

A Typology of Real-Time Parallel Geoprocessing for the Sensor Web Era

Aengus McCullough^{1,1}, Stuart Barr¹, Philip James¹

¹School of Civil Engineering & Geosciences, Newcastle University, Newcastle-upon-Tyne,
NE1 7RU

{aengus.mccullough, philip.james, s.l.barr} @ncl.ac.uk

Keywords: sensor web, parallel geoprocessing, grid computing, typology

1 Introduction

The rise of digital sensors and the Sensor Web is expected to have wide reaching implications for the monitoring of the physical and human world [1] and has already resulted in an explosion in the volume and availability of spatially referenced data pertaining to our surroundings. While this deluge of easily accessible near real-time data brings numerous opportunities, it also presents a significant challenge in terms of geoprocessing. Existing geoprocessing systems must be adapted to meet the requirements of this new era of spatial data infrastructure.

Although real-time geoprocessing systems have existed for some time in fields such as environmental monitoring, they have usually been part of a stove-piped system in which the geoprocessing component was specifically engineered for the given application [2]. In today's world of service orientation, geoprocessing components are often developed as services that can be swapped in and out of systems with ease. Web service standards defined by the Open Geospatial Consortium (OGC) have become widely adopted. The OGC Web Processing Service (WPS) defines a uniform interface to encapsulate heterogeneous geoprocessing functionality [3]; by chaining OGC data and processing services geoprocessing workflows can be rapidly composed. As a result, we have come to expect generic geoprocessing services to be available that meet our requirements.

However, the requirements of real-time monitoring and prediction scenarios differ significantly from offline geoprocessing in terms of usage patterns, computational characteristics and data processing methodologies. Real-time systems must often process continuous jobs of an unknown size or duration [4]. They may be required to work to a hard real-time deadline, or to keep pace with the rate of data arrival [5]. Additionally, they must be capable of operating on data streams as well as static datasets, and in some cases to perform complex event or pattern detection [6]. Furthermore, data acquired from sensors is often unreliable so geoprocessing systems need to be robust to corrupt and missing observations [7]. For these reasons generic geoprocessing services designed for offline analysis are often unsuited to operating on near real-time sensor data.

2 Geoprocessing in Real-Time

To assist in the definition of a typology of real-time geoprocessing it is helpful to consider the various attempts that have been made to represent the dynamic nature of real world phenomena within GIS. Worboys and Duckham [8] outline the following four stages in this progression:

- | | |
|----------------------------------|---|
| 1. Static | A single static view of the world. |
| 2. Snapshot | Dynamic phenomena are represented as a collection of time-stamped states. |
| 3. Object lifeline | The lifecycle of objects including creation and destruction are recognised. |
| 4. Events, actions and processes | Continuous and instantaneous phenomena can be modelled. |

Towards the events, actions and processes end of this spectrum the complexities inherent in modelling the real-world in time and space become apparent. Whereas events occur at a fixed instant in time, processes occur over a time interval. This disparity between instantaneous and interval representations of spatial phenomena is formalised by Grenon and Smith [9] with their SNAP and SPAN ontology. In terms of geoprocessing systems, 'real-time' implies we are dealing with temporal representations at the snapshot level or above. As such, real-time geoprocessing covers a range of temporal scenarios. In the typology defined here we have simplified real-time geoprocessing into two categories: snapshot and stream geoprocessing.

At the simplest level an operation may involve the processing of a fixed snapshot of recently collected spatial data. Snapshot geoprocessing is comparable to static geoprocessing in that input and output data are discrete and the operation has a finite lifetime. However, snapshot geoprocessing operations may form part of a real-time monitoring or prediction system, and may thus be required to produce results within a fixed time-frame. Furthermore, the snapshot paradigm is concerned with processing data acquired from a potentially unreliable sensor source rather than a consistent input dataset.

