CCSC Consortium for Computing Sciences in Colleges

Southeastern Region

27th Annual Southeastern Conference

Student Research Contest

Extended Abstracts

November 15 and 16, 2013
Furman University
Greenville, South Carolina

Table of Contents

Performance of Quasi-anharmonic Analysis (QAA) using GPGPU

Carol Sadek (Wofford College, Spartanburg, SC 29303)

Mentors: Arvind Ramanathan (Oak Ridge National Laboratory, Oak Ridge, TN 37830) and Chakra S. Chennubhotla (University of Pittsburgh, PA 15260)

Faculty Advisor: Dr. Angela Shiflet (Wofford College, Spartanburg, SC 29303)

Quasi-anharmonic analysis (QAA) is a novel computational technique developed for analyzing long time-scale Molecular Dynamics (MD) simulations, which provides insights into the conformational sub-states accessed by proteins and other bio-molecules. The statistical information offered by these sub-states represent the functionality of the proteins. The concomitant increase in MD data and the intrinsic high dimensionality of bio-molecular dynamics makes it inefficient for Central Processing Units (CPU)-based implementations of QAA to scale to large simulations. We hypothesized that QAA implemented on General Purpose computing on Graphics Processing Units (GPGPU) will enable these large data sets to be analyzed in near real time.

GPUGPU uses graphics processing units to perform computations generally performed by the central processing units in order to further parallelize the computation. In this research, we examined the performance of independent component analysis (ICA) and how it scales with GPGPU implementation. ICA is a component within QAA and is used to solve the Blind Source Seperation (BSS) problem. This problem attempts to separate a mixture of signals into its original signal matrix and a mixing matrix. The two main ICA algorithms studied in this project are Joint Approximate Diagonalization of Eigenmatrices (JADE) and FastICA. The JADE algorithm uses fourth order cumulant matrices to calculate the orthogonal transformation matrix. The FastICA algorithm uses negentropy, which measures the distance between the probability density functions of a random variable and a Gaussian variable, to calculate the orthogonal transformation matrix. The following three contrast functions are used to approximate the negentropy of a variable: tanh, cube, and Gaussian. We studied how the use of each function affected the performance of the FastICA algorithm using two different approaches. The mixing matrix is used to calculate the original source signals.

We implemented both the JADE and FastICA algorithms using a CPU approach as well as a Graphics Processing Units (GPU) – CPU hybrid approach and calculated the performance gains. Differently sized signal matrices were generated and analyzed using the two different algorithms to see how the performance timings changed as the matrix sizes increased. The GPU – CPU hybrid approach was found to overpower the CPU approach after matrices exceeded size 10^6 using both the JADE and FastICA algorithms. We also found that the cube contrast function outperformed the Gaussian and tanh functions using the CPU approach. However, using the GPU – CPU hybrid approach, all three contrast functions performed at about the same speed. Regardless of which contrast function was used, the FastICA algorithm performed faster when implemented using the GPU-CPU hybrid approach after matrix sizes exceeded 10^6. Further, the CPU approach could not handle matrices larger than 10^8. The JADE and FastICA algorithms will be used to analyze large molecular simulations on supercomputers using the GPU – CPU hybrid approach. The purpose of this project is to enable further analysis of large data sets efficiently and in near real-time.

GGC Sustainability: Modeling Parking System Dynamics

Author: Joshua Popp
Georgia Gwinnett College
Lawrenceville, Georgia
jpopp@ggc.edu
Faculty Sponsor: Dr. Anatoly Kurkovsky, School of Science and Technology

Georgia Gwinnett College (GGC) opened in 2006 and is experiencing an extremely rapid growth in student population, reaching about 10,000 in 2012. Subsequently GGC also experienced a rapid growth in faculty and staff, reaching about 1,000 in 2012. Due to this rapid population growth, there is a growing number of people on campus grounds at any given time and thus a greater utilization of parking spaces and shortages. The end result is members of GGC spending excessive amounts of valuable time locating an available parking space.

As of Fall 2012, GGC's parking infrastructure consisted of a total of just over 3000 parking spaces distributed across ten parking lots. Among these parking lots, three have less than 150 spaces, four have between 150 and 500 spaces, and the last three have over 500 spaces. During the peak hours of 9am to 2pm, at least five lots are above 90% utilization. The populations are more heavily concentrated the closer a parking lot is to a major course building.

