Nexus: developing a schema to contain disparate traffic data types for parallel distributed processing

Shannon. Fehlmann[1], Judd. Kelly[1], Josh. Mitchell[1], Nikolaos Vogiatzis[2]
[1] School of Computer and Information Science, The University of South Australia
[2] Transport Systems Centre, The University of South Australia

1 Introduction

The increasing signalisation and automation of road traffic networks presents an opportunity. Traffic network control infrastructure is now adaptively configurable to account for factors such as varying saturation in different portions of the network and even external factors such as environment. An example of this adaptive technology is Sydney Coordinate Adaptive Traffic System (SCATS) controlled signalised intersections.

A traffic network configuration strategy is only as good as the data that informs the strategy. Consider a small urban centre which could reasonably produce several terabytes of traffic data every year from signalised intersections alone. Naturally, the more of this data that is included in any traffic data analysis, the more robust the analysis is likely to be.

Traffic Engineers require powerful data processing tools to perform meaningful analysis over these large collections of traffic data. Entry level data processing platforms, while inexpensive, are computationally impotent in such cases and off the shelf enterprise solutions, while capable impose an unacceptable cost burden both in initial outlay and maintenance.

To address this issue the University of South Australia Transport Systems Centre (TSC) has adopted a specialised data processing architecture termed the 'shared nothing' architecture (Stonebraker 1986). This paper presents an experimental schema 'Nexus II'. Building on earlier work, (Vogiatzis et al. 2005) Nexus II integrates data types required for comprehensive traffic analysis and leverages the inherent extensibility of the shared nothing architecture and price-per-performance of commodity hardware.

2 Background

There is an ongoing integration effort with regard to traffic control and traffic information systems, its goal is improved traffic network efficiency. (Ikeda et al. 2004) Data processing infrastructure able to manage large collections of traffic data is required and the obvious solution is a database.

Databases are often used to address computationally challenging data processing tasks. A suitable database technology for traffic data processing is a decision support tool called a data warehouse. (Gardner 1998) The processing burden inherent to data warehousing operations is often addressed by introducing parallel processing or parallelism.

One way parallelism can be achieved by distributing the data warehouse over several computers which are connected by a standard network. Together the computers function as a single unified system called a processing cluster with processing load shared by each computer. A single query can then be partitioned and executed in parallel on several computers at once. Ideally this means a data processing operation is completed in significantly less time compared to one executed on a centralised data warehouse.

*A portion of this research was supported by ExtenDB Inc (USA) with the provision of the ExtenDB Parallel Server software and associated technical support at no cost to the Transport Systems Centre. Additionally, a portion of this research was supported by the Department of Transport, Energy and Infrastructure with the provision of the ACTS/SCATS 5 minute volume Detector Store data used in the various experiments and in the analysis of the study area.

Data from all 9 regions of the Adelaide Coordinated Traffic Systems (ACTS) encompassing all traffic signals within the metropolitan Adelaide region excluding the Adelaide CBD over a period of 4 months between May and August 2005 inclusive was analysed. The study area used for the transport analysis is in Figure 0.

