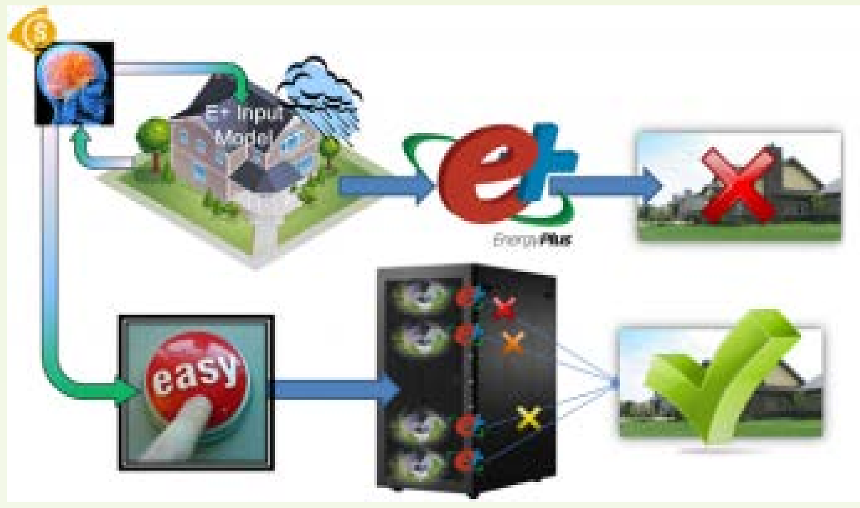
In-situ Statistical Analysis of Autotune Simulation Data using Graphical Processing Units

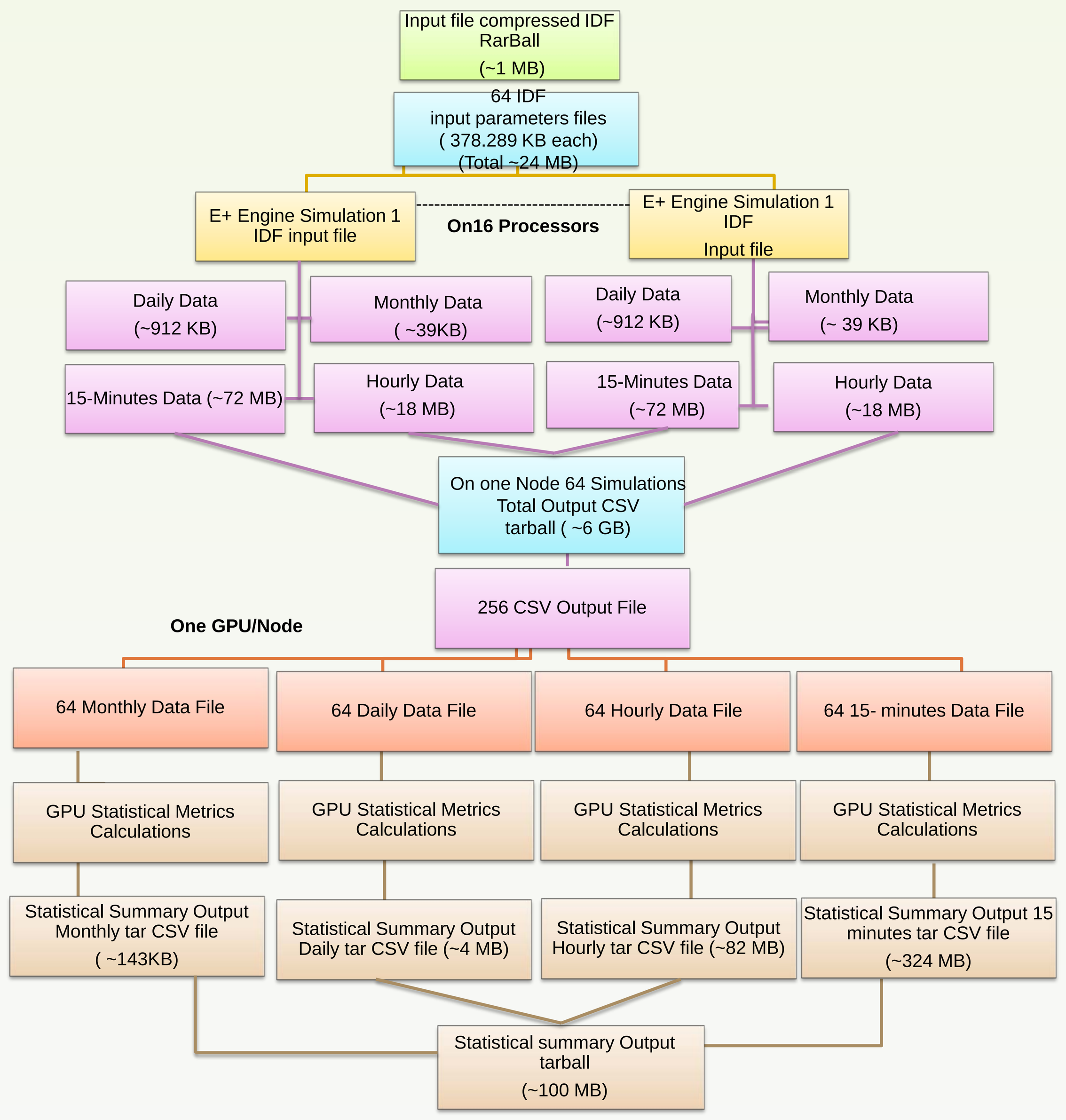
Niloo Ranjan[1], Jibonananda Sanyal[2], Joshua New[2]
1 nranjan@pstcc.edu, Pellissippi State Community College, Knoxville, TN 37932
2 {sanyalj, newjr} @ornl.gov, Oak Ridge National Laboratory, Oak Ridge, TN 37830