Major goals of this project are to analyze specific aspects of GGC sustainability related to existing/extended capacity of the campus parking facilities, growing of the student population and design/run a simulation model in order to find an appropriate balance between them.

The projects boundaries and scope were clearly defined and outlined during the early stages. During this phase, three types of system variables and the logical modules needed for the system were identified. The inputs of the system (entities) include faculty, staff, and students arriving on campus grounds. The capacities of each parking lot, each destination building, and travel time from a parking lot to the destination building are all control variables. Outputs of the system are recorded as success or failures base on how long an entity remains in the system. Included within the system are decision modules for assigning entities destination buildings and record modules to keep track of data flow.

The basic information flow of the model starts with the input entity entering the system. The entity then transitions to a decision module, where the entity is assigned a building destination. A checking order for parking lots is assigned based on the building destination. A system clock is applied to the entity where each parking lot visited that is at capacity adds 1.5 minutes to an entity counter. Once the entity reaches a parking lot below capacity, it allocates a parking space resource. Finally the entity transitions from the parking lot to the building destination, adding the travel time to the entity counter. The entity passes through a final decision module, checking the counter's value. If the value is 15 minutes or less, then the entity is recorded as a successfully passed the system. If not it is recorded as a failure to pass the system. After this point the entity is terminated (exits the system).

The simulation was run through multiple scenarios, testing various aspects of the system. The model has been validated based upon data collected by the college as well as verifying entities within the model transitioning as intended, following the intended information flow.

Keywords: University System, System Dynamics, and Parking Simulation

Classroom Response Systems: Integrating Bring Your Own Device and Brain Computer Interface Capabilities

ROBERT CURTIS, DEREK DONALDSON
Georgia Gwinnett College
rcurtis@ggc.edu, ddonalds@ggc.edu

Faculty Sponsors:
Dr. Evelyn Brannock, Georgia Gwinnett College
Dr. Robert Lutz, Georgia Gwinnett College

BYOD, or bring your own device, is a relatively new concept that is emerging with the introduction of new technologies. This idea is more popular now since more and more people are bringing handheld devices, such as smartphones and tablets, with them wherever they go. The education and research fields are starting to take interest now that many students have the technology to support BYOD and BYOD has demonstrated added benefits in education, such as increased student retention (1).

Using BYOD to guide us, we created a virtual classroom response system, often referred to as a clicker system. A presenter (usually a professor in a learning environment) asks the listeners (usually students) questions in order to gauge the listeners' knowledge. The listeners anonymously answer the questions. Conventionally, this is accomplished with a special piece of hardware that usually costs around $40. The student often bears this cost. The answer statistics from the listeners are gathered and used by the presenter to enhance the learning experience. In past studies, use of clickers has been observed to increase attendance, class participation, student attention and to promote student engagement in the classroom (2). With our virtual clicker, we eliminate the need for proprietary hardware. The clicker app, named Ursi, was designed with device flexibility in mind; it can be accessed from any device that has a web browser and an Internet connection. Whenever a professor creates a session a small unique code is generated. The professor can share this code with the students so they can connect to that session on their devices. After the students answer a question, the professor can display statistics on that question; the data is then presented to the professor and audience in the form of graphs, numeric readings and other compelling user interfaces.

Another interesting aspect of creating our own clicker application is that it allows us to integrate it with other hardware, such as a Brain Computer Interface (BCI) device. Prior research into BCI devices was being driven by the medical industry to power prosthetics for amputees (3) and for neurological rehabilitation (4). The cutting edge of research is mostly focused on intrusive intracranial BCI devices that are poorly suited to the questions posed in the classroom, but are making huge leaps in reading and normalizing brain activity readings. The second tier of research involves wet/dry caps with hundreds of electroencephalography (EEG) reading sensors. These caps provide good readings but are still too intrusive for a classroom/casual environment. The caps usually require the subject's head/hair be coated with gel to get better readings, which could alienate many students. The retail market has several casual BCI devices available that are more suited to the classroom environment. These devices resemble headbands and even have cosmetic accessories available. This semester, research was conducted to build a working system to test the usefulness of the current generation of consumer BCI devices and explore their implications in the classroom, driven by our clicker application.

