

GridStix: Supporting Flood Prediction using Embedded Hardware and Next Generation Grid Middleware

Danny Hughes, Phil Greenwood, Geoff Coulson, Gordon Blair
Computing Department
Lancaster University, UK
(danny/greenwop/geoff/gordon)@comp.lancs.ac.uk

Abstract

The cost of damage caused by flooding is directly related to the warning-time given before a flood occurs. Therefore, improving the coverage, accuracy and reliability of flood prediction systems is of great importance. This paper proposes a novel Grid-based approach to supporting flood prediction through the use of embedded sensor nodes equipped with wireless networking technology. These nodes implement a light-weight Grid capable of collecting and transmitting data gathered by flood sensors for off-site analysis. Additionally, the nodes are capable of performing basic on-site analysis that is used to inform system behavior. The proposed system utilizes Lancaster's next-generation component-based Grid middleware platform 'GridKit', which offers inherent support for reconfiguration, embedded devices and heterogeneous network technologies, making it ideally suited to this application.

1. Introduction

Flooding is a significant and growing problem in the UK, as was dramatically highlighted by the widespread flooding of autumn 2000 and the more recent January 2005 flooding of Carlisle. Both events had a significant impact on businesses, communities and transport infrastructure. The cost of the 2000 floods alone was estimated to be in the order of £1 billion and was described by the deputy Prime Minister, John Prescott, as "A wake-up call to the impacts of climate change" [1]. Following the damage wrought by the 2000 floods, major efforts have been initiated to improve the UK's flood readiness. These efforts focus on four key areas: improving flood defenses, supporting emergency response, raising public awareness and, importantly for this paper, improving flood warning and prediction systems. The cost of damage incurred during a flood correlates with two factors: the depth of flood water and the warning-time that is given before flooding occurs [2]. Despite

the importance of advance flood warning, flood monitoring systems in the UK remain inadequate to support the timely analysis of flood events.

Hydrologists have traditionally approached the problem of modeling and predicting floods by deploying sensing and logging equipment in areas susceptible to flooding. River profile data from these sensors, such as depth and rate of flow, is then either transmitted off-site or collected manually. This data is then used as the input to complex spatial flood prediction models [3] [4], which typically run on high-performance computers and Grid technologies. This data is also used as the basis for much simpler single-point models that can run on simpler devices. Unfortunately, these traditional approaches suffer from a number of drawbacks, which the proposed system seeks to address:

1. Sensor nodes are assigned rigid roles. Changing the function of a sensor node often requires on-site maintenance.
2. Sensor nodes are generally incapable of adapting their behavior to meet the demands of their environment.
3. Little, if any support is provided for dealing with node failure.
4. Nodes are unable to make intelligent local use of sensor data.

The remainder of this paper is structured as follows: section 2 discusses the requirements for supporting flood prediction. Section 3 describes the hardware and network technologies used. Section 4 provides a brief overview of the GridKit middleware. Section 5 details the proposed system functionality, section 6 discusses related work and finally section 7 describes the current state of the project and directions for future work.

2. Requirements of Advanced Flood Prediction

In order to address the limitations of current flood

prediction procedures identified above, and to improve upon existing systems in terms of performance, accuracy and reliability, certain requirements need to be met. The remainder of this section will describe and justify these.

The capability to *modify sensor node roles and adapt their behaviour* allows more efficient use to be made of available hardware resources (i.e. power, processor time and network resources). Additionally, by adapting system behaviour to match environmental conditions, the sensor network can focus on the most critical river properties, which may improve the accuracy of collected data. However, existing approaches do allow a small degree of adaptation; such as sensors simply transmitting readings more frequently when high water is detected.

The ability to deal with *node failure* is important in any distributed system; however, it is particularly critical in the case of remote environmental monitoring where sensors will often be deployed in remote areas, thus making maintenance a difficult and time consuming activity. It is therefore essential for the system to be capable of reconfiguring itself to handle node failure and, where possible, support off-site system maintenance. To cope with failures in existing approaches, results are simply interpolated to mask the sensor that failed.