The processing of a series of observations representing a time interval requires a radically different approach to static or snapshot geoprocessing and draws on techniques from the field of Data Stream Processing (DSP). In DSP terms a data stream is a potentially unbounded sequence of tuple timestamp pairs. DSP systems can be considered an alternative to database technology for coping with streams of data as opposed to persistent datasets [10]. In reality, data streams are comprised of a series of discrete observations although they are finely spaced enough to interpolate between values.

3 Distributed Computing for Real-Time Geoprocessing

Both snapshot and stream based geoprocessing can be performed in parallel on distributed systems such as grid and cloud computing in order to improve performance and scalability. Processing stream data in parallel is relatively

straightforward; either one sensor stream can be assigned to each processor, or pipeline parallelism can be exploited as the data is already divided into an ordered sequence. Alternatively, for stream based processing operations that carry a high time complexity, data stream partitioning can be used to divide the workload amongst several processors. Furthermore, the small but relentless torrent of data associated with the stream paradigm is easily managed in a distributed network environment whereas larger data files are more cumbersome to work with as they require longer transfer times and can often not be read until the transfer is complete.

The appropriate parallelisation technique for processing snapshot data is dependant on the granularity of the geoprocessing operation. Granularity refers to the level of synchronisation required between sub-tasks that are executed in parallel [11]. At one end of the scale are tasks referred to as coarse-grained which require virtually no synchronisation. The task is simply split into sub-tasks which are processed independently; once complete the results from each sub-task are merged together. Coarse-grained tasks are well suited to grid and cloud computing as they can easily be processed using a standard cluster of commodity processors [12]. At the other end of the scale are fine-grained tasks that require significant communication between sub-tasks. Because of the high degree of communication required it is recommended to perform fine-grained tasks in close proximity to the data [13]. Two approaches are commonly taken; processing can either be performed at the database, or a dedicated High Performance Computing cluster with high bandwidth low-latency interconnections can be used as they are specifically designed to reduce latency in communication between sub-tasks.

Because coarse and fine grained snapshot geoprocessing operations are each suited to such different distributed computing architectures we have disaggregated the snapshot geoprocessing category in our typology into two subcategories: coarse-grained snapshot and fine-grained snapshot. The characteristics of each real-time geoprocessing category in our typology are displayed in Table 1.

Table 1 Characteristics of Real-Time Geoprocessing Types

| Characteristic | Stream | Coarse-grained Snapshot | Fine-grained Snapshot |
|---------------------------------|----------------|--------------------------------|------------------------------|
| Regularity of invocation | regular | regular or irregular | regular or irregular |
| Trigger | time | time or event | time or event |
| Temporal Representation | interval | instant | instant |
| Granularity | coarse or fine | coarse | fine |

4 System Implementations

We have implemented end to end web service based systems representative of each of the categories outlined in Table 1 using application scenarios from the fields of road traffic monitoring and satellite-sensed image processing. To exemplify a stream

geoprocessing system we developed a near real-time map-matching system that determines the road segment a vehicle is travelling on from its GPS derived position and orientation. The system uses a 52 North WPS configured to submit processing jobs to the UK National Grid Service (NGS) infrastructure through an Open Grid Services Architecture compliant endpoint. Map-matching is initiated for each vehicle by sending an *Execute* request to the WPS which invokes the algorithm on an NGS compute node where it runs continuously. The algorithm retrieves live vehicle position and orientation information from a Sensor Observation Service (SOS) and performs matching using road network features obtained from a Web Feature Service.

Another stream geoprocessing system was developed to determine travel-time estimates for road segments throughout a city. This system uses the results produced by the map-matching system, i.e. time-stamped lists of road segment identifiers representing vehicle routes, to update a road network database containing road segment traversal times. Map-matching results were forwarded from an SOS to a Sensor Event Service (SES) which detected each vehicle movement from one road segment to the next, and sent a notification to the database to update the travel-time weighting of the traversed road segment accordingly.

To exemplify fine-grained snapshot geoprocessing we developed a least cost path routing system that uses a road network database containing regularly updated live travel-time information, to determine the fastest route between any two given points in a city's road network. As the shortest path algorithm is fine-grained and not easily sub divisible for parallel processing it was opted to perform this geoprocessing operation close to the data source. Consequently this operation was performed at the database to eliminate the need for costly network data transfers.