We incorporated the use of a Brain Computer Interface (BCI) into Ursi with the intent to measure

students' brain waves to gauge activity, attention and meditation. An incorrect answer indicates a need for remediation and perhaps a re-think on the instructor's part as to how the content was presented. There are an untold number of reasons that may explain this state. Has the student not paid attention? Has the professor performed poorly in the tasks? A correct answer in a clicker system usually shows progress has been made. However, what does a correct answer really tell you about a student's current cognitive state? What does it tell you about the pupil's confidence? Did they guess the answer? Could they have glanced at a fellow student's answer? Were they influenced by other sources? Were they able to guess the answer by the way the question was worded?

To incorporate both BYOD and BCI devices, an open architecture managed by a small footprint web server was implemented. We considered making native applications for computers then making native applications for Android and iOS. After doing some research, we decided to utilize the best of both of these options. We built the default front-end of Ursi as an HTML application, but designed it with device responsiveness as a key goal. It should not matter the size of the screen nor the performance of the viewing device to achieve great results and performance. In order to integrate with the BCI device, we needed a native app that would allow us to make custom library calls to fetch data from the device. We wrap our existing HTML application in a lightweight native thin client that has the capability to retrieve data from the BCI as well as integrate into existing data services on our servers. The application was written with firm boundaries in place that separate the BCI communication interface from the implementation. This allows us to easily replace the current BCI with the next generation when they are released.

Moving forward, we have ambitions of drawing more data out of the BCI and better presentation of cognitive data. This may require advances in BCI technology and more research into Human Computer Interaction to display these large quantities of data so that a professor can make use of it. The professor should be able to feel the "temperature" of the classroom from this data. This empowers the professor to know how well his/her teaching style is being received without waiting until the next big exam. This quick feedback loop could be used tailor instruction, and signal when remediation is required before a student is left behind.

Reference List:

(1) Nelson D. BYOD. Internet@Schools [serial online]. November 2012;19(5):12-15. Available from: Computer Source, Ipswich, MA. Accessed September 12, 2013.Ruth G. Lennon. 2012. Bring your own device (BYOD) with Cloud 4 education. In Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity (SPLASH '12). ACM, New York, NY, USA, 171-180. DOI=10.1145/2384716.2384771 http://doi.acm.org/10.1145/2384716.2384771

(2) Taneja, A. The Influence of Personal Response Systems on Student's Perceived Learning Outcomes and Course Satisfaction, C. 2009, Consortium for Computing Sciences in Colleges.

(3) Postelnicu C, Talabă D, Toma M. BRAIN COMPUTER INTERFACES FOR MEDICAL APPLICATIONS. Bulletin Of The Transilvania University Of Brasov, Series I: Engineering Sciences [serial online]. January 2010;3(52):99-106. Available from: Academic Search Complete, Ipswich, MA. Accessed September 12, 2013.

(4) Chee Siang A, Sakel M, Pepper M, Phillips M. Use of brain computer interfaces in neurological rehabilitation. British Journal Of Neuroscience Nursing [serial online]. June 2011;7(3):523-528. Available from: CINAHL Plus with Full Text, Ipswich, MA. Accessed September 12, 2013.

Computational Modeling of the pre-Bötzinger complex

Maya Shende
Roanoke College
Salem, VA
mshende@mail.roanoke.edu
Faculty Advisor: Dr. Daniel Robb
Department of Math, Computer Science and Physics, Roanoke College

The part of the brainstem which controls breathing, the ventral respiratory column (VRC), contains a small region called the pre-Bötzinger complex (preBötc). The preBötc is vital in generating and regulating the breathing rhythm, but the details of how it accomplishes this feat are not well understood. During the course of this summer, we created a simulation to explore the effect of different connectivity patterns within a computational model of the preBötc network, in order to better understand how connectivity influences the function of the preBötc.

The first step towards creating a computational model of a neural network was to model one neuron. In keeping with previous research done in this area, we used what is known as a Butera bursting neuron model [1], an extension of the basic Morris-Lecar model [2], enabling it to act in either spiking or bursting model. From here, we computationally networked multiple neurons together using various connection schemes. To create multiple networks, we started with random networks, and used an evolutionary algorithm, rewarding those networks with desired properties. In addition to coordinated rhythmic activity, an important test that can be conducted on a network of neurons to determine whether a connection scheme is is one that is likely to be seen in the human brain is to analyze the robustness of that network scheme to deletion of neurons ("ablation"). My portion of the project focused on developing a useful fitness function for characterizing the robustness of each network to ablation of neurons.

