Mobile Intelligent Sensor Network used for Data Processing

Joshua Etchison, Gordon Skelton, Qing Pang, Timothy Hulitt

Department of Computer Engineering Jackson State University, Jackson, MS 39217

***Abstract*—This paper describes the concept of a mobile wireless cluster used as an intelligent sensor network for data processing. The use of five portal computers operating under Linux communicating with each other to process and calculate was built as the framework of a mobile intelligent sensor network. The use of mobile intelligent sensor networks is only limited to the imagination of the world's engineers. The applications of wireless sensor networks are varied, typically involving some kind of monitoring, tracking or controlling. Our goal is to merge a wireless sensor network with a mobile cluster into one system capable of high performance computing. The system will collect and process data such as images for mapping. Each node will acquire data and send selected information to the server node, the sever node divides the data and has the client nodes process the information using parallel computing algorithms. This research is specifically designed for the Mars Rover application.**

***Index Terms*— Client node, intelligent sensor network, mote, Open MPI, server node, wireless Linux cluster**

I. Introduction

The recent advances in wireless communications, power systems, mobile robots, and sensor networks have opened the door for mobile intelligent sensor networks to be used for data processing. Wireless sensor networks (WSNs) are made up of sensors that are located throughout the environment and collectively work together to process information [1]. A WSN can be used to record an environment's conditions such as the temperature, pressure, moisture, pollutants, and other attributes. The development of many wireless sensor networks has been driven for use of military applications such as battlefield surveillance and solider awareness. They are now used in many industrial and civilian application areas, including machine health monitoring, industrial process monitoring and control, environment and habitat monitoring, healthcare applications, home automation, and traffic control [2], [3]. And soon, WSNs may become a part of everyday life.

The exploration of space and other planets has always intrigued scientist and engineers. Since the first human has set foot on the moon, the next goal of astronauts is to explore Mars. Many questions arise about the other planets. Is there water, oxygen, or life? What species are present, if any? If so, what kind? Are they intelligent? But the ultimate question is if humans survive on Mars? Using rovers capable of data acquisition and processing, all these questions and many more can be answered.

This research proposes to use various sensors, a low-powered atom processor, wireless router, solar panels, and a battery on mobile platform, like the Mars Rover, to be used as an mobile intelligent sensor network (MISN). The wireless network is a Linux cluster with one server node and multiple client nodes. The server node will send the commands for the client nodes to perform complex computations similar to how a multi-core system would perform multi-threaded processes for the same computation.

Many milestones must be overcome in order to complete the research proposed. The first milestone is to set up a Linux wireless cluster with one server node and four client nodes on a private network. First, experiments of the communication between the server node and each client node are needed. Then the ability to perform parallel computing with the entire cluster needs to be tested. Once parallel computing on the Linux cluster has been tested and verified, the development of parallel algorithms can begin.

The development of the robot and sensor network is not the focus of this research, at this time, since almost any mobile platform can be used for experiments. Our goal is to focus on the development of the wireless Linux cluster and multi-threaded programming for data processing, adapting the system into a battery powered and solar rechargeable format allowing for mobility of high performance computing.

The main components chosen for this project have been researched and selected due to their relatively power consumption, performance, and afford-ability. The need for a processor that is capable of processing large amounts of data as well as multi-core capabilities was our main requirement. Our main restrictions were to minimize the power consumption vital for the life of the battery attached to each node and find all devices necessary to be affordable.

II. Related Work

With the vast available applications of WSNs ranging from military to industry to academic research, many different applications have researched related to this technology. As of now they have been developed and used for military surveillance, perimeter security, structural health monitoring, industrial automation control, environmental monitoring, predictive maintenance. Most of the development for these WSNs are static location sensors, meaning these sensors do not change their position. However some research has been completed for mobile sensor networks. These systems have been researched for applications in rescue or disaster relief.

.

Primarily for local navigation, obstacle avoidance, and data collection.

Octave Technology is a company that specializes in the software side of WSNs. The framework they developed is a software platform between the WSN and the existing information system. Thus, allowing the integration of wireless sensors and information systems to be quicker and cheaper than for a company to develop a custom WSN. Octave Technology sells their framework to industries for the use of industrial equipment monitoring, environmental monitoring, security and safety applications, infrastructure and defense, building automation, and quality control [6].