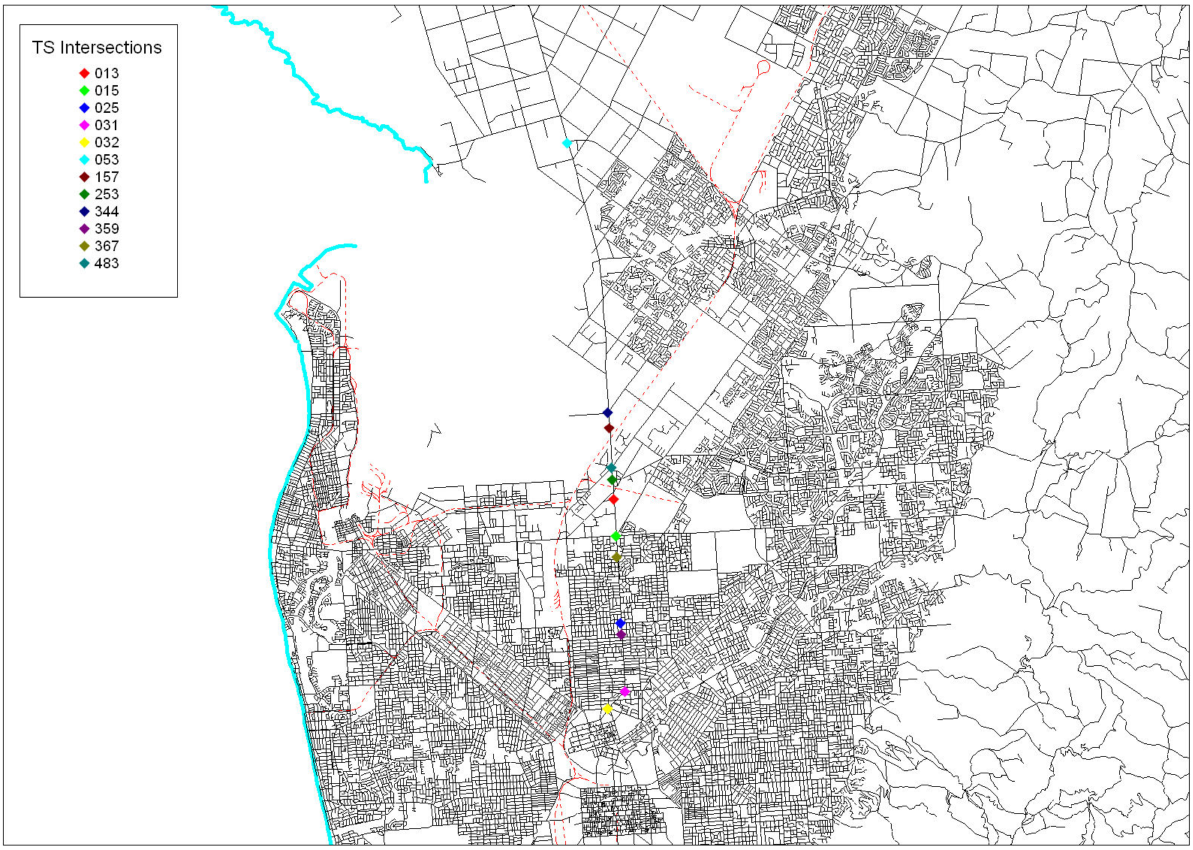


Figure 0: The study area used during the analysis of the performance of the NEXUS II system.

The reason for this was two-fold: first the effectiveness of applying parallelism in a data warehousing environment was tested; and second its direct applicability to improving the tools available for transportation research was assessed (the results of which are in an associated paper by Vogiatzis and Taylor (2006)).

Typically, analysis using large databases is time consuming and difficult; thus statistical techniques have been employed to simplify the task; however this leads to a loss of flexibility and data generality. The ability to analyse at will very large datasets in a short space of time with the elasticity to allow analysis on more than a 'day-by-day' limited variable basis and that allows for the cross-analysis of disparate data sources over varying temporal and geographic ranges will allow transport engineers to spend less time worrying about the amount and type of data they have and more on what they can learn from the data available. This will lead to viewing data with varying resolutions and, hopefully, to a more complete understanding of the transportation system in a holistic sense.

Furthermore, using some of the techniques expressed in this paper will allow future Urban Traffic Control systems to use a greater number of different types of sensory data across a wide area network providing a high level analysis and improvements in traffic flow, reductions in travel time, thus emissions and improvements in freight scheduling, just to name a few.

3 Traffic Data

Data applicable to traffic control is diverse. Several source data types are considered relevant for traffic analysis work performed by the TSC:

- data available from signalised intersections (in this case SCATS generated)
- emissions data gathered during test vehicle traversals of the road network
- geospatial data describing the road network

These data collections are large, two challenges are presented: first, to design a data schema which integrates these diverse data types and second, implement the schema on an extensible architecture. An extensible architecture is important to effectively scale the system to meet high initial computational needs and to ensure future capability as available data grows.

Each of these source data types is initially in a format unsuitable for immediate insertion into a database. For any meaningful processing to be performed on the data it needs to first be parsed into a compatible format and then loaded into the schema. Nexus II development work to date has shown that conversion from these proprietary formats (such as the SCATS binary format) to a format suitable for relational database processing leads to a size increase of between 5 and 10 times. Data collections reaching terabyte scales are easily accumulated.

Data of interest available from signalised intersections includes traffic flow rates, road saturation, state data about the traffic signals and decision support data derived by the signal control device.