Using data collected in previous research of in vitro slices of the preBötc [3], I wrote a program that took the network activity data generated from the network being analyzed, and compared the amplitude at the beginning of ablation to the amplitude at the end of ablation. Based on the results of previous research, the amplitude, or amount of network activity, at the start of ablation is approximately three times that at the end of ablation. Therefore, in the evolutionary algorithm we are using to create connection schemes, we want to reward those networks that emulate the results of this previous research. The next step in this robustness-algorithm is to analyze how the period changes over the course of ablation in the networks being modeled. After this has been accomplished, I plan to conduct several simulations to analyze networks for this robustness element, in order to refine the fitness function I have created.

References:

1. Butera, R. J., Rinzel, J. and Smith, J. C. Models of respiratory rhythm generation in the pre-Bötzinger complex. I. Bursting pacemaker neurons. J. Neurophysiol. 82: 382-397, 1999.
2. Morris, C., & Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. Biophysical Journal, 35(1), 193-213.
3. Del Negro, C. A., Morgado-Valle, C., Hayes, J. A., Mackay, D. D., Pace, R. W., Crowder, E. A., & Feldman, J. L. (2005). Sodium and calcium current-mediated pacemaker neurons and respiratory rhythm generation. The Journal of Neuroscience, 25(2), 446-453.

Improving Biological Network Analysis with High Performance Computing

Chris Mims & Joe McKenzie, The Citadel

Faculty Advisor Dr. Michael Verdicchio

Modeling biological systems such as cancer in network form facilitates computational analysis and simulation that can lead to direct patient benefit. Using computational models can speed up efforts to discover critical processes or significant properties, and there is also no need for living subjects to conduct the studies. Boolean networks are among the simplest of network-based models, yet they have been shown to adequately model many of the complex dynamics of real biological systems. But, like all models, there are computational limits that hinder our ability to work with more complex systems.

The primary issue with using Boolean networks is the increasing computational burden as we increase the number of variables in the network. The computational time needed to simulate the system and study its properties increases exponentially with the number of variables. Therefore, an average computer requires an irrationally (or impossibly) large amount of time to analyze a highly complex system. Many efforts have been made to overcome this computational limitation since there are many biological models worthy of investigation that are just beyond the reach of current methods.

In recent work, sampling the Boolean states in a system showed significant improvements in feasible network sizes while still maintaining the ability to find critical properties of the system.

The sampling was an attempt to decrease the actual number of variables being considered, thus decreasing the amount of time needed to run simulations. While the sampling approach had success in increasing the feasible size of networks, it was still ultimately hindered by the same computational limits.

In order to further increase allowable network size while dealing with this persistent computation problem, we decided to take advantage of parallel resources to add more computational power to recent approaches. Harnessing the power of parallel computing gives us the ability to decrease the amount of time needed by increasing the number of tasks that can be performed concurrently. When using high-performance computing (HPC) systems, performance should theoretically increase by a factor of the number of processing units available.

Developing a parallel version of the recent sampling algorithm will allow us to take advantage of high-performance computing resources and facilitate the study of larger Boolean networks. We can then benchmark the parallel algorithm to gauge the performance improvements. The research will show the amount of performance increase that the new algorithm returns when using parallel resources, as well as analysis as to where we can possibly improve the algorithm by identifying and eliminating bottlenecks. Strategies for future experiments on real biological data using the HPC approach will be proposed. Even though a parallel approach will eventually face the same computational limits, even facilitating study of slightly larger network sizes will allow analysis of important systems previously just out of reach.

Improving Website Performance and Load Time

Zach Hall
Faculty Advisor: Dr. Kevin Treu
Furman University

Background Information

In the first phase of my research, I created a searchable database of all of the courses offered at Furman University. The database has a web front-end and runs on my inexpensive web hosting account. My web server is located in Germany, and many people complained to me about how slowly my pages would load, especially due to server ping. In the second phase of my research, I looked into various methods of speeding up my site without spending any money on relocating my web hosting.

