Invited paper in *CSI Communications*, Special Issue on Grid Computing, Computer Society of India, July 2005

SensorGrid: Integrating Sensor Networks and Grid Computing

Chen-Khong Tham[1] and Rajkumar Buyya[2]

Integrating sensor networks and grid computing in *sensor-grid computing* is like giving 'eyes' and 'ears' to the computational grid. Real-time information about phenomena in the physical world can be processed, modelled, correlated and mined to permit on-the-fly decisions and actions to be taken on a large scale. Examples include environment monitoring with prediction and early warning of natural disasters, and missile detection, tracking and interception. We describe some early work in sensor-grid computing, and discuss the research challenges that need to be overcome before such a vision can become reality, such as web services and service discovery, interconnection and networking, coordinated quality of service (QoS) mechanisms, robust and scalable distributed algorithms and efficient querying.

Keywords: Sensors, Sensor Networks, Grid computing, SensorML, SensorWeb.

1. Introduction

Recent advances in electronic circuit miniaturization and micro-electromechanical systems (MEMS) have led to the creation of small sensor nodes which integrate several kinds of sensors, a central processing unit (CPU), memory and a wireless transceiver. A collection of these sensor nodes forms a *sensor network* which is easily deployable to provide a high degree of visibility into real-world physical processes as they happen, thus benefitting a variety of applications such as environmental monitoring, surveillance and target tracking. Some of these sensor nodes may also incorporate actuators such as buzzers and switches which can affect the environment directly. We shall simply use the generic term sensor node to refer to these sensor-actuator nodes as well.

A parallel development in the technology landscape is *grid computing*, which is essentially the federation of heterogeneous computational servers connected by high-speed network connections. Middleware technologies such as Globus and Gridbus [1] enable secure and convenient sharing of resources such as CPU, memory, storage, content and databases by users and applications. This has caused grid computing to be referred to as 'computing on tap', utility computing and IBM's mantra, 'on demand' computing. Many countries have recognized the importance of grid computing for 'eScience' and the grid has a number of success stories from the fields of bioinformatics, drug design, engineering design, business, manufacturing and logistics.

The combination of sensor networks and grid computing in *sensor-grid computing* executing on a *sensor-grid architecture* (or simply a 'sensor-grid' in short) enables the complementary strengths and characteristics of sensor networks and grid computing to be realized on a single integrated platform – see Figure 1. Essentially, sensor-grid computing combines real-time data about the environment with vast computational resources. This enables the construction of real-time models and databases of the environment and physical processes as they unfold, from which high-value computations like decision-making, analytics, data mining, optimization and prediction can be carried out to generate 'on-the-fly' results. This powerful combination would enable, for example, effective early warning of natural disasters such as tornados and tsunamis, and real-time business process optimization.

The organization of this article is as follows. In Section 2, we describe a simple way to realize sensor-grid computing which we call the *centralized* approach. We then point out some of its weaknesses and describe a *distributed* approach. In Section 3, we describe two applications of distributed sensor-grid computing which we have implemented. In Section 4, the challenges and research issues related to sensor-grid computing are discussed. Finally, we conclude in Section 5.