Available emissions data is sourced from a test vehicle that records its current location with a GPS unit, records vehicle state (throttle position, speed, rpm etc.) and records relevant emissions readings (CO, CO_2, NOX, etc.).

An accurate geospatial representation of the road network has been obtained by the TSC. The Australian road network is described as a collection of edges, each with known geospatial location.

If we have this data available, why not try to add value as done in earlier work? (Vogiatzis 2003) This new work introduces the following capabilities to support interlinked processing of the above data types, namely:

- a geospatial description of available and actual road network traversals
- a way to record characteristics of interest of a road network
- definition of signalised intersection location
- a facility for traffic data and project data administration

A capability to record actual traversals of the traffic network is important. This is considered separate from the actual description of the network. Traversal descriptions need to account for the network characteristics at the time the traversal occurred. A traversal might be described as either an ordering of nodes (points of interest) in the traffic network or as an ordering of edges (street segments).

A *point of interest* in the road network for the purposes of this research is defined as a change in the characteristics of the road network. This might be a change in road heading or classification, the occurrence of an intersection or any kind of noteworthy area or location of interest. Another technique for recording a traversal is to have an ordering of connected street segments in the road network. A street segment occurs between two points of interest in the system. The approach used for traversal definition for this research was to use an ordering of points on the road network.

Ordering of points of interest in the network was considered the better approach because it made describing interaction that occurs at these points easier. This interaction might be for example a vehicle stopping at a signalised intersection. The state of a signalised intersection at the time of a particular point traversal might be of interest. So for these reasons and others the location of signalised intersections must be known.

Finally, recording data ownership is important in a schema for processing data from different sources. For example data may be collected and owned by one organisation, operated on by another; with final analysis conducted by a third. The schema should track data ownership and other relevancies such as who collected the data originally and in what projects data has been used.

4 ExtenDB

ExtenDB is a 'shared-nothing clustered database system'. (ExtenDB 2006) As the automation of traffic networks increases so does the amount of available traffic data. High end enterprise data processing technologies, while up to the challenge, are only available at unacceptable costs. A promising solution, ExtenDB, appears to offer the computational muscle to meet this challenge with reasonable total cost of ownership.

ExtenDB can be used in the creation of an extensible data processing system. This is achieved by leveraging many, individual commodity computers interconnected by a standard network to create a distributed database management system. ExtenDB Standard Edition was used for the work described in this paper.

Standard Edition only requires ExtenDB to reside on one coordinating computer, other computers in the cluster need only be loaded with a standard relational database management system. The preferred relational database management system PostgreSQL (Postgres) was used for the implementation under consideration in this paper. ExtenDB coordinates the communication between each networked computer and provides the user with a standard SQL interface.

ExtenDB database clusters can be implemented on either Linux or Windows operating systems. Postgres is also available for Linux and Windows so a cluster may be assembled from computers using either or both operating systems.

5 The Shared Nothing Architecture

The shared nothing architectural approach uses parallelism. Complex data operations are partitioned and executed in parallel with the result that execution time is reduced. (Thompson & Kung 1977).

The term 'shared nothing architecture' refers to a type of distributed system where each computer has independent processor, main memory and hard disks. This can be contrasted to the traditional multi processor shared memory, shared disk architecture normally used for enterprise systems.

As more computers are added to a system with a shared nothing architecture, comparative performance gains are reduced due to the relative increase in coordination overhead. (Marek & Rahm 1993) There are exceptions however; adding more computers to the system in general means more available cache and less reliance on hard disks for running a given query. In some cases this might lead to dramatic performance benefits.

The shared nothing architectural approach was chosen after significant investigation by the TSC. The independent nature of the computers in such a system should allow easy scaling, more computers can be added to the system with comparatively little effort. (Dineen, M &

Cahill, V. 2002) When a new computer is added to a shared nothing data warehouse, adjustment of data distribution is typically straightforward and when implemented on commodity hardware price-per-performance can be favourable.

Figure 1 describes how data warehouse performance theoretically scales linearly with the shared nothing architecture. For large queries however, reality does not often reflect this with performance returns diminishing after the addition of each new computer. Even so large systems of for example one hundred computers have demonstrated potency (Tamura, Oguchi & Kitsuregawa 1997).