What I Did

First, I noticed that all my static assets (images, scripts, and style sheets) weren't being cached by the browser. To improve performance, I enabled Apache's caching modules on all static files with an expiration date one month in advance. Caching dynamic content, including search queries, could also be done, except that my database synchronizes with Furman's database every 20 minutes, and data can change during these synchronizations. Therefore, I set all my PHP scripts to cache for 15 minutes, as to save a history of previous queries in a single session for quick retrieval.

In addition to caching, I enabled GZIP compression on all pages, resulting in a large performance bump the first time that resources were served. I finally added a Content Distribution Network (CDN) to serve all my static content from multiple servers all throughout the world, rather than from my web hosting provider in Germany. A CDN is a collection of servers in various locations all over the world that will cache webpages from a website and re-serve them from locations that are closer to the visitor. I found a company, CloudFlare, that provides a CDN service for free, and enabling it on my website decreased the latency of all response times from 143ms on average to 11ms on average in the eastern United States. With this combination of client-side caching and server caching through a CDN, I eliminated most of the speed problems that I encountered.

As far as the application code itself, I rewrote large SQL queries to perform fewer operations and used UNION and JOIN statements more sparingly, resulting in an overall reduction in query times. I utilized Amdahl's law on my mobile site by starting users on a "Browse" page that would list commonly accessed categories that could be readily cached by my CDN and MySQL's query caching methods. Any category on the browse page will load nearly instantly since they're accessed all the time.

Results

I tested my website before and after I did these optimizations with two different widely-used analysis tools: Yahoo!'s YSlow and Google's Page Speed. These optimization services are recognized as authoritative standards on website loading speeds, and provide scores (0-100, F-A) to assess how a webpage's load time compares to others. Using my optimization techniques, I improved my YSlow score from 74 to 81 and my Google Page Speed score from B to A.

Future Work

While my dynamic content caching system works well, it's not entirely optimal. I cache query results for 15 minutes, and updates happen every 20 minutes. If someone requests a new page at 10:15, the results will be saved until 10:30, even if the database changes at 10:20. I've observed that changing a GET parameter in my

search script will re-cache the results, however, and since I know exactly what times the database updates occur, I can fix this problem by including a JavaScript function in my search page that will append a GET parameter containing a timestamp that changes every 20 minutes when the database updates, therefore allowing the user to cache data without worrying that their data is out of date.

Caching example:

Before:

10:00 AM	Database update	
10:15 AM	`/search.php?query=compsci`	Cache expires: 15 minutes
10:20 AM	Database update	
10:25 AM	`/search.php?query=compsci`	Cached version. **Out of date.**

After:

10:00 AM	Database update. Set variable “t” to 1000	
10:15 AM	`/search.php?query=compsci&t=1000`	Cache expires: 20 mins
10:20 AM	Database update. Set variable “t” to 1020	
10:25 AM	`/search.php?query=compsci&t=1020`	Re-cached for 20 mins

Using a GET parameter added by JavaScript, it’s possible to force a cached result to re-cache before it expires.

SQL optimization example

Before:

```
SELECT
        lec.*,
        lab.location AS 'Lab Location',
        /*lab.`Start Time` AS 'Lab Start Time', TESTING*/
        COALESCE(NULLIF(lab.`Start Time`, ''),'TBA') AS 'Lab Start Time',
        lab.`End Time` AS 'Lab End Time',
        lab.`Meeting Days` AS 'Lab Meeting Days',
        ins.`rating` AS 'Rating',
        enr.`Seats Remaining` AS 'Seats Remaining'
FROM
        classes AS lec
LEFT OUTER JOIN
        classes AS lab
ON
        lec.section = lab.section
LEFT OUTER JOIN
        instructors AS ins
ON
        lec.`Instructor(s)` = ins.name
LEFT OUTER JOIN
        enrollment AS enr
ON
        lec.section = enr.section
WHERE
        lec.`Instructional method` = 'LEC'
        AND lab.`Instructional method` != 'LEC'
        AND lab.`Instructional method` != 'EXAM'
        AND (lec.`section` LIKE '%$query%' OR lec.`title` LIKE '%$query%' OR lec.`Instructor(s)` LIKE '%$query%')
        AND lec.`term` LIKE '%$term%'
        $core
        $global
        $start_time
        $days
GROUP BY
        lec.Section
UNION
SELECT
        lec.*,
        NULL AS 'Lab Location',
        NULL AS 'Lab Start Time',
        NULL AS 'Lab End Time',
        NULL AS 'Lab Meeting Days',
        ins.`rating` AS 'Rating',
        enr.`Seats Remaining` AS 'Seats Remaining'
FROM
        classes lec
LEFT OUTER JOIN
        instructors AS ins
ON
        lec.`Instructor(s)` = ins.name
LEFT OUTER JOIN
        enrollment AS enr
ON
        lec.section = enr.section
WHERE
        `Instructional Method` != 'EXAM'
        AND `Instructional Method` != ''
        AND (lec.Section LIKE '%$query%'
                        OR lec.Title LIKE '%$query%'
                        OR lec.`Instructor(s)` LIKE '%$query%')
        AND lec.`term` LIKE '%$term%'
        $core
        $global
        $start_time
        $days
AND NOT EXISTS
        (SELECT *
```

