# Native Support for x100 Xeon Phi Co-Processor

## Purpose

This document exists to discuss relevant experimental factors and data affecting the optimum completion time of a particular computer algorithm that renders advanced fractals. To supplement design efforts, this full factorial experiment documents multiple enhancements of algorithm productivity by means of an analysis of variance approach. Utilization of Python distributes the experimentation across multiple co-processors to expedite production of the results. **The results provide documented data illustrating the effects of increased performance and fidelity from enhanced efficiency within Mandelbulber2.**

## Introduction

Multi-core processors are the modern approach to rendering 3D fractals. *Parallel Ray Marching* unites the innovative essence for some 3D algorithms in modern computer science. Mandelbulber2, a collection of open source software, exists today exclusively for this purpose. The goal of this experiment is to record optimal software configurations using an x100 co-processor hardware system for rendering 3D Mandelbrot fractals.

## Experiment Description

The experiment considers three key factors for affecting the collective runtime of the algorithm under each experimental configuration. These factors include a patch for random number generation, a set of patches for the not-a-number function, and a patch for the shade and lights functions. Scripts and miscellaneous files have been prepared to compile and execute the algorithm. The algorithm executes eight times sequentially, once under each software configuration. Python scripts track the timing metrics for each experimental configuration. Each trial considers over 80 different examples that practice a majority of the code base within the algorithm.

### Null Hypothesis

H0: µ1 = µ2 = µ3 = µ4 = µ5 = µ6 = µ7 = µ8

### Alternative Hypothesis

H1: µi ≠ µj for at least one configuration pair

### Deliverables

|  |  |
| --- | --- |
| **ITEM** | **FORMAT** |
| README | Markdown |
| RESULTS | Markdown |
| REPORT | Latex |

#### README.md

This file describes the objective.

#### RESULTS.md

This file contains the experimental data.

#### REPORT

This folder contains the latex formatted report.

## Experiment Design

At this time, the design reflects a two-level full factorial experiment with k being three factors. The full combination of factors, with their respective patches applied to the baseline; compose test configurations compiled as experimental binaries.

### Baseline

* Mandelbulber v2.08-1 - Jul 3, 2016
* https://github.com/buddhi1980/mandelbulber2/commit/daec61022499faf1f75a8fa9d3bf5e861635c503

### Factor A

* Park-Miller Random Number Generator - Jul 31, 2016
* https://github.com/buddhi1980/mandelbulber2/commit/898fd8a1db7ad789ebe6381a66046cf99ed1456a.patch

### Factor B

* Eliminated Get Light Issue - Aug 1, 2016
* https://github.com/buddhi1980/mandelbulber2/commit/c494aee5e9a7bccc8773437e775c96a43ec37e16.patch

### Factor C

* Utilized GSL Finite for Not-a-Number - Sept 4, 2016
* https://github.com/buddhi1980/mandelbulber2/commit/68cacbfbafe623030978d21b2e3ade36c1df7f3c.patch
* https://github.com/buddhi1980/mandelbulber2/commit/56c6082db20fe2532d6b4b23c66b0149be58f81c.patch

## Linux Operating System

This experiment targets the Linux Operating System environment.

### Local Machine Configuration

#### local.sh

This shell script contains the definitions of many variables used throughout the experimental builds and throughout execution of experimental trial runs.

|  |  |  |
| --- | --- | --- |
| **VARIABLE** | **PATH** | **DESCRIPTION** |
| DOE | $PWD | The current directory of local.sh |
| SRC | $PWD/../../../ | The source code root directory |
| KNC\_LIB | $SRC/../KNC-Lib | x100 Co-Processor Library |
| MPSSDIR | $SRC/../mpss-3.7/k1om | x100 Co-Processor SDK |
| build\_logs | $SRC/../build\_logs | Experimental logs directory |
| BUILD | $SRC/../build | x64 build directory |
| BUILDMIC | $SRC/../build-mic | k1om build directory |
| BUILDNAN | $SRC/../NaNtest | Test build directory |

### Required Software Libraries

#### build\_all.sh

This shell script targets k1om architecture for the x100 co-processor. It compiles libraries from the KNC-Lib repository. The libraries used in this experiment include FreeType2 to render fonts, GNU Scientific Library for common mathematics, libjpeg for the application's JPEG support, libpng for Portable Network Graphics support, Perl Compatible Regular Expressions for pattern matching, and Qt5 as a development framework.

#### build.bot.sh

This experimental shell script uses the local Linux machine variables to execute a full build for both x64 binaries and Knights Corner binaries.

#### build.bot.log.sh

Log the Linux machine build process.

### Configurations

|  |  |  |  |
| --- | --- | --- | --- |
| Factors | NAN Patched | Light Patched | Random Patched |
| CONFIG000 | No | No | No |
| CONFIG001 | No | No | Yes |
| CONFIG010 | No | Yes | No |
| CONFIG011 | No | Yes | Yes |
| CONFIG100 | Yes | No | No |
| CONFIG101 | Yes | No | Yes |
| CONFIG110 | Yes | Yes | No |
| CONFIG111 | Yes | Yes | Yes |

### Compilation

#### white-space-v2.08-1.patch

Non-functional whitespace changes required for clean application of patches to baseline source revision.