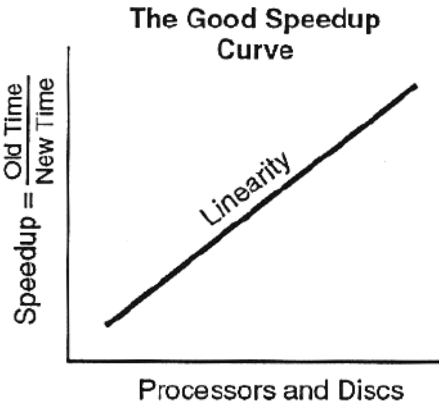


Figure 1: Number of computers vs. performance. (DeWitt & Gray 1992)

When implementing ExtenDB on this architecture the database administrator can configure indexing and partitioning in the same manner as on a traditional centralised database. Distribution configuration such as relation duplication and replication are performed with simple commands, ExtenDB handles most of the complexity. Replication is often worthwhile on a shared nothing architecture for small relations such as lookup tables as it makes common joins faster by reducing row shipping (data transmission between computers).

A disadvantage of most distributed systems is the increase in architectural complexity. A distributed system offering no redundancy on some part of the data means the failure of a single data storage component can be catastrophic. For a decision support data warehouse with the data archived elsewhere this is in reality probably only an inconvenience but the point remains. With regard to the shared nothing architecture on commodity hardware if a failure leaves data storage devices unaffected the component should be simple to replace with minimal system downtime.

For high transaction count databases the shared nothing architecture may be unsuitable as maintaining consistency and reasonably balanced distribution might prove problematic.

6 Nexus II Overview

It is clear that good traffic control strategies do have worthwhile real world impact on traffic network performance (Skabardonis Bertini & Gallagher 1998). In an effort to better inform traffic control strategy, the TSC developed Nexus (Vogiatzis et al. 2005). Work has continued with the Nexus II traffic control database prototype.

Nexus II is a catch-all system intended to warehouse large and disparate collections of road network and traffic monitoring data in such a way that it can be accessed and manipulated quickly. Nexus II is a move towards a complete traffic data solution to inform all aspects of traffic control and traffic research. This differs from other agent based approaches where some of the available information might be abstracted. (Roozemod & Rogier 2000) While abstraction may simplify decision making, forcing simpler decisions excludes some data from the decision making process, this may be undesirable.

6.1 Chronology of Nexus II

The University of South Australia Transport Systems Centre relies on a collection of independent databases and data processing applications for retaining, processing and tracking traffic data. Data resident in these databases is not uniform and has surplus redundancy. Answering research questions which require comparison between these databases requires time intensive conversion.

In the spirit of traffic data normalisation efforts at other Traffic Engineering Research Organisations (ITS Australia 2006, ITS Japan 2006) Nexus II normalises a number of available disparate data sources into a unified schema. This will enrich existing data and so allow TSC Traffic Engineers to obtain better value from data already available. Cross comparison will also be easier with data conversion no longer required. Investigating the derivation of extra value from existing data is important with others engaging in work of this type (McCabe 2004).

Most of the Nexus II development effort focussed on schema refinement. Finding a suitable way to integrate the available data types proved challenging. The current design incorporates data and schema characteristics from four sources:

- an existing SCATS database
- Dynamis Emissions, a database containing vehicle travel time and emissions data
- Nexus, an earlier transport research system
- a road network database, a GIS representation of the Australian road network

Earlier work flagged a real time processing capability as important, (Vogiatzis et al. 2003) for example to inform intersection signalling in real time. This has not yet been addressed and remains open to future investigation

6.2 Nexus (Nexus I)

Once an initial merged and centralised schema had been developed, an indexing, partitioning and distribution strategy was devised and used to define a distributed schema (Nexus II) suitable for implementation on a shared nothing architecture. The distributed schema is suitable for parallel processing on a commodity hardware platform interconnected by standard networking hardware.

Nexus I was a seminal effort that (among other issues) considered how a road network and road network traversals might be described in a database (Vogiatzis et al. 2005). It used an object relational approach to define points of interest on the road network.

The design offered some modelling solutions of the following:

- point of interest on a road network, namely street intersections
- road network link (street) characteristics, for example road type
- geospatial definition of the road network
- traffic project management data

Nexus I had not been adapted to function over a shared nothing architecture or to provide all required road network and road network traversal modelling capability. The system required extension and adaptation to robustly fill traffic data processing needs and to leverage the power of the shared nothing architecture.

