1
ECG	180000	500 (2s)	2.6	0.32	1
Power	180000	1325 (8h)	14.0	0	0.1
Seismology	180000	50 (2.5s)	6.96	-1.86	0.01
Human Activity	7997	120 (12s)	2.51	-1.9	1
Penguin Behavior	109842	800	0.52	-0.21	1

Analyzed time series in this work

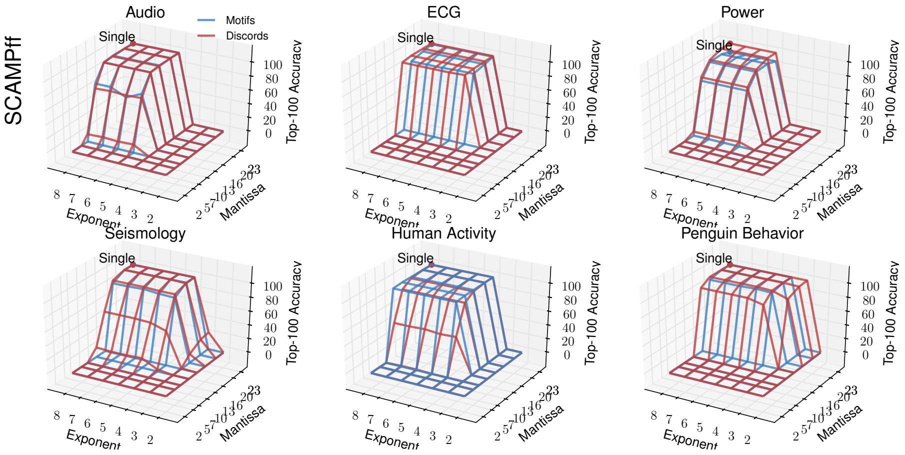
SCRIMP Accuracy



Observation

In most cases **single precision provides 100% accuracy** w.r.t. double.
Accuracy decreases dramatically after a given combination

SCAMP Accuracy



Observation

SCAMP is more robust and presents a better numeric stability than SCRIMP for all analyzed datasets

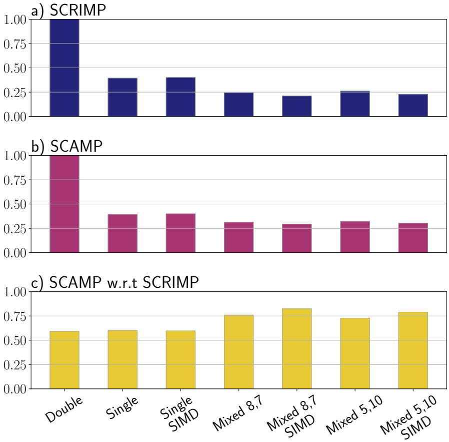
SCRIMP versus SCAMP Accuracy

T	High Exp/Man	Low Exp/Man	SCRIMPff Accuracy Mot/Disc	SCRIMPff Accuracy ±10 Mot/Disc	SCAMPff Accuracy Mot/Disc	SCAMPff Accuracy ±10 Mot/Disc
Audio	8/23	8/7	14/9	16/31	54/86	100/97
	”	5/10	38/0	99/0	95/99	100/100
	”	5/2	0/0	0/0	1/0	1/0
ECG	8/23	8/7	0/1	0/1	0/51	0/56
	”	5/10	10/57	10/60	25/99	30/100
	”	5/2	0/0	0/0	0/0	0/0
Power	8/23	8/7	47/31	67/95	39/25	78/99
	”	5/10	68/92	96/100	81/65	100/100
	”	5/2	0/0	0/0	0/0	0/0
Seis.	8/23	8/7	3/17	55/21	0/3	0/6
	”	5/10	7/68	86/70	12/40	15/45
	”	5/2	0/0	0/0	0/0	0/0
Hum.	8/23	8/7	72/24	80/63	91/84	99/92
	”	5/10	100/85	100/97	100/98	100/99
	”	5/2	0/3	0/4	0/0	0/1
Peng.	8/23	8/7	0/0	0/0	15/89	85/98
	”	5/10	0/0	0/0	81/99	100/99
	”	5/2	0/0	0/0	0/0	0/0

Observation

A mixed precision of 8/23 and 5/10 provides more than 98% accuracy for most datasets when using SCAMP

Energy



Observation

Using single instead of double precision, energy consumption of the FPU can be **reduced by 60%**. Using mixed precision, it can **be further reduced between 25% and 50%**, depending on the algorithm

Table of Contents

Conclusions and Future Work

- This work studies the **benefits from using a transprecision approach for time series analysis**
- We develop `SCRIMPff` and `SCAMPff` implementations that will **help the community to design energy-efficient time series analysis solutions**
- Our analysis reveals that the **energy consumption of a transprecision FPU is reduced up to 3.3**$\times$ compared with double precision when computing Matrix Profile algorithms
- An interesting future work would be the **evaluation of the transprecision analysis of time series in complete implementations** of RISC-V processors and FPGA-based devices

Thanks for your attention!

Energy-Efficient Time Series Analysis using Transprecision Computing

Ivan Fernandez Ricardo Quislant
Eladio Gutierrez Oscar Plata

Department of Computer Architecture
University of Malaga (Spain)
ivanfv@uma.es

IEEE 32nd SBAC-PAD, September 2020