Also, *intelligent processing* of the collected data has the potential to reduce the amount of data transmitted off-site, and thus remove a significant bottleneck [5] in the flood prediction process. Local processing could also be used to perform basic on-site flood prediction that can be used to trigger events such as activating additional sensors (and thus increasing the resolution of the monitored area) or providing warnings to local stakeholders.

Increasing the *integration of the sensor nodes into the Grid* has the potential to improve the timeliness of predictions in addition to creating a more autonomous and versatile flood prediction system. Existing approaches to integrating sensor networks with Grid technologies tend to treat the sensor networks as conceptually distinct entities from the Grid, which are computationally ‘dumb’ [5] [3]. However, emerging Grid middleware such as Lancaster’s GridKit [6] is capable of being deployed directly onto the sensor nodes themselves, allowing far greater autonomy, flexibility and fine-grained node management. This would be infeasible with the current generation of Grid technologies [17], which offer little support for embedded hardware, or the heterogeneous network technologies (e.g. Zigbee, Bluetooth, WiFi) that feature in modern sensor networks.

This paper presents a flood-prediction system which improves upon the data acquisition process currently used. Built using GridKit middleware [6], embedded computing hardware and ad-hoc wireless networking, the key to our approach is the notion that sensor nodes are promoted to full Grid entities, which we call ‘*GridStix*’, which are capable of forming autonomous, adaptable and resilient sensor Grids.

3. Hardware Platform

Our primary hardware platform consists of solar powered embedded computing devices [11] running the GridKit middleware [6]. These units maintain their own wireless network infrastructure and can be deployed independently of any local infrastructure. We refer to these self-contained grid entities as ‘GridStix’.

Each GridStix node is built on the ‘Gumstix’ embedded computing platform [11] (so named as the board is roughly the same size as a pack of gum). Despite their small size, Gumstix feature an Intel XScale CPU running at up to 400MHz and come equipped with 64MB of RAM, 16MB of flash memory and Bluetooth radio. As such, they are vastly more powerful than highly-embedded platforms such as Berkeley Motes [12], or Equator Smart-Its [13], having roughly the same processing power as a 233MHz Pentium class CPU. Gumstix are also capable of running a standard Linux kernel, making them compatible with the GridKit middleware ‘out of the box’. Gumstix can also be expanded using a wide range of off-the-shelf Compact Flash and USB devices, making them a flexible platform for supporting heterogeneous sensor and network technology. As might be expected, their computational power and flexibility comes at the expense of power consumption. During typical operation a Gumstix consumes up to 1 watt of power with a maximum theoretical power draw of up to 3 watts [14]. This is not a significant problem, however, as it is quite feasible for these power-requirements to be met using a mid-sized solar panel and high-capacity battery.

Multiple sensors may be attached to each GridStix node. The primary sensor devices used in this system are *pressure sensors*, which are used to gauge the depth of flood-water. Pressure sensors are low-cost, durable sensors and can be interfaced to the GridStix via a serial connection. The system will also make use of a smaller number of *ultrasound flow sensors*. These are used to gauge the average speed of flood water and, like the depth sensors, connect to the GridStix via a serial connection.

Unfortunately, the cost of ultrasound flow sensors is currently prohibitive for large-scale deployment and therefore a number of *image-based flow sensors* will also be used. Image-based flow measurement calculates water flow-rate using the velocity of particles on the water surface (described in more detail in section 5.2.2).

An on-site networking infrastructure is also required to disseminate information between the GridStix. Unfortunately, it cannot be assumed that any network infrastructure will exist at a flood site, which by their nature tend not to be in developed areas. It is essential therefore that the GridStix maintain their own communications infrastructure. Typically, wired deployment at flood sites is impractical and so GridStix nodes must maintain their own ad-hoc wireless communications infrastructure. We envisage that this local network will include a range of wireless network technologies such as WiFi and Bluetooth. Additionally, a sub-set of nodes will act as gateways to the wider Grid and so will be equipped with GPRS modems to allow data to be sent off-site. The exact topology and network type used may be configured depending upon: the changing requirements of the system, current environmental conditions and/or the state of the GridStix themselves (e.g. battery charge status). As detailed in the next section, the GridKit infrastructure is ideally suited to support this kind of adaptive behaviour.