After:

```
SELECT
        lec.*,
        lab.location AS 'Lab Location',
        lab.`Start Time` AS 'Lab Start Time',
        lab.`End Time` AS 'Lab End Time',
        lab.`Meeting Days` AS 'Lab Meeting Days',
        enr.`Seats Remaining` AS 'Seats Remaining',
        descriptions.`Description`
FROM
        classes AS lec
LEFT OUTER JOIN
        classes AS lab
ON
        lec.section = lab.section
        AND lab.`instructional method` != 'LEC'
        AND lab.`instructional method` != 'EXAM'
LEFT OUTER JOIN
        enrollment AS enr
ON
        lec.section = enr.section
        AND lec.term = enr.term
LEFT OUTER JOIN
        descriptions
ON
        lec.`Courses ID` = descriptions.`Courses ID`
WHERE
        lec.`instructional method` = 'LEC'
        AND (lec.`section` LIKE '%$query%' OR lec.`title` LIKE '%$query%' OR lec.`Instructor(s)` LIKE '%$query%')
        AND lec.`term` LIKE '%$term%'
        $core
        $global
        $start_time
        $days
GROUP BY
        lec.Section
ORDER BY Section ASC
        LIMIT $starting_point , 20";
```

In my first SQL queries, I used nearly twice as many SELECT and JOIN statements as I do now in my optimized query that does the same thing. By refactoring my old code, I decreased the amount of operations necessary for each query and also gave each query a large bump in performance.

Conclusion

In conclusion, I’ve found that you can greatly speed up page speed on web pages without spending a lot of money to relocate your servers. I found that refactoring old code is something that should be done regularly, and that using multiple speed benchmarks (ping tests, caching tests, and testing tools such as Google Page Speed and Yahoo! YSlow) can give a much better picture of overall performance than just relying on a single type of test or trying to implement a single fix.

Bipedal Robots: What are the issues with dynamic bipedal movement?

Cash Willis
Murray State University
Murray, KY
Cwillis3@murraystate.edu
Faculty Sponsor: Dr. Robert Pilgrim, Computer Science

In bipedal humanoid robots there are two types of walking. They are static and dynamic. Static walking assumes that the robot is statically stable. This means that, at any time, if all motion is stopped the robot will stay indefinitely in a stable position. It is necessary that the projection of the center of gravity of the robot on the ground must be contained within the foot support area. This kind of walking requires large feet, strong ankle joints and can achieve only slow walking speeds. Today, most of the active research is indynamic walking, which provides more realistic and agile movements [1]. This type of walking is much more challenging since it requires additional sensors, and control software for maintaining balance while starting and stopping.

Bipedal dynamic walking allows the center of gravity to be outside the support region for limited amounts of time. There is no absolute criterion that determines whether the dynamic walking is stable or not. Indeed a walker can be designed to recover from different kinds of instabilities. However, if the robot has active ankle joints and always keeps at least one foot at on the ground then the Zero Momentum Point (ZMP) can be used as a stability criterion. The ZMP is the point where the robot's total moment at the ground is zero [4]. Dynamic walking ensures that the robot can start walking from any point.

The goal of this research is to determine the main issues faced with dynamic movement in bipedal robots and whether an Arduino processor can be used to construct a low cost bipedal robot that walks dynamically. For the sake of greater efficiency in walking and a low energy requirement, the construction of a passive dynamic bipedal robot is being employed to research the main issues faced with dynamic movement in bipedal robots.