At the University of Southern California in Los Angeles, the Computer Science Department has researched the area of mobile sensor networks deployment. They have added the feature of mobility to the WSN using potential fields in their deployment method. In their simulated experiments they have validated their algorithms for sensor network deployment in which each mote has an average spacing of 1.6 meters [7].

J. Lee and H. Hashimoto have developed a method of controlling mobile robots throughout an intelligent space. They use static locating distributed intelligent network devices, CPUs and sensors to detect obstacles and then send control information to the mobile robot to avoid obstacles. The algorithms they use are performed on a wired network, then communicates to the robot on what to do [1].

N. Hoe and P. Varshney have tested three different algorithms in order to find the most efficient method of deployment for MISNs. Hoe and Varshney tested a peer-to-peer algorithm, clustering algorithm, and a Voronoi diagram-based method. The peer-to-peer algorithm was achieved by placing a determined amount of nodes randomly throughout a testing area. They found this is not optimal because some nodes have sensors that overlap each other. The clustering algorithm is much like our method, where a hierarchy is implemented among the network of nodes. They have found this method to have energy efficiencies and if the remaining energy level is high then the node will contribute more to the performance of the WSN [5][9].

The research we are conducting integrates both mobility of sensor networks and high performance computing. Such that our motes are capable of performing obstacle avoidance and deployment as well as high performance computing such as, capturing and processing image data for mapping of an unknown environment.

III. System Functionality

One of the main goals of this project is to maximize functionality by allowing each node to become more than just a calculator on wheels. This project will allow each mote to operate independently by allowing them to make decisions based on it's individual needs, for instance, to relocate to read new data, relocate to recharge its battery under a better sunlight, or simply go to sleep when the need of several nodes is not necessary and wake up from power-saving mode when a "wake-up" bit is received from the server.

The server-client concept is being used for parallel programming. The server node will send requests to each client node to process data that requires high performance computing much like a typical multi-core PC runs multi-threads. Once the client returns its results, the server will process the results and may send commands to the client to modify their behavior or location.

The nodes should be able send request to the server for support as well. For instance, if a node gets into a position that it cannot get out of, the server node will be able to request a nearby node to aid in the retrieval of the node in need.

As the number of motes over the network increases, the time to complete complex computations will decrease. A network of multiple nodes readily available to process any data sent by the host machine is needed in order to processes image and mapping data as well being able to calculated decisions based on the surrounding environmental conditions. Concluding that the more nodes available on the MISN the faster complex algorithms can be computed and responded to. The Department of Mathematics at Colorado State University has validated this theory as can be seen in figure 1.

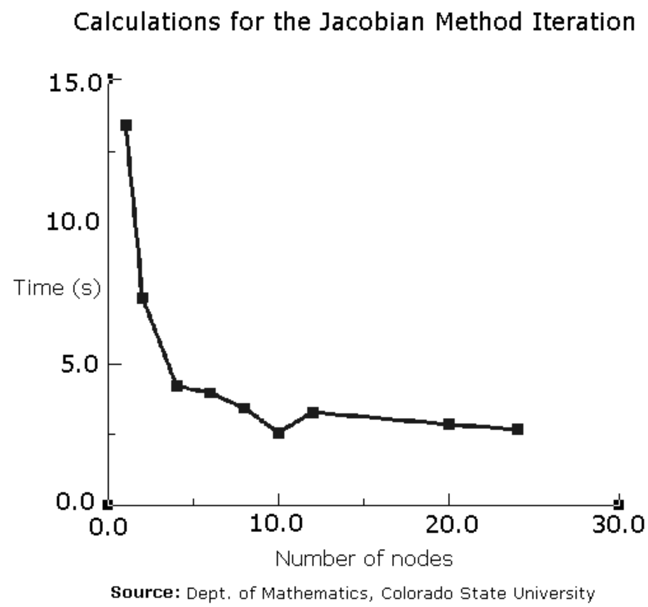

FIGURE 1

IV. System Architecture

A. Hardware Components

For this project, we have incorporated a Micro ATX motherboard with an Intel Atom dual core processor, 2GB of RAM, a 160 GB 3.5 inch hard drive, a Linksys USB wireless Ethernet adapter and a Linksys wireless router. The hardware is powered by a 12 volt / 5 amp DC power supply for lab testing. To enable mobility, we will use 12 volt batteries and commercially available solar cells. The battery and solar panels have not been implemented at this time. The mobile power source will be added once the communication and parallel algorithms have been verified. Each mote's three main jobs will be the acquisition, processing and transmission of data.