4. The GridKit Middleware Platform

Our proposed approach is to build our software on top of the GridKit [6] middleware platform. This provides the key functionality required to support Grid applications: service binding, resource discovery, resource management and security. It is based upon the language independent OpenCOM¹ component model [7], with each area of Grid functionality being implemented as an independent component framework. As GridKit is software-component based, it is inherently configurable and extensible. By selectively combining component frameworks, either rich and complex, or basic and simple, Grid support can be created. In particular, minimal deployments can be developed that contain just the minimum functionality necessary to perform a particular task. By removing unnecessary components in this way, and thus reducing the associated computation and storage requirements, it is possible for GridKit to be deployed on embedded platforms.

¹ Available for download from: www.comp.lancs.ac.uk/computing/research/mpg/reflection/download.php

Of particular relevance to the proposed system is the *overlay component framework*, which is shown in Figure 1. The overlay component framework is GridKit's core networking abstraction and provides a base for building application level networks (or overlays), which are typically used to implement new networking services not provided by the underlying network and to provide application-level functionality that is not within the scope of the underlying network. Classic examples include: resource discovery, content distribution and event notification services. The use of overlay abstractions allow multiple overlays to be easily used side-by-side and introduces the possibility of reducing system overhead still further by sharing components, particularly state between different overlays. As their name suggests, overlay networks may also be layered to combine properties of different overlays. For example, structured overlay networks [8] are an efficient and resilient base for object lookup, but do not support complex search, while unstructured networks [9] provide inherent support for resource location. Using the overlay component framework, it is possible to build an application level network capable of supporting both efficient object location and complex searches, similar to [10] (see figure 2). GridKit's support for building a wide range of diverse yet interoperable overlays, allows the networking approach used in our system to be tailored to support the heterogeneous network technologies and application requirements that the system must support.

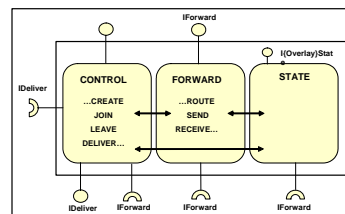


Figure 1: Individual Overlay Plug-in

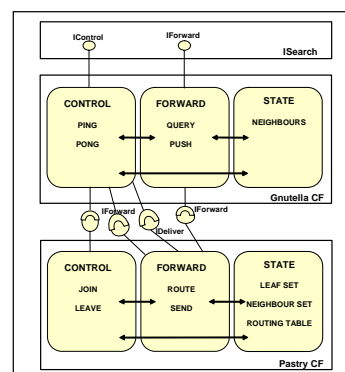


Figure 2: An Open Overlays Configuration

The ability to customise the system behaviour also includes run-time configuration provided by OpenCOM. OpenCOM implements a series of

reflective facilities to allow run-time changes to be made whilst maintaining a low-overhead. The *architecture meta-model* allows the programmer to introspect and manipulate the topological graph of components. The *interface meta-model* allows the discovery of the interfaces supported by a particular component and also to dynamically invoke these discovered interfaces. Finally, the *interception meta-model* can be used to trap invocations and transparently execute before or after these invocations.

The combination of the OpenCOM component model with overlays creates components known as ‘*open overlays*’. These open overlays allow Grid applications to be easily developed that are highly heterogeneous both in terms of the end-system and the underlying network infrastructure with the ability to perform dynamic reconfiguration in response to environmental changes. Clearly the GridStix deployed to collect river data will be highly susceptible to environmental change such as coping with node failure, switching to different network types and altering the network configuration depending on available resources such as battery power.

5. SYSTEM OVERVIEW

The core functionality of the proposed system is threefold.

1. To act as a resilient and adaptive sensor network, which collects information from a range of sensors and transmits the collated data off-site where it is used to perform spatial flood prediction.
2. To act as a light-weight computational Grid, which performs local point-based flood prediction and the calculation of flow rates from images.
3. To provide flood warnings to local stakeholders, such as residents and businesses.