Autotune EnergyPlus Simulation

The aim of the Autotune project is to speed up the automated calibration of building energy models to match measured utility or sensor data. The workflow of this project takes input parameters and runs multiple EnergyPlus simulations on the Titan supercomputer. These simulations running parallely on multiple nodes, having 16 processors and a graphics processing unit (GPU) on each node, produces a 5.7 GB output file comprising 256 files from 64 simulations. A total of 270TB+ of data has been produced. We used CUDA along with C/MPI to calculate statistical metrics such as sum, mean, variance, and standard deviation leveraging GPU acceleration. The workflow developed in this project produces statistical summaries of the data which reduces the output data size that needs to be stored from 5.7 GB to approximately 100 MB. These statistical capabilities are expected to be used for sensitivity analysis of EnergyPlus simulations.



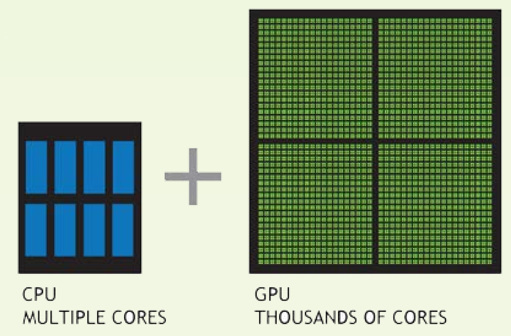
Autotune EnergyPlus building energy modeling to match measured data .



Simulation Workflow with In-situ Statistical analysis of output simulation data.
The size of the output data reduces from ~6 GB to ~100 MB with the Statistical summary of all one data type. This workflow shows the simulation process running on one node. Currently, the simulations are running on 8,000 nodes and has potential to run on 16,000 nodes.

Graphical Processing Units

Graphical Processing units are (GPUs) are combinations of thousands of smaller but efficient many-core co-processors that have capability to accelerate high performance computing. General-Purpose Graphical Processing Units (GPGPU) have general-purpose parallel processors that support many programming interfaces. GPGPU computing can accelerate general-purpose scientific and engineering applications. A CPU often executes the serial portion of the program while the GPU calculates the parallel part of the application using the principle of same instruction on multiple data (SIMD). GPU computing is best for data-parallel computation, such as operations on matrices and vectors, where elements of the data set are independent of each other and can be computed simultaneously.



CPUs and GPUs have different architectures with GPUs containing many cores compared to CPUs. The combination of CPUs and GPUs can greatly accelerate an algorithm process. (images courtesy of NVIDIA).