[1] National University of Singapore
[2] University of Melbourne

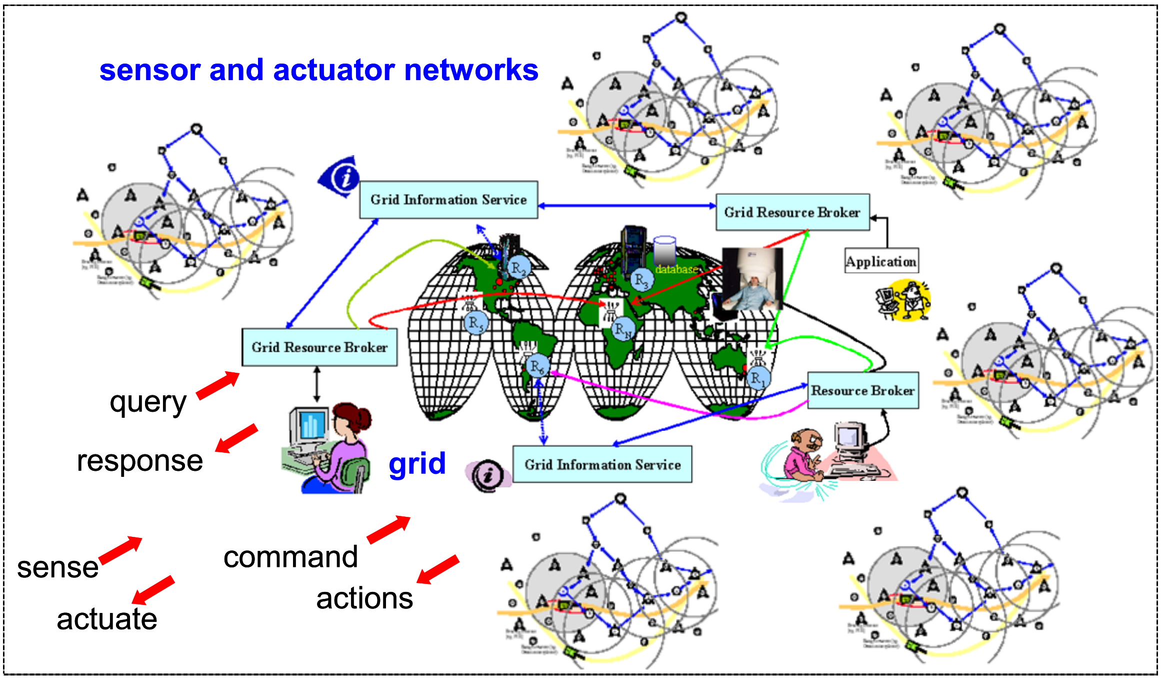

Figure 1 – SensorGrid integrating Sensor Networks and Grid Computing.

2. Approaches to Sensor-Grid Computing

One simple way to achieve sensor-grid computing is to simply connect and interface sensors and sensor networks to the grid and let all computations take place there. The grid will then issue commands to the appropriate actuators. In this case, all that is needed are high-speed communication links between the sensor-actuator nodes and the grid. We refer to this as the *centralized sensor-grid computing* approach executing on a centralized sensor-grid architecture.

However, the centralized approach has several serious drawbacks. Firstly, it leads to excessive communications in the sensor network which rapidly depletes the batteries. It also does not take advantage of the in-network processing capability of sensor networks which permits simple processing and decision-making to be carried out close to the source of the sensed data. In the event of communication failure, such as when wireless communication in the sensor network is unavailable, e.g. due to jamming, the entire system becomes inoperational.

The more robust and efficient alternative is the decentralized or *distributed sensor-grid computing* approach which executes on a distributed sensor-grid architecture and alleviates most of the drawbacks of the centralized approach. The distributed sensor-grid computing approach involves processing and decision-making within the sensor network and at other levels of the sensor-grid architecture.

3. Implementations of Distributed Sensor-Grid Computing

Distributed information fusion and distributed decision-making are two applications that are well-suited for distributed sensor-grid computing.

3.1 Distributed information fusion

Since the nodes in a sensor network are independently sensing the environment, this gives rise to a high degree of redundant information. However, due to the severely resource-constrained nature of sensor nodes, some of these readings may be inaccurate. Information fusion algorithms compute the most probable sensor readings and have been studied extensively over the years in the context of target detection and tracking.

We implemented a hierarchical decision fusion system comprising two levels of Crossbow motes (at the local or ground level), grid clients (at the regional level) and grid server nodes (at the global level) to detect and classify forest fires of varying degrees of severity, ranging from 'local fire', 'small forest fire' to 'large forest fire'. The local classifier in each sensor node is a Bayesian Maximum A Posteriori (MAP) classifier and the decision fusion algorithm described in Duarte and Hu [2] is implemented at the fusion centers. During operation, the decision fusion algorithm produces the final classification outcome based on the most frequent class label among the training samples which produced the same decision vector as the one encountered during operation. This decision fusion algorithm is robust and produces high classification accuracy in the final classification even in the presence of faulty or noisy sensors.

In an enhanced version of the above, two further levels in the form of Stargate and iPAQ cluster heads were added between the sensor nodes and grid client levels. The resulting system can be seen in Figure 2. The addition of these two levels enable more complex processing to be done close to the source of the sensor data and reduces the communication distances between the different levels, thus conserving power and improving the timeliness of the global classification.