6.3 Nexus II and the shared nothing architecture

The Nexus II distributed schema first partitions and then distributes data with a view to providing parallelism. Distribution is at present decided by the default ExtenDB hash. Smaller relations are completely replicated to all computers to minimise data shipping. Partitioning is used on SCATS data which is partitioned by geographical region and month in order to restrict relation size.

Indexing is by primary key, fields likely to be used for joins and on distribution fields.

6.4 Data Representation Issues

ExtendDB does not support the PostGIS geometry data type commonly used for geospatial data representation and so geospatial location definition of the road network could not simply be copied into the schema from the source data. A conversion was required so that coordinates could be recorded as integers. This was achieved with SQL statements. An unavoidable side effect is the ExtenDB implementation may require additional middleware to service applications which depend on a PostGIS geographical representation.

In addition to this, because the source schemas integrated into Nexus II were diverse in terms of the kinds of data they contained, it was difficult to directly relate them to each other. To resolve this issue a series of relations were made as a bridge and in some cases new relations were defined for a richer data representation.

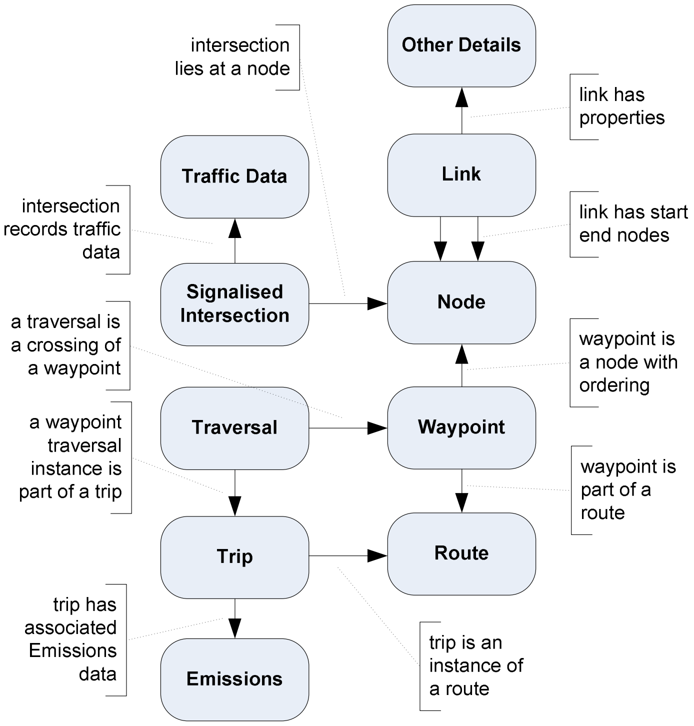

intersection lies at a node
Other Details
link has properties
Traffic Data
Link
intersection records traffic data
link has start end nodes
Signalised Intersection
Node
a traversal is a crossing of a waypoint
waypoint is a node with ordering
a waypoint traversal instance is part of a trip
Traversal
Waypoint
waypoint is part of a route
Trip
Route
trip has associated Emissions data
trip is an instance of a route
Emissions

Figure 2: Summary of data integration.

Relations are used to model the traffic network as a set of nodes and links. Possible routes are described as an ordering of waypoints. Instances of travelling the road network (a trip) were a collection of waypoints with an additional temporal aspect. Emission data may also be linked to a trip if required. Other traffic data (i.e. SCATS records) may also be referenced; a bridging relation (signalised intersection) was created to indicate that specific data were recorded at a particular location (node) at a particular time. A road segment (link) may also have properties, (other details) for example road type or speed limit. See Figure 2 for a summary of the integration approach.

6.4.1 Describing the Traffic Network

The traffic network is described with two simple concepts; a node and a link. Each node represents a point of significance in the traffic network, such as an intersection, the changing of the style of road, a change in street name or anything else considered noteworthy by the researcher. A link is used to model a traversable roadway edge between two nodes in the network. As a segment of roadway, each link contains and references its own set of properties, referencing street, road classification and road details to store the street name, class of road and any additional road information about a given stretch of road.