#### build\_patches.sh

This script builds all possible combinations of patches and maintains separate binaries for each configuration.

#### build\_patches.log.sh

Logs the process of building all the patched binaries

## Analysis

### Floating Point Validation

#### NaNtester.sh

Test suite for k1om and x64 architectures

#### NaNtester.log.sh

Log the test suite process

### Amplifier Project

#### mandelbulber2-vTune

Investigate Algorithm Scalability

### Advisor Project

#### mandelbulber2-advisor

Investigate Algorithm Vectorization

## Data Collection

### Settings Tested

|  |
| --- |
| File Names |
| aboxmod1\_001 |
| aboxmod2\_001 |
| aboxvsicen1\_001 |
| aexion\_octopus\_001 |
| amoxmodkali\_001 |
| benesi\_t1\_pine\_tree\_001 |
| bristorbrot001 |
| hybrid002 |
| hybrid004 |
| hybrid007 |
| IFS\_002 |
| iq\_bulb\_001 |
| keyframe\_anim\_mandelbox\_boxes |
| Makin3D-Julia\_001 |
| mandelbox001 |
| mandelbox002 |
| mandelbulb002 |
| menger-mod1\_001 |
| msltoe\_julia\_bulb\_eiffie\_001 |
| primitives001 |
| quaternion\_001 |
| riemann\_sphere\_msltoe\_001 |
| KM/aexion01 |
| KM/benesi03 |
| KM/Bristorbrot |
| KM/bristorbrot01 |
| KM/constant factor 2.0 - mandelbox scale 2.0 |
| KM/equirectangular menger sponge |
| KM/fish eye |
| KM/folded mender sponge |
| KM/hybrid16 |
| KM/hybrid17 |
| KM/hybrid18\_2 |
| KM/hybrid19 |
| KM/hybrid20 |
| KM/hybrid22 - foldigIntPow v 2 |
| KM/hybrid42 |
| KM/hypercomplex 03 |
| KM/IFS 19 - maxiter |
| KM/IFS 20 |
| KM/IFS 21 |
| KM/IFS 25 |
| KM/IFS 29\_2 |
| KM/IFS 31 |
| KM/IFS 32 |
| KM/iter fog 002\_2 |
| KM/iter fog 2 |
| KM/mandelbox 22 |
| KM/mandelbox10 |
| KM/mandelbox11 - rotations |
| KM/mandelbox15 - rotations |
| KM/mandelbox17 |
| KM/mandelbox18 |
| KM/mandelbox19 |
| KM/mandelbox24 |
| KM/mandelbox25\_3 |
| KM/mandelbox27 |
| KM/mandelbox28 |
| KM/mandelbox29 |
| KM/mandelbox30 |
| KM/mandelbox31 |
| KM/mandelbox36 |
| KM/mandelbox49 |
| KM/mandelbox52\_2 |
| KM/mandelbox8 - maxiter |
| KM/mandelbulb powe 6 - circle |
| KM/mandelbulb power 2 - slice 4 |
| KM/mandelbulb power 2 - slice 5 |
| KM/mandelbulb power 4 - water |
| KM/mandelbulb power 8 - 4\_2 |
| KM/mandelbulb power 8 - 7 - volmetric fog |
| KM/menger sponge 004\_2 |
| KM/modified mandelbulb 001 |
| KM/orbitTraps 003 |
| KM/orbitTraps 004 |
| KM/orbitTraps 005 |
| KM/orbitTraps 006 |
| KM/primitive objects - water |
| KM/xenodreambuie2 |
| KM/xenodreambuie3 |
| SJ/hybrid 02 - rectangle hieroglyphs animation |
| SJ/hybrid 02 - rectangle hieroglyphs |

### Automation of Experiment

#### trial.sh

Linux Machine Trial Run

#### trial.log.sh

Log the Linux Machine Trial Run

#### trial.log.Loop.10.sh

Logs 10 replications the Linux Machine Trial Run

#### trial-run-host.py

Executes experimental binaries and records the timing data from numerous example settings.

### Automation of Render

#### render.bot.nodes.sh

#### render.bot.sh

#### host.py

Render sequential animation frames to collect timing data.

## Windows Operating System

### For Reference Only

Subsets of the experimental files enable partial Windows support for experimentation. These results contain many uncontrollable factors. Due to these nuisance factors, the results from Windows based machines exist as reference only.

#### build.batch.bat

This experimental batch file generates x64 binaries for Windows using MSBUILD.

#### build.batch.log.bat

Log the Windows machine build process.

#### trial.bat

Windows Machine Trial Run

#### trial.log.bat

Log the Windows Machine Trial Run.

## Conclusion

### Results

The entire experiment will replicate through multiple iterations to reinforce a sound statistical conclusion. The results, contained in a separate plain-text file, require further review.

### Release

* Mandelbulber v2.09 – Oct 4, 2016
* https://github.com/buddhi1980/mandelbulber2/commit/013208e3ee969c498a01b4160f834ad20f2a2f13
* Easily scales past 240 threads