Each of these system roles is discussed in more detail in the following section.

5.1 Building an Adaptive and Resilient Sensor Network

Although GridStix are more computationally rich than previous devices deployed in similar scenarios, it is still not possible for all analysis to be performed on-site. In particular, highly complex spatial models can only be executed on high performance cluster computers. For this reason, one of the system’s core roles is to act as a sensor network which collates data and transmits it off-site. As flood-sites are often isolated and difficult to access, it is particularly important that the need for manual intervention be minimized. To this end, the sensor grid must be

capable of adapting to changes in the local environment and be resilient to node failure.

As described in Section 3, GridStix are capable of maintaining both Bluetooth and WiFi network infrastructures. This is desirable as the two networks have quite different properties and one could be selected over the other to reflect changes or failures in the network. For example, a set of nodes in close proximity may be disseminating data between themselves in a multi-hop fashion via a low power consumption Bluetooth network. However, a critically placed node in this network could fail, resulting in network segmentation. To allow the sensor network to smoothly continue operating, a sub-set of the nodes should then switch to 802.11b (due to the improved range) to repair the network segmentation.

As previously described, GridKit [6] provides rich support for the construction of application-level networks through the concept of open overlays, as introduced in section 4. Switching between application-level overlay networks allows for more fine-grained network management in response to changes in the system. For example, consider GridStix nodes equipped with GPRS modems which act as ‘gateways’ transmitting data off-site. Typically, data will be sent to these gateway nodes from other sensing nodes using some form of spanning tree. Spanning trees may be optimized based upon performance, which tends to result in high bandwidth consumption; or alternatively they could be optimized to minimize bandwidth consumption, which tends to result in lower performance [16]. During normal operation, a bandwidth-optimized spanning tree may be used to transmit data to the gateway nodes and minimize power consumption. However, when flooding becomes imminent and flood-modeling becomes time-critical, the GridStix may switch to using a performance-optimized spanning tree.

Application-level overlays can also be used in response to failure. For example, under normal operation, GridStix nodes will create some form of optimized spanning tree to transmit sensor data to the closest gateway node, thus minimizing power consumption and transmission time. However, GridStix nodes may also participate in a broadcast based resource-discovery network, implemented through a second application-level network such as [9]. If the active Gateway node fails, this resource discovery network is used to locate another gateway and initiate the re-building of the spanning tree.

5.2 Supporting Local Computation

In our system, local distributed computation is used for two main purposes. Firstly, the GridStix perform local point-based flood prediction, which is used to modify system behavior. Secondly local

distributed computation is used to support image-based flow prediction. The two aspects are discussed in the following two sub-sections.

5.2.1 Local Point-Based Flood Prediction. Point based flood prediction requires sensor data from a minimum of one upstream node and also from the locally connected sensors (we assume that the sensors are deployed along a river). Using this input data and site-specific location parameters, point predictions can be performed in a matter of seconds on a GridStix. Each GridStix engaged in performing point predictions will simultaneously be a source of data for a downstream node whilst being dependent on data from an upstream node (see Figure 3). The specific dependencies between nodes will thus determine the structure of the application-level network used to disseminate data. In cases where interdependent nodes are out of range, for example due to node failure, secondary networks such as GPRS may be required to ensure communication.

The locally computed flood prediction data can be used to trigger adaptation of the way the whole network behaves, such as switching on more nodes to improve accuracy or switching to a faster, but more power hungry network type. Point-predictions could also be used to generate flood warnings for local stakeholders, either through alerts generated on locally connected audio and video hardware or by text-message warnings distributed by the gate-way nodes. Any node predicting a potential flooding would distribute a message on a broadcast overlay such as [9], which would be used as a prompt by each receiving node to adapt their behavior.

