Many-Core Computing for Space-based Stereoscopic Imaging

Abstract—The potential benefits of using parallel computing in real-time visual-based satellite proximity operations missions are investigated. Improvements in performance and relative navigation solutions over single thread systems can be achieved through multi- and many-core computing. Stochastic relative orbit determination methods benefit from the higher measurement frequencies, allowing them to more accurately determine the associated statistical properties of the relative orbital elements. More accurate orbit determination can lead to reduced fuel consumption and extended mission capabilities and duration. Inherent to the process of stereoscopic image processing is the difficulty of loading, managing, parsing, and evaluating large amounts of data efficiently, which may result in delays or highly time consuming processes for single (or few) processor systems or platforms. In this research we utilize the Single-Chip Cloud Computer (SCC), a fully programmable 48-core experimental processor, created by Intel Labs as a platform for many-core software research, provided with a high-speed on-chip network for sharing information along with advanced power management technologies and support for message-passing. The results from utilizing the SCC platform for the stereoscopic image processing application are presented in the form of Performance, Power, Energy, and Energy-Delay-Product (EDP) metrics. Also, a comparison between the SCC results and those obtained from executing the same application on a commercial PC are presented, showing the potential benefits of utilizing the SCC in particular, and any many-core platforms in general for real-time processing of visual-based satellite proximity operations missions.

TABLE OF CONTENTS
1. INTRODUCTION ...........................................1
2. INTEL’S SINGLE-CHIP CLOUD COMPUTER ..........2
3. ALGORITHM OVERVIEW AND IMPLEMENTATION  ........3
4. RESULTS ....................................................4
5. CONCLUSIONS AND FUTURE WORK ...............5
REFERENCES ...............................................6
BIOGRAPHY ...............................................7

1. INTRODUCTION

Research in close proximity operations has been a growing interest of the military and scientific communities. Recently, with the growing need for Space Situational Awareness (SSA), the scope of proximity operations has expanded from cooperative operations to the navigation about both cooperative and uncooperative resident space objects (URSOS). As the name suggests, URSOS are defined as objects which do not communicate with the inspector spacecraft, and range from space debris to adversarial spacecraft. In any case where an inspector spacecraft needs to navigate with respect to a URSO, one or more relative ranging sensors must be used.

The ranging sensor under consideration is a visual-spectrum stereoscopic imager consisting of two low-cost, commercial-off-the-shelf (COTS) cameras and an associated processing platform. Space-based vision-based proximity operations possess the inherent advantage of offering an environment in which a single foreground object is in the field-of-view of the stereoscopic imager. The cameras are aimed in the same direction, which allows the single foreground object to be isolated without the need for a computationally expensive object detection algorithm. The performance of the stereoscopic imager can be improved with higher resolution cameras. In order to maintain a quick relative navigation solution, the speed of the associated processing platform must get faster as the resolution of the cameras increase.
Visual-based proximity operations gather the reflected light off the URSO in order to determine the relative position to the URSO. The imaging process is passive, meaning the URSO cannot directly detect it is being observed. When imaging potentially adversarial URSO’s, it is important to consider the passivity of the ranging sensor. RADAR and LIDAR have been popular choices for ranging sensors. These sensors are active, meaning they emit electromagnetic radiation and detect the radiation when it is scattered by the URSO, making RADAR and LIDAR detectable by the URSO. RADAR and LIDAR also need relatively large amounts of power compared to the stereoscopic imager to range the URSO, since they generate the electromagnetic radiation to be gathered instead of simply gathering the natural radiation reflected by the URSO.

The relative orbit determination process implemented by the inspector spacecraft implements stochastic methods, such as Kalman filtering, to improve the accuracy of the relative navigation solution compared to the raw relative position measured by the stereoscopic imager. Two main parameters affect the accuracy of the relative navigation solution: measurement noise and measurement frequency. If the relative orbit determination filter is given more accurate measurements, it will possess a more accurate relative navigation solution, as intuition suggests. Faster processing capabilities, with the associated higher resolution cameras, will benefit the relative navigation solution with the increased measurement accuracy. A higher measurement rate, which is often limited by processing speed, will offer a more accurate relative navigation solution since more information is being passed to the filter, allowing it to more accurately capture the mean and covariance of the relative navigation solution. Increased processing capabilities will benefit the relative orbit determination with the ability to more accurately and quickly measure the relative position of the inspector with respect to the URSO.