References

[1] EV Cuevas, R Rojas, D Zaldívar, “Bipedal robot description” pp.2, 2005.

CPU Scheduling Simulator

Isuru Wijesundara – iwijesundara@student.methodist.edu
Methodist University
Fayetteville, North Carolina
Advisor: Dr. Tat Chan – tchan@methodist.edu

Introduction:
I am Isuru Wijesundara a senior at Methodist University. My majors are Computer Information Technology and Accounting. Upon completion of my undergraduate studies, I am planning to attend graduate studies in the area of computer/ cyber security.

Credits:
The research that I am conducting for this research contest is a continuation of research that was completed in 2004 by my advisor Dr. Tat Chan. In addition, I am referencing the *Operating System Concepts* text book by Abraham Silberschatz, Peter Galvin, and Greg Gagne.

Problem:
There are many operating system architecture concepts that are harder to digest by reading about them. Those concepts require hands on activities to fully understand the mechanism. Because of this nature many computer science students consider the Operating System Concepts subject as one of the hardest subjects. Most of these concepts were developed by scientists not simply reading about possibilities or imagining how things should work; instead, scientists worked on operating systems and conducted various experiments to come up with better concepts. Therefore, I think in order to study this practical subject students should have access to hands on experiments.

Solution:
As I mentioned before, my advisor, Dr. Chan, first identified this issue back in 2005, and he created a simulator to let students understand the concept of job/ process scheduling. With a simulator, students will be able to understand and challenge the concept that is simulated. Therefore, a simulator will not only be a learning tool but also be a bread board for developing better future concepts.

Simulated Concepts and Outcome:
Initially, Dr. Chan created the simulator to simulate one of the job/ process scheduling concepts, which is round-robin (RR). I am taking this simulator to another level by giving the capability to simulate another three job/ process scheduling concepts such as first-come, first-serve (FCFS), shortest job first (SJF), and Priority (P) scheduling. Furthermore, my simulator will have the ability to simulate the solutions for identified weaknesses of the mentioned scheduling concepts such as convoy effect and starvation. The simulator has the ability to present results either as a report or on graphs. Job/ process scheduling is one of the main concepts associated with operating systems. Proper understanding of process schedule is vital for efficient programming. Therefore, this simulator will help students at Methodist and any other education institution to become better professionals.

Technical Details:
CPU Scheduling Simulator is written in Java language and eclipse is used as the integrated development environment (IDE).

References:
Chan T. 2004. A Software Tool in Java for Teaching CPU Scheduling: Journal of computing science in colleges. Volume 1, April 2004: 257-263.
Silberschatz A, Galvin P, Gagne G. 2011. Operating System Concepts. Hoboken (NJ): John Wiley & Sons, Inc.

Analyzing Techniques for Robotic Localization Mapping

Thomas Lux, Randall Pittman

Faculty Advisor: Dr. Durell Bouchard

Computer Science Department, Roanoke College

In this paper, we compare the accuracy and effectiveness of multiple sensor platforms for autonomously mapping indoor environments. We examined the local and global accuracy of the maps produced by each sensor platform with the end goal of determining a best candidate for robotic mapping. Throughout this paper, we will walk through some basic steps required to prepare the sensors for mapping to allow prerequisite work to be factored into the performance of each sensor. Each platform was be cross-tested on multiple metrics including: the accuracy of individual readings gathered across varying surfaces and angles of incidence, the consistency of readings and error, and how well a simple mapping program can rebuild a room using each sensor.

The robotic platform that we utilized for the project was Lego Mindstorms. This device provided an interface with the Python programming language and a mount for our sensor platforms. After mounting a sensor to the Lego Mindstorms, we could begin the mapping process. We represented our sensor data in memory with a grid cell based two dimensional array. Each grid cell held a value representing a statistical likelihood that the given cell was occupied. The occupancy values were incremented based off of an average range of error for our sensors. Maps were built with multiple consecutive local scans that were then integrated into a global map. With this in mind, our program had the task of determining which areas were unexplored. Algorithms determined where to go by traversing the occupancy grid map in order to locate areas of unknown. Once the desired end location was returned, a path finding algorithm would generate a path to that location. The path finding algorithm relied on simple parameters such as avoiding obstacles and minimizing number of turns. Lastly, navigation functions combined real time sensor readings with known map features in order to more precisely estimate the current location of the robot in the map.

