Search for Extraterrestrial Intelligence:

GPU Accelerated TurboSETI

Luigi Cruz, Wael Farah, Richard Elkins
SETI Institute

BACKGROUND

- ✦ Radio telescopes are used to monitor the sky.
- ✦ They search for radio-based technosignatures.
- ✦ Each telescope antenna generates Tb/s of data.
- ✦ Recordings are scanned for drifting signals.
- ✦ This analysis is performed by TurboSETI.
- ✦ Taylor's De-Doppler algorithm is employed.
- ✦ Memory and compute-intensive process.

APPROACH

An effort to optimize the existing CPU code and create a GPU accelerated backend for TurboSETI.

CuPy

- ✦ Numpy is used by TurboSETI existing algorithms.
- ✦ Taylor De-Doppler algorithm rewritten with CuPy.
- ✦ CuPy offers GPU acceleration via NVIDIA CUDA.

Numba

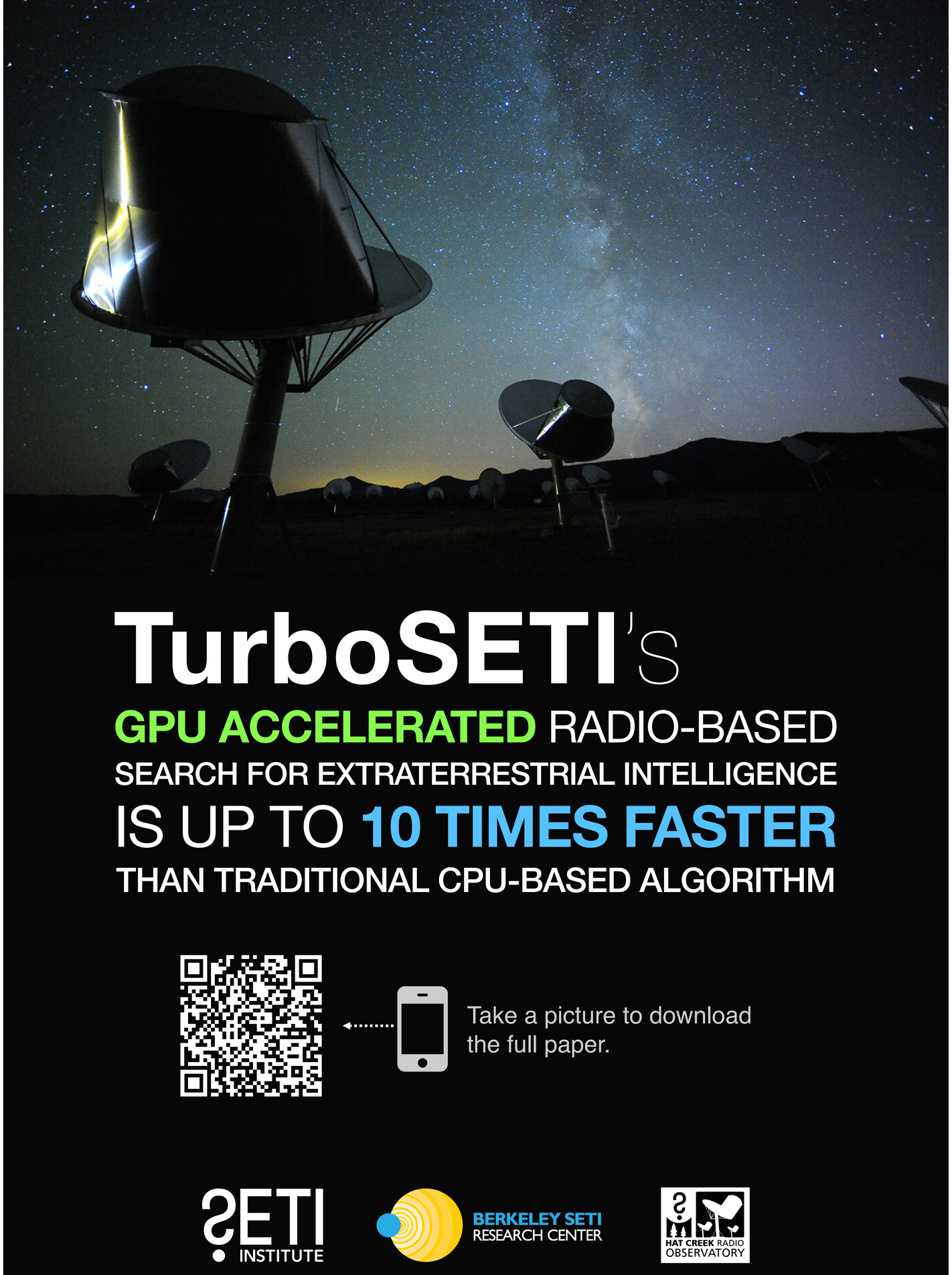
- ✦ Cython was used by some TurboSETI functions.
- ✦ Difficult to maintain due to different syntax.
- ✦ Functions rewritten in pure Python and Numba.
- ✦ Numba uses JIT compilation to generate ASM.
- ✦ Yielding C-like performance with Python code.

Single-Precision Floating-Point

- ✦ Original implementation handled input as 64-bit.
- ✦ Unnecessary due to hardware limitations.
- ✦ Mathematical operations with 64-bits take longer.
- ✦ Logic added to support 32-bits floating-point.

RESULTS

- ✦ New CPU and GPU implementations are faster.
- ✦ Benchmark shows faster CPU performance.
- ✦ Single-precision backend considerably faster.
- ✦ New GPU code is up to 9.3x faster than CPU.
- ✦ Optimizations mean more data processing.



TurboSETI's

GPU ACCELERATED RADIO-BASED
SEARCH FOR EXTRATERRESTRIAL INTELLIGENCE
IS UP TO **10 TIMES FASTER**
THAN TRADITIONAL CPU-BASED ALGORITHM

Take a picture to download the full paper.

SETI INSTITUTE
BERKELEY SETI RESEARCH CENTER
HAT CREEK RADIO OBSERVATORY

PRESENTER
Luigi Cruz
EMAIL
lfcruz@seti.org
TWITTER & GITHUB
@luigifcruz

Double-Precision (float64)

Impl.	Device	File A	File B	File C
Cython	CPU	0.44 min	25.26 min	23.06 min
Numba	CPU	0.36 min	20.67 min	22.44 min
CuPy	GPU	0.05 min	2.73 min	3.40 min

TABLE 1
Double precision processing time benchmark with Cython, Numba and CuPy implementation.

Single-Precision (float32)

Impl.	Device	File A	File B	File C
Numba	CPU	0.26 min	16.13 min	16.15 min
CuPy	GPU	0.03 min	1.52 min	2.14 min

TABLE 2
Single precision processing time benchmark with Numba and CuPy implementation.