GPU Accelerated Libraries

	Thrust	CUDPP	cuBLAS	MAGMA
Description	C++ like Standard Template library	CUDA Data Parallel Primitives Library	CUDA Basic Linear Algebra Subroutines	Matrix Algebra on GPU and Multicore architectures
Availability / Latest Version / date	Included in nVIDIA CUDAtoolkit / Thrust v1.7 / July 2013	Open source (New BSD License)/ CUDPP 2.0 / August 2011	Included in nVIDIA CUDAtoolkit / CUDA 5.0 / October 2012	Open Source(UT)/ MAGMA 1.4 Beta2 / June 2013
Requirement / Operating System (OS) support / Programming Language support	nVIDIA CUDA 4.0 or better / Linux,Windows, Mac OS X / CUDA C/C++, OpenMP, TBB (Threading Building Blocks, Intel)	nVIDIA CUDA 3.0 or better/ Linux,Windows, Mac OS X / CUDA, C/C++	nVIDIA CUDA 4.0 or better /Linux,Windows, Mac OS X / CUDA, C/C++	nVIDIA CUDA/ Linux,Windows, Mac OSX/ CUDA, OpenCl, Intel Xeon Phi
Main Subroutines	Scan, search, search by key, count, merge, reorder, prefix-sum, set, sort, transform	Sort, stream compaction, scan, prefix-sum, parallel reduction	min, max, sum, copy, dot product, norm(Euclidean norm of the vector), scal, swap, multiplication, rank	80+ hybrid algorithms (total of 320+ routines), linear and least squares solvers, and all of basic linear algebra subroutines

Method

This project used CUDA and the C programming language to write a program that generates a statistical summary of simulation output. The statistical summary includes sum, mean, and Standard deviation of the 64 *.csv output files each for Monthly, Daily, Hourly, and 15-minute resolution simulation data. The EnergyPlus simulation engine runs and stores the *.csv files in RAMDisk. The program created for this project then reads one file at a time into a matrix. The data is then transferred to the GPU in order to calculate running sum, mean, number of elements in the data set, and sum of squares of difference between the data and the mean. After processing all 64 files of a type, it calculates the variance by dividing the sum of squares by the number of elements. The standard deviation is computed by taking the square root of the variance. The program then generates output *.csv files with statistical summaries and stores compressed files in the output directory provided by the workflow.