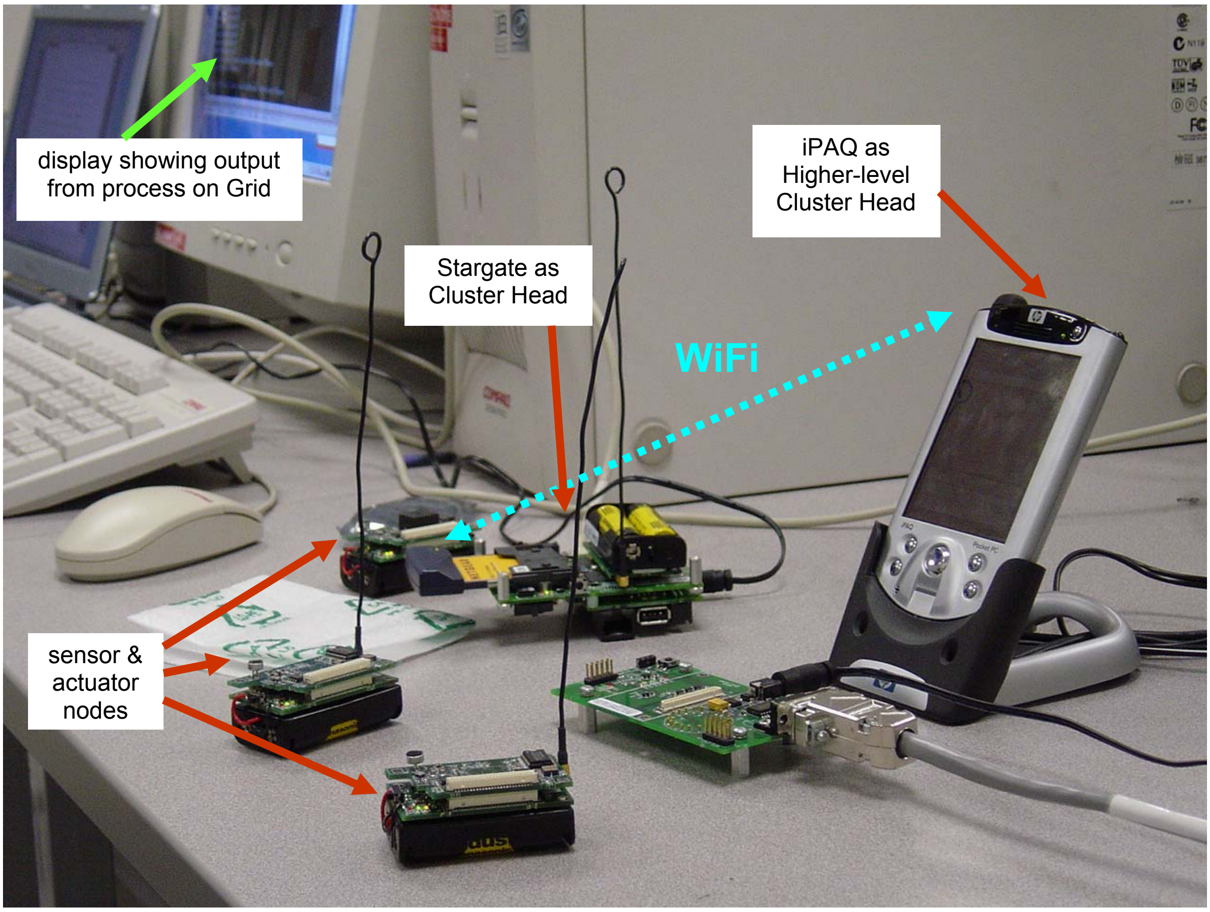

Figure 2 – Hierarchical decision fusion system on sensor-grid architecture.

3.2 Distributed autonomous decision-making

There are many cases in which some response is needed from the sensor-grid system, but the best action to take in different situations or *state*s is not known in advance. This can be determined through an adaptive learning process, such as the Markov Decision Process (MDP) or reinforcement learning (RL) [3] approach. MDP problems can be solved off-line using methods such as policy- or value-iteration, or on-line using RL or neuro-dynamic programming (NDP) methods.