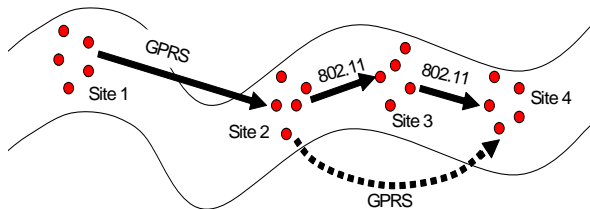


Figure 3: Local Point-based Flood Prediction

Point-prediction models have already been implemented and widely used by the Environmental Science Department at Lancaster University. These models are currently implemented in Matlab; however, Matlab code can be easily converted, using tools packaged with the Matlab execution environment, into C code which can be easily integrated with the GridKit middleware.

5.2.2 Image-based Flow Measurement. In addition to ultrasound flow sensors and depth sensors, our deployment will also employ a small number of cameras which will also be used to estimate the flow rate. This approach has a number of advantages over ultrasound flow sensors. As the cameras are mounted

out of the water, deployment is significantly easier and the cameras are significantly cheaper.

The image-based flow measurement process works as follows. A camera is deployed in a fixed position on site, such as on a bridge or other structure of sufficient height overlooking the stretch of river to be analysed. The camera is then used to take a series of still images over a set period of time. From these images ‘tracer’ particles are identified on the surface of the water. These can be naturally occurring elements such as foam, boils, ripples or debris. Alternatively, artificial tracer particles can be placed in the river. Interrogation spots, which consist of blocks of contiguous pixels, are selected in an attempt to find the tracer particle in subsequent images. When the tracer particle is located, an estimate of the surface velocity can be made based on the original position of the tracer particle and its position in subsequent frames in relation to known control points present in all the images. This is repeated a number of times to produce an average surface velocity. As with the local point-prediction described above, image based flow prediction is dependent on depth information from up-stream nodes, as changes in depth cause a change in the angle between the camera lens and the water surface, affecting velocity computations.

The image analysis process is very computationally intensive, which together with the easily divisible dataset makes image-based flow prediction a good candidate for the processing on a Grid infrastructure. Unfortunately, due to the large number of high quality images required to successfully identify tracer particles, the data generated by the camera would be too large to be continually transmitted off-site using GPRS technology. An alternative approach would be to perform velocity estimates on-site; however, the GridStix do not have the necessary computational power to process this data continually, particularly given the multiple roles that each GridStix must also fulfill (sensor node, gateway, point prediction host etc.). A promising approach is for the GridStix nodes to perform some basic preliminary analysis on a subset of the images. From this, an estimate of the flow rate can be made and used to trigger off-site transmissions of the complete image set for the time period of interest. This would reduce the amount of data transmitted by the gateway nodes while minimizing the computational load on the GridStix.

Support for image-based flow measurement is a particularly good example of the power of the GridStix approach. Image-based flow analysis could not be performed on a traditional ‘dumb’ sensor network, without the provision of costly networking infrastructure (such as broadband satellite links) as the raw sensor data would easily overwhelm a GSM or GPRS connections.

6. Related Work

There are a number of current projects that relate to the work presented in this paper. The Floodnet [2] project shares many of the goals of our work, i.e. to improve flood warning times by deploying an intelligent grid-connected sensor network and it uses a similar hardware platform to the GridStix - XScale-based embedded computers and WiFi networking. However, while Floodnet also recognizes the migration of Grid services onto sensor nodes as an open research question, it still treats the sensor network as a conceptually separate entity from the Grid. As previously discussed, this limits the flexibility and adaptability of the resulting system. The Equator project [18] has also looked at ways of integrating sensor data with the grid, however, like Floodnet, their proxy based approaches to integrating sensors with the grid, effectively treat sensor nodes as dumb entities, which is increasingly at odds with the growing power and versatility of sensor nodes, such as those used in our work and in Floodnet [2]. In each of the above cases, and compared to the general body of work on integrating sensor data into the Grid, the primary difference with our approach is the elevation of the sensor nodes themselves to become integrated parts of the Grid. The RUNES project [15] focuses on deploying reconfigurable technology related to the GridKit middleware on embedded hardware. The primary focus is on porting to highly embedded hardware which is connected by short-range, low-bandwidth network technologies, such as Berkeley 'Mica' Motes [12]. These networked embedded systems can then be used to build intelligent and adaptive Grid-connected sensor networks. Our work differs from RUNES in that we seek, where possible to move traditional Grid functions onto the sensor nodes themselves and to elevate these nodes to the role of first-class Grid entities.