Results

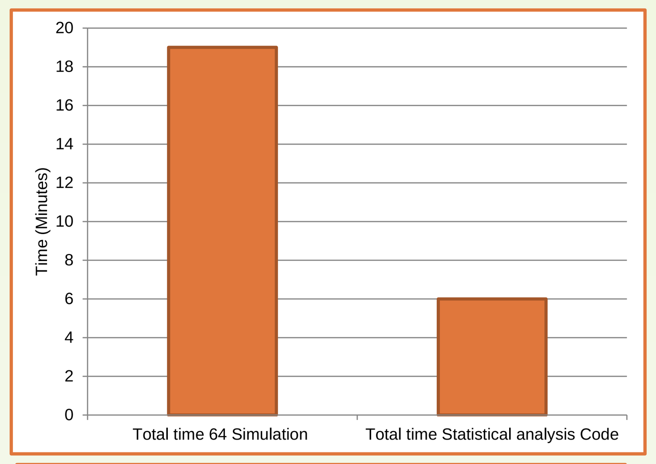


Figure 1. Time Simulation Run With GPU statistical analysis

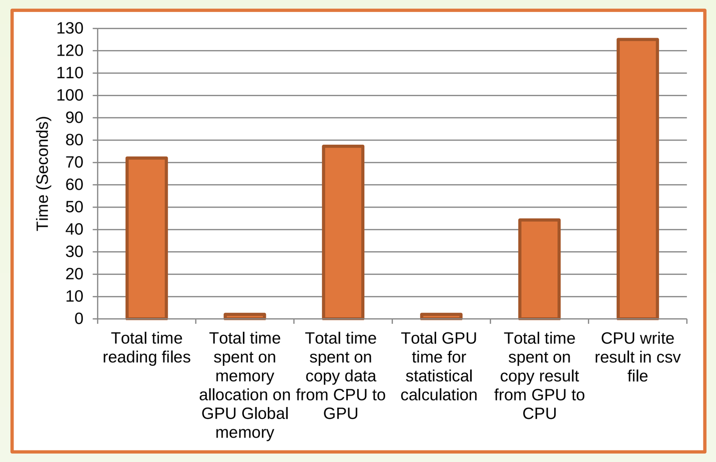


Figure 2. GPU Statistical Analysis Time Division

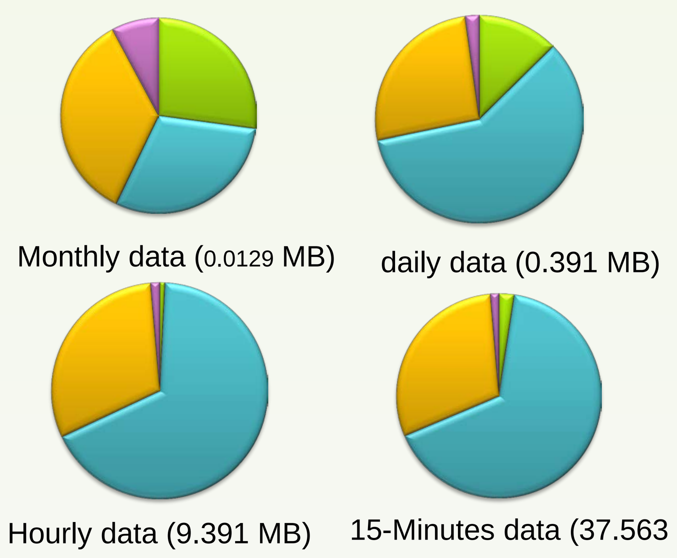
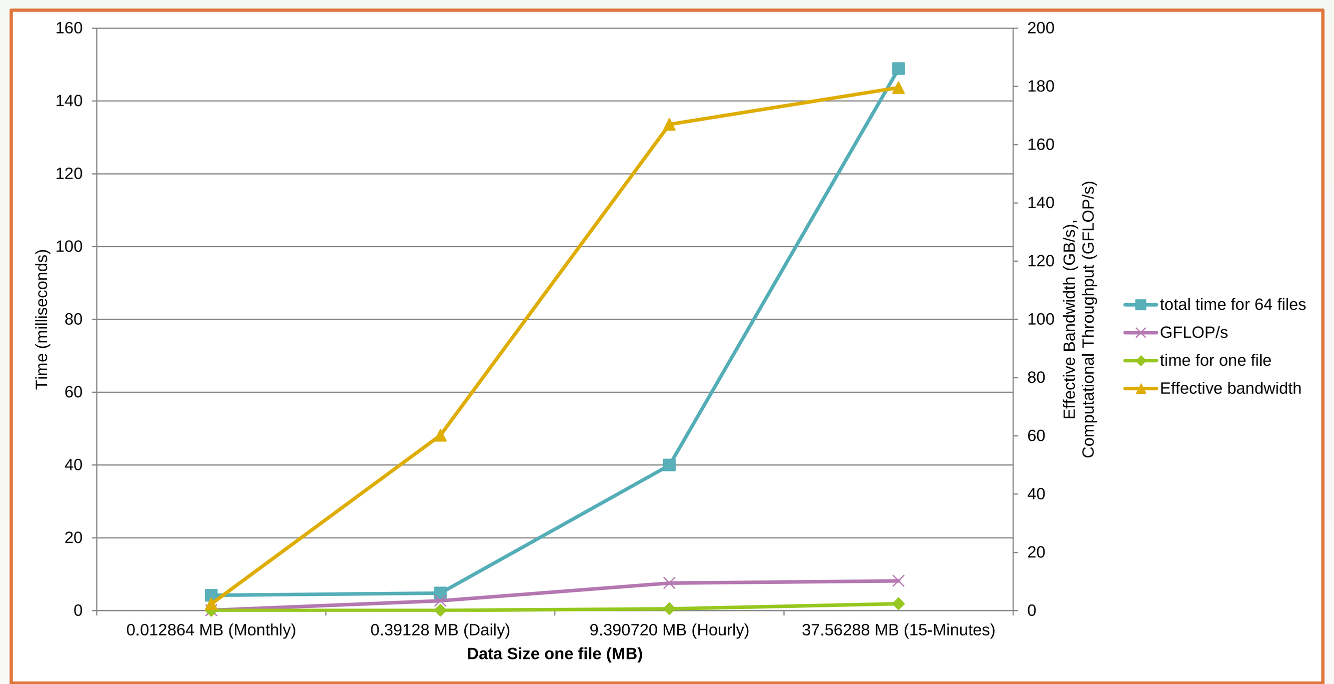


Figure 3. GPU time division for four data types



Theoretical Maximum Bandwidth = 250 GB/s
Peak Computational Throughput (double precision) = 1.31 TFlop/s

Figure 4. GPU Performance Metrics for Statistical Summary Generation

Acknowledgements: This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program.

OAK RIDGE National Laboratory
U.S. DEPARTMENT OF ENERGY Office of Science
ORAU
OAK RIDGE INSTITUTE FOR SCIENCE AND EDUCATION Managed by ORAU for DOE
PELLISSIPPI STATE COMMUNITY COLLEGE