A multi-level distributed autonomous decision-making system can be implemented on the hierarchical sensor-grid architecture shown in Figure 2. We implemented basic NDP agents in Crossbow motes at the local or ground level, and more complex NDP agents at grid server nodes at the core of the grid. Each NDP agent is able to act autonomously such that the entire sensor-grid remains responsive despite communication failures due to radio jamming, router faults etc.

4. Research Issues

Sensor networks is a relatively recent field and there are many research issues pertaining to sensor networks such as energy management, coverage, localization, medium access control, routing and transport, security, as well as distributed algorithms for target tracking, information fusion, inference and optimization.

Grid computing has been in existence longer, but nevertheless, still has a number of research challenges such as fair and efficient resource (i.e. CPU, network, storage) allocation to achieve QoS and high resource utilization, workflow management, the development of grid and web services for ease of discovery and access of services on the grid, and security. Resource allocation itself involves a number of aspects such as scheduling at the grid and cluster or node-levels, Service Level Agreements (SLAs) and market-based mechanisms such as pricing.

Apart from the afore-mentioned research issues in sensor networks and grid computing, sensor-grid computing gives rise to additional research challenges, especially when it is used in mission-critical situations. These research challenges are: web services and service discovery which work across both sensor networks and the grid, interconnection and networking, coordinated quality of service (QoS) mechanisms, robust and scalable distributed algorithms, and efficient querying. Each of these will be discussed in greater detail in the following sub-sections.

4.1 Web services-based Sensor Networks and Distributed Processing

The Grid is rapidly advancing towards a utility computing paradigm and is increasingly based on web services standards. The Service-Oriented Architecture (SOA) approach has become a cornerstone in many recent grid efforts. It makes good sense to have an SOA-approach as it enables the discovery, access and sharing of the services, data, computational and communication resources in the grid by many different users.

Likewise, in sensor networks, it makes sense to share the sensor-actuator infrastructure among a number of different applications and users so that the environment is not swamped with an excessive number of sensor nodes, especially since these nodes are likely to interfere with one another when they communicate over the shared wireless medium and decrease the effectiveness of each node, and actuators may also take conflicting actions.

There has been some recent work on adopting service-oriented architecture and web services approach to sensors and sensor networks. The OpenGeospatial Consortium's Sensor Model Language (SensorML) [6] standard provides the XML schema for defining the geometric, dynamic and observational characteristics of sensors. The purpose of SensorML is to:

(1) provide general sensor information in support of data discovery,
(2) support the processing and analysis of the sensor measurements,
(3) support the geolocation of the measured data,
(4) provide performance characteristics (e.g. accuracy, threshold, etc.), and
(5) archive fundamental properties and assumptions regarding sensor.

SensorML provides a functional model for sensor, not necessarily a detailed description of hardware. It supports rigorous geolocation models, which can describe sensor parameters independent of platform and target, as well as mathematical models which can directly map between sensor and target space. SensorML can be applied to virtually any sensor, whether in-situ or remote sensors, and whether it is mounted on a stationary or dynamic platform.