The sensor platforms tested include the Microsoft Kinect, standard RGB web cameras, and a pair of ultrasonic sensors. Each sensor platform had a specific set of challenges regarding autonomous mapping. In order to produce a consistent metric for gauging the effectiveness of each

sensor, we had to break down the capabilities of each platform. Next, we were able to generalize commonly encountered situations into fairly comparable, platform-independent, states. For example, both the Microsoft Kinect and RGB cameras provide multiple data readings per frame known to be adjacent to each other. Whereas, for ultrasonic sensors, data points are gathered in lone pairs. In order to compensate, we processed the ultrasonic data across larger time frames to find common map features. The results of this are evident in the produced local scans.

The quality of each sensor platform was based particularly on the accuracy of distance measurements alongside the consistency of data produced across varying surface textures and angles. The locations used to produce the local maps were generated by our program running with the Microsoft Kinect. The same locations were then used for local scans on each platform so as to properly compare their results.

After gathering data from each sensor platform, we discovered that the Kinect produces the most consistent data with the least noise. Currently we believe this to be the most effective sensor for robotic mapping. The RGB web cameras are considerably more accurate than the Kinect, but provide little consistency and fewer readings per second. This slower rate is due to the large quantities of image processing and data filtering required to obtain usable data from RGB. The Ultrasonic sensors produce a high accuracy when facing perpendicular surfaces, however they fail to detect surfaces at angles of impact greater than 30 degrees from perpendicular. We are currently processing the remaining data from the local scans. Thus far, the results show that RGB and ultrasonic data requires much heavier filtering than the Kinect. Ultimately, each sensor proved to be quite accurate in terms of distance, but the Kinect was the most reliable and consistent.

Beat and Emotion Tracking Mobile Radio

John Guidry
jrguidry@mail.roanoke.edu
Advisor: Dr. Durell Bouchard
bouchard@roanoke.edu

Often, people are too focused on their mobile devices. The focus should be on exercising and not the mobile device. Limiting the focus away from the phone and towards exercise leads to a safer environment for everyone. To do this, we created an application that has the capabilities of selecting songs based on user pace and user emotion.

The application is able to calculate the user's pace in order to make the user experience less involved. The application calculates the user's steps per minute using the accelerometer that is found in smart phones. This is done by calculating the magnitude of the x-axis, y-axis and z-axis given from the raw accelerometer data. The magnitude data is then smoothed for a better understanding of the user's steps per minute. We then used a moving threshold algorithm that looked at the smoothed magnitude graph and detects what a beat is by looking at the graph's peaks. After finding several beats, the application then sets the steps per minute of the user that is compared to the song's BPM.

The song BPM is found by scraping a website. This is done by entering the song's name and artist into a search function for a database. If the song is found, the song has a defined BPM that can be later compared to the user's steps per minute. If no song is found, the song has a defined zero BPM. Setting the song's BPM to zero allows for songs to be played if there was no match between user steps per minute and the song's with a defined BPM.

The application selects a song in a plus or minus range of ten beats per minute. If a song is found, then that will be the next song to be played. If no song is found, then the application picks a song that has a listed BPM of zero to allow for the continuation of music. The user can skip songs and the same algorithm for song selection will then run again to find the next song for the user.

Testing still needs to occur to see if users find the application helpful. Users are asked to walk or run for ten minutes using one of two applications. One of the applications simply shuffles the music library of the user and the other application does the steps per minute matching to song BPM. Both applications look identical. The user is then asked to do the same process again with the other application. A questionnaire is then given to the user that entails which application was preferred, what improvements could be made, and if the user would pay $0.99 for the application.

For future improvements and work, emotion applications need to be implemented. The emotion detection would work with a simple question given to the user at the start of the application. The question would entail asking about their current emotion(happy, sad, fine, etc.). The song selection would then work by comparing a dictionary of words that describe the user's emotion to the song's emotion. The process of emotion and BPM detection allows for a more relative song for the user.

Because of our application, the user would only need to access their phone for talking while they exercise. The application is changing the idea of music for the user. Instead of just shuffling their library, the application is selecting songs relevant to what the user would like to hear. We hypothesize the user would be using their phone far less for song selection since they will not be changing songs as often, and therefore allowing for a safer jog or walk to class.