The data acquisition will be done locally by each node. The data the node gathers from its sensors will be analyzed directly then the node will send the needed information to the server node. Since we do not have direct access to the peripherals of the processor, the data transmission between the sensor and its respective node is yet unclear; some possibilities are to either transmit through a COMM port using RS-232 for a simple sensing device, or through an USB port for more advanced sensing and data acquiring devices such as video cameras, LIDAR equipment, or some high end GPS devices.

For data processing we are using the Intel Atom CPU. It was chosen due to its new high-performance architecture and low power consumption. If we compare the power consumption of the Intel Atom to a Pentium 4 2.0GHz, the P4 has a TDP rating of 38W while the Atom 2.0GHz processor has a TDP rating of 2.65W, which consumes just about 7% of what a Pentium 4 2.0GHz will consume.

Data is being transmitted using a wireless 802.11g router connected to the nodes at 54Mbps, offering enough data transmission speed while providing a range of about 300ft (outdoor under optimal conditions). This range can be extended if we decide to attach a high gain antenna to our wireless router.

B. Software

We have installed various software packages for this project. The first software we installed was Ubuntu Server Edition, version 8.1. It is an energy efficient, low memory, and low disk-footprint operating system. Ubuntu is an expanding ecosystem with minimal maintenance. One reason we chose the Linux software because it easily integrates with networks. We were also able to enjoy the unprecedented performance and security that Linux provides.

The second software we installed was Open-MPI. Open-MPI is a high performance message passing library. It is an open source MPI-2 implementation that is developed and maintained by a consortium of academic, research, and industry partners. Open MPI single library supports all networks as well as many operating systems. It also is tunable by installers and end-users. Therefore giving us the ability to modify and optimize if necessary. We installed Open MPI because we wanted to be able to pass a message between the server and client nodes. In order to communicate between the nodes, we had to create a host file with the IP addresses since the host file is included inside the MPI code.

The third software we installed was Open-SSH Client and Server packages. Open-SSH is a set of computer programs providing encrypted communication sessions over a computer network using the SSH protocol. It encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other attacks. Additionally, Open-SSH provides secure tunneling capabilities and several authentication methods. We installed the Open-SSH Client and Server because we required a secure way of communicating between the nodes. By us having an authentication method, we were able to make sure that the client and server were safely connected to each other. After the installation, we created a password and username that only the client and server could have access to.

V. Network Communication

Each node will interact directly with the server but not each other. Instead the host application in the server will decide if the unprocessed data from an unresponsive node (outside network range) should be delivered to a responding node and later discard the repeated results, should the unresponsive node return to the network.