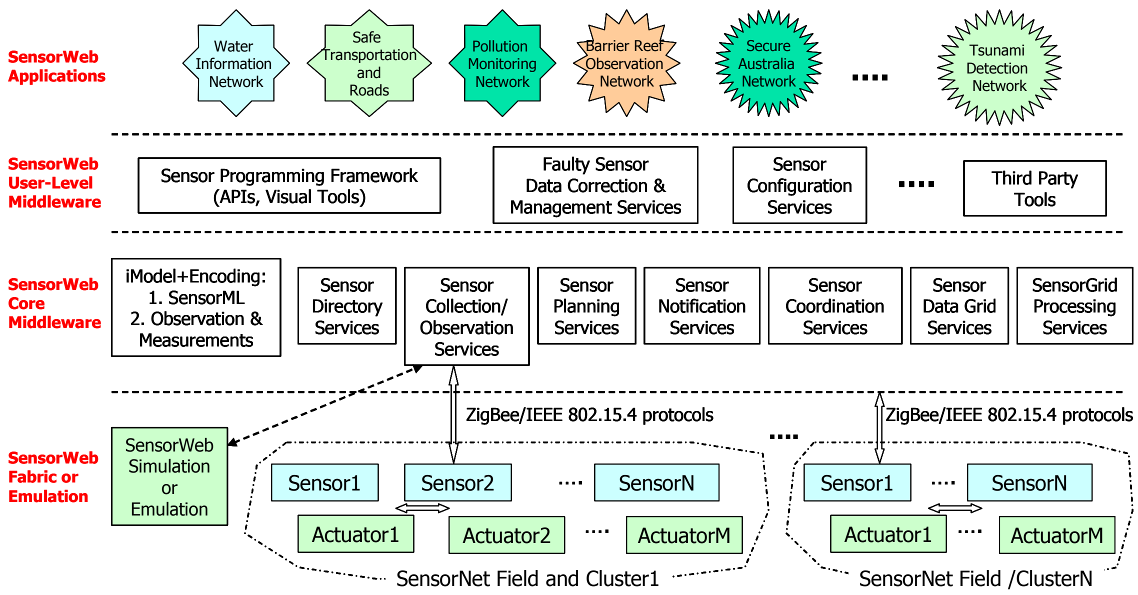

Figure 3 – Open SensorWeb Architecture.

At NICTA/Melbourne University, there is an effort to develop a SensorML standard compliant software infrastructure for providing Web Services based access to and management of sensors. The project defines Open SensorWeb Architecture (OSWA) that provides a complete standards compliant platform for integration of sensor networks with emerging distributed computing platforms such as Grids. This integration brings out dual benefits: (i) sensor networks can off-load heavy processing activities to the Grid and (ii) Grid-based sensor applications can provide advance services for smart-sensing by deploying scenario-specific operators at runtime. The various components of OSWA are shown in Figure 3. Fundamental services are provided by lower-level components whereas components at higher-level provide tools for creation of applications and management of life-cycle of data captured through sensor networks.

The OSWA-based platform provides a number of sensor and actuation services, such as:

- sensor notification, collection and observation,
- data collection, aggregation and archiving,
- sensor coordination and data processing,
- faulty sensor data correction & management, and
- sensor configuration and directory services.

The project primarily aims to provide (a) an interactive development environment, (b) an open and standards-compliant SensorWeb application services middleware, and (c) a coordination language to support the development of sensor applications for various domains, including water observation networks, safe road transportation management systems, and Tsunami detection network for the early warning systems.

4.2 Interconnection and networking

The communications and networking situations in sensor networks and grid computing are worlds apart. In sensor networks, the emphasis is on low power wireless communications which unfortunately has limited bandwidth and time-varying channel characteristics, while in grid computing, high-speed optical network interconnects are the norm. Thus, communications protocols for sensor-grids will have to be designed take into account this wide disparity.

ZigBee has emerged as one of the first standards-based low power wireless communications technologies for sensor networks, and a machine-to-machine (M2M) interface between ZigBee and GPRS has recently been announced, thus enabling sensor networks to be connected to the cellular network infrastructure. One other promising development is low-rate Ultra-Wide Band (UWB) wireless technology which has characteristics suitable for sensor networks, i.e. extremely low power consumption, reasonable communication range, and likely integration with UWB-based positioning technology.

4.3 Coordinated QoS in large distributed system

The timeliness and correctness of computations have been studied extensively in the real-time systems community, while performance guarantees in terms of delay, loss, jitter and throughput in communication networks and have also been studied extensively by the networking research community. We shall refer to these as application-level and network-level QoS, respectively.

A number of QoS control mechanisms such as scheduling, admission control, buffer management and traffic regulation or shaping have been developed to achieve application-level and network-level QoS. However, all these QoS mechanisms usually relate to a particular attribute such as delay or loss, or operate at a particular router or server in the system. In order to bring about the desired system-level outcome such as meeting an end-to-end computational and communication delay requirement, these QoS mechanisms need to be coordinated instead of operating independently.

There are several methods to achieve coordinated QoS. For example, coordinated QoS can be viewed as a multi-agent Markov Decision Process (MDP) problem which can be solved using online stochastic optimal control techniques such as reinforcement learning (RL) [3] or neuro-dynamic programming (NDP). Tham *et al* [4] have shown that this technique can achieve end-to-end QoS in a multi-domain DiffServ network with multiple resource managers in a cost effective manner.