We also developed a coarse-grained snapshot geoprocessing system that used Hadoop MapReduce to perform a convolution filter based image processing algorithm on the Amazon Cloud. This geoprocessing operation was found to be easily sub-divisible and thus highly suited to grid and cloud processing architectures. The satellite image was subdivided into a set of kernel windows, each of which was sent to a different node to be processed during the map stage. In the reduce stage the outputs from the map stage were merged together to form the resulting output image.

5 Discussion and Conclusion

The implemented systems enabled us to identify some potential improvements to existing OGC specifications with regards to real-time geoprocessing. Firstly, we concluded that the WPS interface requires modification to run stream geoprocessing tasks because in its current state it is assumed that each processing task has a finite lifetime. In order to facilitate the management of continuous compute jobs we modified the WPS interface by adding a *StopExecuting* operation. Secondly, we suggest that the SOS and SES be integrated into a unified interface. We found that latency was introduced by polling the SOS and forwarding observations to the SES. Integration would enable real-time observations to be subjected to push-based filtering and archived for pull-based retrieval simultaneously. Thirdly, we found that

standard web service tools were unable to parse OGC schema, suggesting that well documented OGC schema compatibility issues [14] have yet to be resolved.

We found the typology defined here of practical value in the design and implementation of our systems. Our implementations reinforced the validity of our typology, although the stream processing category was found to be over simplistic and we therefore recommend a further subdivision of this category into atomic transformations, stream dependent transformations and event correlation. It is our belief that this taxonomy will be of use in the future development of toolkits to facilitate real-time geoprocessing.

References

1. Gross, N.: 21 Ideas for the 21st Century. Business Week Online, vol. 30 (1999)
2. Resch, B., Blaschke, T., Mittlboeck, M.: Live Geography - Interoperable Geo-Sensor Webs Facilitating the Vision of Digital Earth. *International Journal on Advances in Networks and Services*. 3, 323--332 (2011)
3. Schut, P.: OpenGIS Web Processing Service version 1.0.0. OpenGIS Standard OGC 05-007r7. Open Geospatial Consortium (2007)
4. Madden, S.: Query Processing for Streaming Sensor Data. PhD Qualifying Exam Proposal. Computer Science Division, University of Berkeley, California, USA (2002)
5. Golab, L., Ozsu, M.T.: Issues in data stream management. *ACM SIGMOD Record* 32, 5--14 (2003)
6. Resch, B., Mittlboeck, M., Girardin, F., Britter, R., Ratti, C.: Real-time Geo-awareness - Sensor Data Integration for Environmental Monitoring in the City. *International Conference on Advanced Geographic Information Systems and Web Services*. IEEE, Cancun, Mexico (2009)
7. Balazinska, M., Deshpande, A., Franklin, M.J., Gibbons, P.B., Gray, J., Nath, S., Hansen, M., Liebhold, M., Szalay, A., Tao, V.: Data Management in the Worldwide Sensor Web. *IEEE Pervasive Computing* 6, 30-- 40 (2007)
8. Worboys, M., Duckham, M.: *GIS: A Computing Perspective*. CRC Press, Boca Raton (2004)
9. Grenon, P., Smith, B.: SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition and Computation* 5, 69--104 (2004)
10. Babu, S., Widom, J.: Continuous Queries over Data Streams. *ACM SIGMOD Record* 30, 109--120 (2001)
11. Abbas, A.: *Grid Computing: A Practical Guide to Technology and Applications*. Charles River Media (2004)
12. Foster, I.: *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison Wesley (1995)
13. Müller, M., Bernard, L., Brauner, J.: Moving Code in Spatial Data Infrastructures - Web Service Based Deployment of Geoprocessing Algorithms. *Transactions in GIS* 14, 101--118 (2010)
14. Sonnet, J., Savage, C.: OWS 1.2 SOAP Experiment Report OGC03-014. Open Geospatial Consortium Inc. (2003)