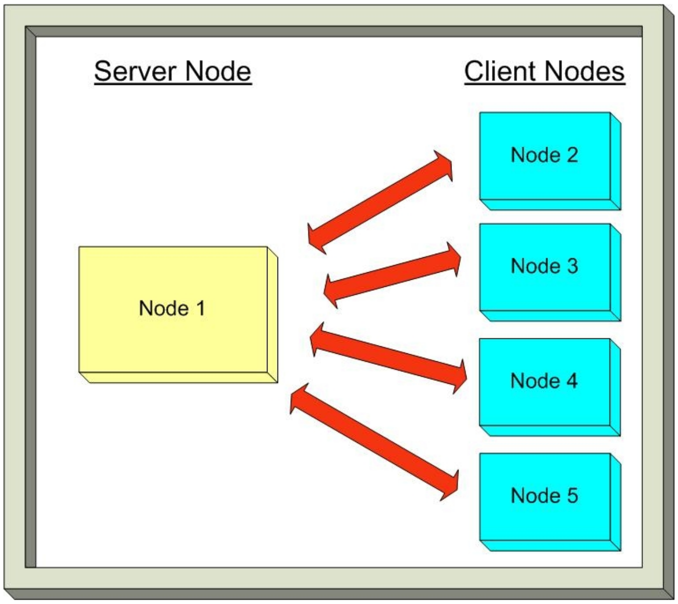

FIGURE 2

Assuming that we are dealing with automated mobile clusters, an algorithm to maintain proper signal strength is needed in order to relocate a node that is reaching the limits of the network signal range. Each node individually (multi-core system) will be running an independent process whose main goal is to maximize power efficiency by shutting down unnecessary system devices and relocate its current location to an optimal location that will still allow to capture the required data trough its sensors at the same time maintaining a live connection with the server. This same node will also report to the server if it needs to relocate to a position outside the range so that the server will tell the rest of the nodes to follow him, or relay his new GPS destination, and as soon as each node is through reading data from its location, they will automatically head near the GPS location relayed by the server in order to rejoin the network.

VI. Experiment

The first milestone is setting up the Linux cluster for parallel computing. Then testing and verifying the functionality of the the wireless Linux cluster must be completed before the development of the parallel algorithms can begin. To begin testing the Linux wireless cluster, the local accounts and private network needed to be set-up. This includes the wireless router and static IP addresses issued to each node.

The testing steps are as follows:

1. Ensure that all nodes are logged onto the private network.
2. List the static IP addresses of the each client node in the server node's host file.
3. Verify that the server node can log onto each client node using the Open-SSH command line.
4. Run the MPI output file on the server node to calculate pi.

To verify the test has run properly, we compiled a short program to converge pi using a central explicit scheme while measuring the turn around time of the computation. Three tests were done using this same program. The first test used only the server node and the turn around time was noted. The second test used the server node and two client nodes to converge pi. The turn around time was noted and compared to the first test. The second test, using three nodes, had a shorter turn around time than the first test, using only the server node to converge pi. The third test utilized all five nodes, the server and four client nodes. The turn around time was noted compared to the two previous tests which proved that the five nodes can indeed yield a shorter turn around time to converge pi.

In order to test for performance improvements, we have benchmarked the system and compared the calculation time and broadcast time of the cluster under both wired and wireless communications. Unfortunately, we have noticed some inconsistencies of communication time in our experiments with the wired cluster. Therefore we are unable to compare the two clusters effectively.

VII. Conclusion

A MISN is a new research topic for the authors. This research is still in progress and the only result obtained thus far is the verification that parallel computation was completed on the Linux wireless cluster. We tested a program that converged pi using a central explicit scheme and measures the turn around time of the parallel computations between the server node and four client nodes. We have verified that the computational time is reduced as the number of client nodes increase on a wireless cluster. However, we have noticed some inconsistencies in our data. We are currently debugging the system to discover where the inconsistencies are coming from. The next step will be to begin developing parallel algorithms and begin work on power systems utilizing batteries and solar cells.

References

[1] J. H. Lee and H. Hashimoto, "Controlling Mobile Robots in Distributed Intelligent Sensor Networks," *IEEE Trans. on Industrial Electron*, Vol. 50, No. 5, Oct. 2003.

[2] E. Sazonov, K. D. Janoyan, and R. Jha,"Wireless Intelligent Sensor Network for Autonomous Structural Health Monitoring," Smart Structures/NDE 2, 2004, San Diego, California..

[3] K. Romer and F. Mattern, "The Design of Wireless Sensor Networks," *IEEE Wireless Comm.*, Dec. 2004

[4] A. Garrison, "Parallel Computing: Comparing Different Domain Decomposition Techniques," Department of Mathematics Colorado State University, http://cauchy.math.colostate.edu/Projects/Garrison/paper.html

[5] N. Heo, P. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks," IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol 35, No. 1, Jan. 2005.

[6] Octave Technology Inc. website and data sheet, http://www.octavetech.com/index.html.

[7] A. Howard, M. Mataric, G. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem." 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02), Fukuoka, Japan June 2002.

[8] G. Pottie, W. Kaiser, "Wireless Integrated Network Sensors," Communications of the ACM, Vol. 43, No. 5, May 2000.

[9] N. Heo and P. Varshney, An intelligent deployment and clustering algorithm for a distributed mobile sensor network, *IEEE International Conference on Systems, Man and Cybernetics* (2003)