The rest of the paper is organized as follows. Section II presents an overview of the Intel Single chip cloud computer. The stereoscopic imaging algorithm and an overview of its parallel implementation is outlined in Section III. Section IV presents and comments on the experimental results. The paper concludes in Section VI and gives an outlook for future work.

2. Intel’s Single-Chip Cloud Computer

In the commercial field, it is common nowadays to have multi-core chips housing 2, 4, 6 or even more cores; while the research community makes use of experimental many-core architectures. There is no hard definition for these many-core architectures. Basically, it is a microprocessor with tens, hundreds, or even thousands of processor cores.

Intel Labs’ concept vehicle, the Single-Chip Cloud Computer (SCC), is a 48-core chip prototype created to study many-core architectures, and the techniques used to program them. It is introduced as an ideal research platform to help accelerate many-core software research [1].

A general overview of the SCC chip architecture is presented in Figure 1. It consists of 48 second-generation Pentium™ class IA-32 cores on a 6x4 2D-mesh network of tiled core clusters with high-speed I/Os on the periphery [2]. Each core has 16KB L1 data and 16KB L1 instruction caches as well as a 256KB unified L2 cache. Each tile has a 16KB SRAM called Message Passing Buffer (MPB), shared by every two cores (in one tile), which is optimized to support message passing programming model for communication among all the cores (See Figure 1). These MPBs form a shared address space used for data exchange. The cores and MPB within each tile are connected to a router by the Mesh Interface unit.

There are four DDR3 memory controllers that provide each core in the SCC with its own private memory, supported by moving across the network to one of the four memory controllers and finally to the off-chip DRAM. The same DRAM can be configured as shared memory as well, but as with all shared memory on the SCC architecture, there is no built-in support for cache coherence between cores. Coherency, if it exists at all between cores, is the responsibility of the software [3].

Figure 1 – Tile Internal Architecture. Message Passing Buffer [3]

Figure 2 – SCC die physical layout [http://intelopenport.hosted.jivesoftware.com/community/marc]
The SCC is implemented using 45nm CMOS technology and has a total of 1.3 billion transistors. Each tile has an area of 18 mm$^2$ with a total die area of 567 mm$^2$ [2], similar to the size of a postage stamp, as seen in Figure 2.

The Single-Chip Cloud Computer was designed with fine-grained power management in mind. It comes with advanced power management technologies. Containing a configurable voltage regulator controller (VRC), the programmer has the capability of independently changing the voltage across the entire chip [1]. The cores in the SCC are divided into six voltage domains, each containing a 2x2 array of tiles (a total of eight cores) as shown in Figure 3. There is one voltage domain for the memory controllers and another one for the mesh, completing a total of eight voltage domains along the chip.

Clocking may be adjusted at an even finer granularity with each tile on SCC able to have its own operating frequency. There is also a separate setting for the mesh, providing a total of 28 distinct frequency domains. The voltage and frequency islands, together called power domains, enable the programmer to adjust the power/performance settings of different parts of the SCC to a particular performance level. These features would allow the implementation of Dynamic Voltage and Frequency Scaling (DVFS) techniques [5] that may continuously adapt to use the minimum power needed at a given moment, resulting in energy savings whenever possible.

3. ALGORITHM OVERVIEW AND IMPLEMENTATION

The stereoscopic imaging algorithm for centroid determination is outlined and detailed in this section, as well as implementation details for simulation on the Intel SCC. The algorithm utilized for this research is comprised of three sections: loading, classical image-processing techniques including contrast enhancement and morphological operators, and finally centroid determination.