7. Status and Future Work

Currently we are focused on porting the GridKit middleware to the Gumstix and interfacing these with sensor and networking hardware. It is this process that is in the early stages of development. However, we anticipate rapid progress with this stage of the project due to the heterogeneous nature of GridKit. Future work in the near-term will focus upon implementing point-based flood prediction and basic sensor network functionality using the GridStix and simple depth sensors. This will be augmented with the other functionality described in the document throughout the coming year.

8. References

- [1] The Environment Agency "Lessons Learned, Autumn 2000 Floods", available online: <http://www.environment-agency.gov.uk/commondata/acrobat/126637>
- [2] D. DeRoure, "Improving Flood Warning times using Pervasive and Grid Computing", submitted to quarterly of Royal Academy of Engineering, UK.
- [3] F. Pappenberger, K. Beven, N. Hunter et al., "Cascading model uncertainty from medium range weather forecasts (10 days) through a rainfall-runoff model to flood inundation predictions within the European Flood Forecasting System (EFFS)", published in *Hydrology and Earth System Science*, 9(4), pp381-393, 2005.
- [4] F. Pappenberger, P. Matgen and K. Beven, "The influence of rating curve and structural uncertainty on flood inundation predictions." Published in *Advances in Water Resources*. (in press) 2006,
- [5] T. Rodden, C. Greenhalgh, D. DeRoure, A. Friday, "Extending GT to Support Remote Medical Monitoring," in the proceedings of UK e-Science All Hands Meeting, Nottingham, UK, September 2005.
- [6] G. Coulson, P. Grace, G. Blair et al, "Open Overlay Support for the Divergent Grid", in the proceedings of the UK E-Science All Hands Meeting, Nottingham, UK, September 2005.
- [7] G. Coulson, G. Blair, M. Clark et al "The Design of a Highly Configurable and Reconfigurable Middleware Platform", in the *ACM Distributed Computing Journal*, Vol. 15, No 2, pp109-126, April 2002
- [8] S. Ratnasamy, P. Francis, M. Handley et al, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", in the proceedings of (SIGCOMM), ACM, San Diego, USA, August 2001.
- [9] Limewire "The Gnutella Protocol Specification v0.4", website: www.limewire.com/developer/gnutella_protocol_0.4.pdf
- [10] M. Castro, M. Costa, A. Rowstron "Should be build Gnutella on a Structured Overlay", published in the proceedings of the Second Workshop on Hot Topics published in *Networks (HotNets-II)*. Cambridge, MA USA, November 2003.
- [11] Waysmall Computers "Gumstix Embedded Computing Platform Specifications", website: <http://gumstix.com/spexboards.html>.
- [12] Crossbow "Mica Motes and Sensors", website: <http://www.xbow.com/>
- [13] The Smart-Its Project. "Smart-Its Artifacts", website: <http://www.smart-its.org/artefacts/artefacts.html>.
- [14] The Gumstix Wiki, "Heat and Power", website: <http://www.gumstix.org/>
- [15] P. Costa, G. Coulson, C. Mascolo et al. "The RUNES Middleware: A Reconfigurable Component-based Approach to Networked Embedded Systems", in the proceedings of (PIMRC05), IEEE, Berlin, Germany, September 2005.
- [16] I. Akyildiz, s. Weilian Su, Y. Sankarasubramaniam, e. Cayirci, "A Survey of Sensor Networks", published in *Communications Magazine*, IEEE, Volume 40, Issue 8, pp102 - 114, August 2002.
- [17] S. Tuecke, K. Czajkowski, I. Foster et al. "Open Grid Services Infrastructure (OGSI) Version 1.0".
- [18] J. Humble, et al, "A Generic Architecture for Sensor Data Integration with the Grid." In the proceedings of (SAG 2004), Springer-Verlag, Beijing, China, May 2005.