4.4 Robust and scalable distributed algorithms

In Section 3, we described implementations of distributed information fusion and distributed autonomous decision-making algorithms on sensor-grids. Generally, it is more difficult to guarantee the optimality, correctness and convergence properties of distributed algorithms compared to their centralized versions, although the distributed versions are usually more appealing from an implementation point of view.

Apart from distributed information fusion and decision-making, distributed hierarchical target-tracking [5], distributed control and distributed optimization are other current research efforts on distributed algorithms which are relevant to sensor-grid computing.

4.5 Efficient querying and data consistency

Another key area in sensor-grid computing is efficient querying of real-time information in sensor networks from grid applications and querying of grid databases by sensor network programs. It is expected that databases will be distributed and replicated at a number of places in the sensor-grid architecture to facilitate efficient storage and retrieval. Hence, the usual challenges of ensuring data consistency in distributed caches and databases would be present, with the added complexity of having to deal with a large amount of possibly redundant real-time data from sensor networks.

5. Conclusion

In this article, we have provided an overview of the potential and challenges in sensor-grid computing. The success of the sensor-grid computing approach will depend on the ability of the sensor network and grid computing research communities to work together to ensure compatibility in the techniques and algorithms that will be developed in the future, as well as the ability of sensor-grid computing technology to provide real value to users and applications in the various industries and application scenarios mentioned in this article.

Acknowledgements
The authors gratefully acknowledge the contributions of Leslie Tan, Jean-Christophe Renaud, Daniel B. Yagan, Wai-Leong Yeow, Jimmy Kwan, Bohdan Durnota, Rao Kotagiri, Chris Leckie, Adrian Pearce, Shanika Karunasekera, and Krishna Nadiminti to the work described in this paper. Chen-Khong Tham is grateful to the Universitas 21 Network for awarding an Edward Clarence Dyason Fellowship at the University of Melbourne.

References

[1] The Gridbus Project. http://www.gridbus.org
[2] MF Duarte and YH Hu. "Optimal decision fusion for sensor network applications", First ACM Conference on Embedded Networked Sensor Systems 2003 (SenSys 2003)

[3] R Sutton and A Barto, "Reinforcement Learning: An Introduction", MIT Press, 1998
[4] CK Tham and Y Liu, "Minimizing Transmission Costs through Adaptive Marking in Differentiated Services Networks", in Springer Lecture Notes in Computer Science (LNCS) 2496, Oct 2002, pp. 237-249.
[5] WL Yeow, CK Tham and LWC Wong, "A Novel Target Movement Model and Efficient Tracking in Sensor Networks", in Proceedings of IEEE VTC2005-Spring, Sweden, 29 May-1 June 2005.
[6] OpenGeospatial Consortium, "Sensor Model Language (SensorML)", http://vast.nsstc.uah.edu/SensorML/

Biography

Dr. Chen-Khong Tham is an Associate Professor at the Department of Electrical and Computer Engineering (ECE) of the National University of Singapore (NUS). His research interests are in coordinated quality of service (QoS) management in wired and wireless computer networks and distributed systems, and distributed decision-making and machine learning. He is the supervisor of the Computer Networks and Distributed Systems (CNDS) Laboratory at the Department of ECE, NUS. He obtained his Ph.D. and M.A. degrees in Electrical and Information Sciences Engineering from the University of Cambridge, United Kingdom. He was awarded a 2004/05 Universitas 21 Fellowship at the University of Melbourne, Australia. He can be contacted by email - eletck@nus.edu.sg.

Dr. Rajkumar Buyya is a Senior Lecturer and the StorageTek Fellow of Grid Computing in the Department of Computer Science and Software Engineering at the University of Melbourne, Australia. He is also serving as the Director of the Grid Computing and Distributed Systems Laboratory. He has authored/co-authored over 100 papers and technical documents that include three books—Microprocessor x86 Programming, Mastering C++, and Design of PARAS Microkernel. He received B.E, M.E, and Ph.D. degrees from Mysore, Bangalore, and Monash Universities respectively. He was awarded Dharma Ratnakara Memorial Trust Gold Medal for academic excellence in Mysore University. He is currently serving as Co-Chair of the IEEE Technical Committee on Scalable Computing and Associate Editor of the Journal of Future Generation Computing Systems, Elsevier Press, Holland.