Stereoscopic imaging

Loading of images will depend on the type of platform on which the simulation is running. However, for this research, the images will be loaded from main memory on a PC and the SCC platforms. Once the data is loaded onto the respective platform, it will then be subjected to the stereoscopic imaging algorithm for centroid determination that is outlined in Figure 4.

The images from the two cameras are in color and are converted to greyscale to avoid the processing and storage of more data than necessary. Conversion to greyscale also allows for a more straight-forward implementation of the image-processing techniques that follow. The next step in the process deals with the pixel-wise subtraction between the two images. Since the direction and orientation of the two cameras is similar and the range to the background is effectively infinite, the pixel-difference operation should eliminate most data in the background of the image while extracting information in the image foreground. This is due to the parallax effect, in which objects further away from an imager appear to move slower as projected across the focal plane array when compared with objects that are closer to the imager.

The resultant of the pixel-difference operation is a single image containing little background data and two foreground objects. In order to further separate the objects from the background, contrast enhancement is performed on the image and it is then subjected to an empirically defined threshold for conversion to a binary image. The morphological image processing operators of erosion and dilation are then performed on the binary image. This is a two-step process which encompasses a total of ninth order erosion and dilation using a 2x2 square operator as well as a disc operator of radius four. The purpose of the erosion and dilation process is to extract the objects of interest while
eliminating spurious data that may be left from the binary conversion process. After the erosion and dilation process is completed, only the two objects of interest are left with no background data, and centroid determination of both objects are determined.

Algorithm Implementation on SCC
The algorithm outlined above will be run in a parallel implementation on the Intel SCC. The SCC is first initialized and configured in regards to how many cores will be loading and executing the algorithm. Each core is designated with a voltage-frequency gear. A gear is defined by a specific voltage and frequency combination. At this point, the specified number of cores for the current simulation will load image data from either main or local memory and run the stereoscopic processing for centroid determination. The results will then be output to a single core. During the process, the SCC will be monitored and measurements will be logged for the power and performance metrics, which will be presented later. The SCC implementation process is illustrated in Figure 5.

4. RESULTS
Although Intel’s SCC may be a relevant platform in many of the aspects mentioned in Section 2, it does not excel for the core microarchitecture employed, which is based on a very simple architecture that was first introduced by Intel in March 1994 as the Pentium P54C version of the Pentium™ class. Although a few modifications were done for this platform, it continues to be a simple architecture, lacking out-of-order execution among a few other aspects incorporated within modern processors [6]. Therefore, a comparison of the results obtained with the SCC using only a few cores rather than trying to exploit the platform as a whole may not illustrate the real benefits that can be obtained when using this platform in particular and many-core architectures in general. In this section we present the performance results offered by the SCC and compared them to those obtained from a commercial PC running Matlab with an Intel core i5 processor operating at 2.30 GHz and containing 8 GB of RAM. It took around 52 seconds for our reference platform to completely process one image pair.

During all the experiments that were run in the SCC, which results are illustrated in Figures 6 – 9, only one image pair is being processed by each core during each run. In the case of
one core, only one image set was processed. In the case of two cores, only two image sets and so on up to a total of forty-eight.

The performance results are presented in Figure 6. It can be observed how increasing the processing load does not linearly impact the processing time. When utilizing one to twelve cores, it takes the platform around 64 seconds to process the images. As we increase the number of cores from thirteen to eighteen, the time to complete the algorithm increases linearly up to approximately 71 seconds, and for a core utilization of nineteen or higher, the processing time remains around 71 seconds.

Figure 6 also presents the ratio between the processing time and the total number of image pairs processed. This curve shows a dramatic decrease in delay per image pair as the number of cores increases, ranging from around 64 seconds when only one core is being used down to less than 1.5 seconds when using all forty-eight cores.

The total power consumed by the cores, as well as power per image pair processed, are presented in Figure 7. This graph shows how increasing the number of cores results in an increase in power consumption. When only one core is used for processing, the platform consumes around 51 watts of power, and for full chip utilization this number increases up to 98 watts. It should be mentioned that in the cases where only a few cores take part processing the application, all the remaining cores are idle but still ON and therefore consuming power. If total power is required to be optimal, selecting the configurations that use lower core count might be the right choice.

Figure 8 – Total Energy Consumption and Ratio Between Total Energy and Number of Samples Processed

From the right side of Figure 7, representing the ratio of power per image pair, one can observe a dramatic decrease in power per image pair as the number of cores increases. It ranges from the initial 51 watts for one image pair down to 2 watts per image pair when utilizing all the cores.

After analyzing the results obtained from the performance and power metrics, results containing the energy consumed by each configuration are shown in Figure 8. This graph reaffirms how the energy consumed by a system is dependent on the balance between performance and power, not simply processor speed or power dissipation alone. For the absolute energy consumption values, it can be observed that when only one core is used for processing, the platform consumes around 3300 joules of energy to complete its computations, while for full chip utilization this number increases up to 6940 joules.

The ratio of energy per image pair presented behaves similarly to the power metric. The relative energy consumed by image pair processed has a quadratic decrease as the number of cores increases. It goes from the initial 3300 joules per image pair down to less than 145 for forty-eight cores.

The ratio of energy per image pair presented behaves similarly to the power metric. The relative energy consumed by image pair processed has a quadratic decrease as the number of cores increases. It goes from the initial 3300 joules per image pair down to less than 145 for forty-eight cores.

From the right side of Figure 7, representing the ratio of power per image pair, one can observe a dramatic decrease in power per image pair as the number of cores increases. It ranges from the initial 51 watts for one image pair down to 2 watts per image pair when utilizing all the cores.

After analyzing the results obtained from the performance and power metrics, results containing the energy consumed by each configuration are shown in Figure 8. This graph reaffirms how the energy consumed by a system is dependent on the balance between performance and power, not simply processor speed or power dissipation alone. For the absolute energy consumption values, it can be observed that when only one core is used for processing, the platform consumes around 3300 joules of energy to complete its computations, while for full chip utilization this number increases up to 6940 joules.

The ratio of energy per image pair presented behaves similarly to the power metric. The relative energy consumed by image pair processed has a quadratic decrease as the number of cores increases. It goes from the initial 3300 joules per image pair down to less than 145 for forty-eight cores.

It is very common to consider the power-performance as the metric for selecting a best configuration to use. Usually, different configurations would impact different metrics; it being very uncommon to have one configuration that best satisfies all the metrics at the same time. From a system point of view, minimizing the execution time, the power or the energy consumption may be the first priority. However, even if the total energy is minimized, the user may not be
satisfied with extended system response time and vice versa [7, 8]. Because of this, the energy-delay product (EDP) is also considered as a metric to evaluate the overall implementation of our application in the Intel’s SCC.

In Figure 9 EDP metric is presented for each used configuration. The results from this metric have a very similar behavior to those of the power and energy. It can be noticed how increasing the amount of cores increases the absolute EDP values, but at the same time the EDP per image pairs ratio rapidly becomes better.

5. CONCLUSIONS AND FUTURE WORK

Improvements in performance and relative navigation solutions over single thread systems can be achieved through multi- and many-core computing. Stochastic relative orbit determination methods benefit from the higher measurement frequencies, allowing them to more accurately determine the associated statistical properties of the relative orbital elements. In addition, more accurate orbit determination can lead to reduced fuel consumption and extended mission capabilities and duration.

The Single-Chip Cloud Computer continues Intel’s and the whole industry’s trend toward employing higher core counts to replace the increases in clock frequency that have resulted in high power requirements, difficult to control high temperature, associated lower reliability, and chip design difficulties [6].

Knowing that the many-core era is close to arriving, we can expect the commercialization and availability of many-core platforms will not be limited to the research community only. Therefore projecting not only future but also actual applications toward many-core architectures is a critical step that would warranty exploiting the potential benefits offered by those architectures over single or few thread systems; specifically for real-time visual-based satellite proximity operations missions, where the nature of these applications allows a maximum utilization of the host platform.

As for future work, the algorithm implemented in this project may be further optimized targeting the SCC architecture. This should offer not only better performance results, but improve the power and energy consumption as well.

Further research can also be done to increase the complexity and number of algorithms to be executed in the Intel’s Single-Chip Cloud Computer. It might serve as an intermediate step, which experiences would be helpful with having a smoother assimilation of future many-core platforms.

REFERENCES


**Biography**

**Paul McCall** received his B.S. degree in Electrical Engineering in 2008 from the Florida International University in Miami, Fl. He received his M.S. degree in Electrical Engineering in May 2010, from Florida International University. Currently, Paul is a Ph.D. candidate in the Department of Electrical and Computer Engineering at Florida International University. He conducts research in the area of space situational awareness and sensor modeling as member of the Center for Advanced Technology and Education (CATE) at Florida International University. He is a student member of IEEE.

**Dr. Malek Adjouadi** currently a Professor of Electrical and Computer Engineering at Florida International University, and the Director of the Center for Advanced Technology and Education funded by NSF since 1993. He is leading the efforts on the joint Neuro-Engineering program between FIU and Miami Children’s Hospital. His research interests are in image and signal processing, human-computer interfaces as assistive technology tools, and neuroscience applications in pediatric epilepsy.

**Gildo Torres** received his B.S. degree in Electrical Engineering in 2006 from the Higher Polytechnic Institute of Havana, Cuba. He received his M.S. degree in Computer Engineering in May 2012, from Florida International University. Currently, Gildo is a Ph.D. student in the Department of Electrical and Computer Engineering at Clarkson University. He conducts research in the area of computer architecture as member of the Computer Architecture and Microprocessor Engineering Lab (CAMEL) at Clarkson University.

**Dr. Chen Liu** received the B.E. degree in Electronics and Information Engineering from University of Science and Technology of China in 2000, the M.S. degree in Electrical Engineering from the University of California, Riverside in 2002, and the Ph.D. degree in Electrical and Computer Engineering from the University of California, Irvine in 2008, respectively. Currently he is an assistant professor in the Department of Electrical and Computer Engineering at Clarkson University, New York, USA. His research interests are in the area of multi-core multi-threading architecture, hardware acceleration for scientific computing and the interaction between system software and micro-architecture. He is a member of IEEE, ACM, and ASEE.

**Keith LeGrand** is currently an undergraduate student in Aerospace Engineering at the Missouri University of Science and Technology. His research interests are focused in the areas of astrodynamics and spacecraft embedded systems. Past internships have included the Air Force Research Laboratory and Garmin International, and he will be employed as an intern at Sandia National Laboratories in Summer 2013. He is the Deputy Chief Engineer for the Missouri S&T Satellite research team, which is developing two flight-worthy microsatellites to perform visual-based proximity operations. He expects to receive his B.S. degree in Aerospace Engineering from Missouri S&T in 2014.

**Jacob Darling** is currently a doctoral graduate student in Aerospace Engineering at the Missouri University of Science and Technology. His research is focused in the areas of astrodynamics and estimation theory. He is the Chief Engineer for the Missouri S&T Satellite research team, which is developing two flight-worthy microsatellites to perform visual-based proximity operations. He received his B.S. degree in Aerospace Engineering summa cum laude from Missouri S&T in 2011.

**Dr. Henry Pernicka** is currently an Associate Professor of Aerospace Engineering at the Missouri University of Science and Technology. He was a member of the Aerospace Engineering faculty at San José State University from 1990-2001. Dr. Pernicka has experience in mission design dating back to collaborating with NASA Goddard Space Flight Center with the SOHO, WIND, and Advanced Composition Explorer missions. He also received a Group Achievement Award for his work at JPL in assisting with formulating the Mars Observer trajectory. His research in astrodynamics involves libration point orbits, trajectory optimization, formation flight/proximity operations in low Earth orbit, and stationkeeping in geostationary orbit. He is also actively investigating technologies with applications to small spacecraft. Dr. Pernicka received his Ph.D., M.S., and B.S. degrees in Aeronautical and Astronautical Engineering from Purdue University.