In particular, a link has the following properties:

- same street name throughout
- uniform speed limit
- same road classification details
- same heading, a curved road would be described by many links

Therefore, a node can be defined as a point in the traffic network where:

- there is an intersection of 2 streets
- the speed limit changes
- road classification details change
- the road bends
- any other point of interest

6.4.2 Describing a route through the Network

A route through the network may be represented by chaining together nodes as *waypoints*. Each waypoint represents the ordered traversal of a node as part of a defined route. Thus, each waypoint has an order field which records the order in which it is traversed in the given route. The trip relation, which is used to reference actual emissions data taken on an actual traversal of a route, relates to many waypoint traversals, each of which store the arrival time of the vehicle doing the trip at a given waypoint in the route.

6.5 Extensibility

The Nexus II schema is extensible across as many computer systems as computational needs demand.

Avoiding data shipping is desirable because of the comparatively slow network links. One strategy to achieve this is to replicate small relations to all computers. Distribution is transparently handled by ExtenDB so the number of computers is irrelevant to the user. Neither the database administrator, nor users need be aware of the physical location of distributed data. Distribution to four, ten or one hundred nodes appears *logically* identical.

When more storage capacity or computational muscle is required, more computers may be added to the cluster and the relevant relations migrated to correct the distribution. No redesign of the database itself necessary.

The solution is cost effective to implement and to extend. Postgres is free and cost effective commodity hardware may be used for implementation. The standard edition of ExtenDB is free to use and supports up to four computers, a single workgroup license of ExtenDB will support up to 16 computers.

7 Nexus II Initial Testing

For a preliminary indication of the Nexus II schema's performance three queries were executed on three different distribution configurations. SCATS data partitioned by month and region were used for all tests.

The first distribution configuration was a traditional centralised (non distributed) Postgres implementation residing entirely on one computer. The second was a distributed version implemented with ExtenDB, in which the ExtenDB hash defined the distribution. For the third configuration distribution was forced by manually allocating individual month/region partitions onto each of the test computers.

A portion of the Nexus II schema was established and 6 GB of SCATS data (which is 344,921,279 records) were loaded. This testing is considered indicative only, confirmation of findings requires a more extensive test suite than this preliminary work presents. Testing investigated whether the platform and architecture under consideration is suitable for fast SCATS data processing, the aim is to investigate performance and if appropriate to flag any likely performance bottlenecks. See Figure 3 to Figure 6 for key performance results.

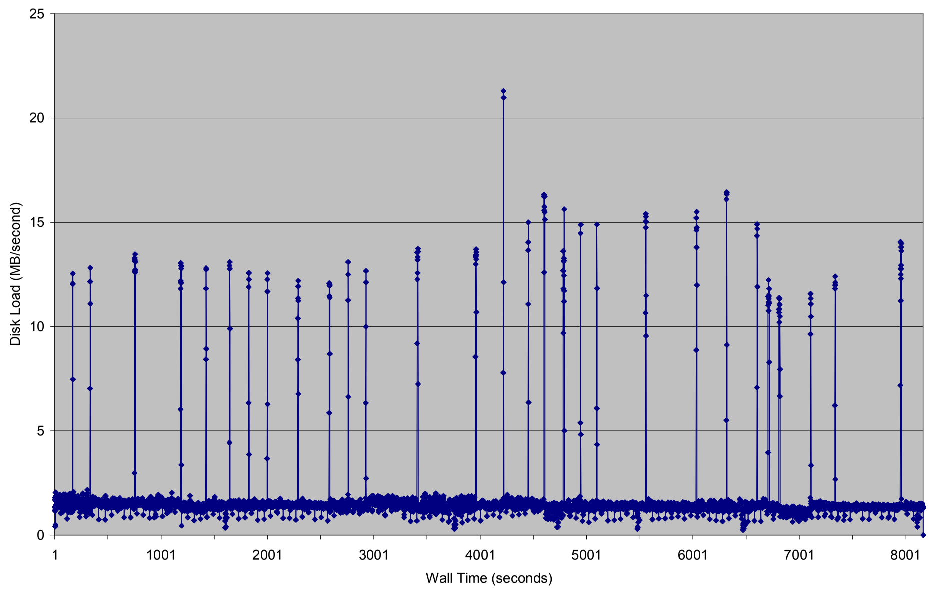


Figure 3: Nexus II centralised, disk performance.

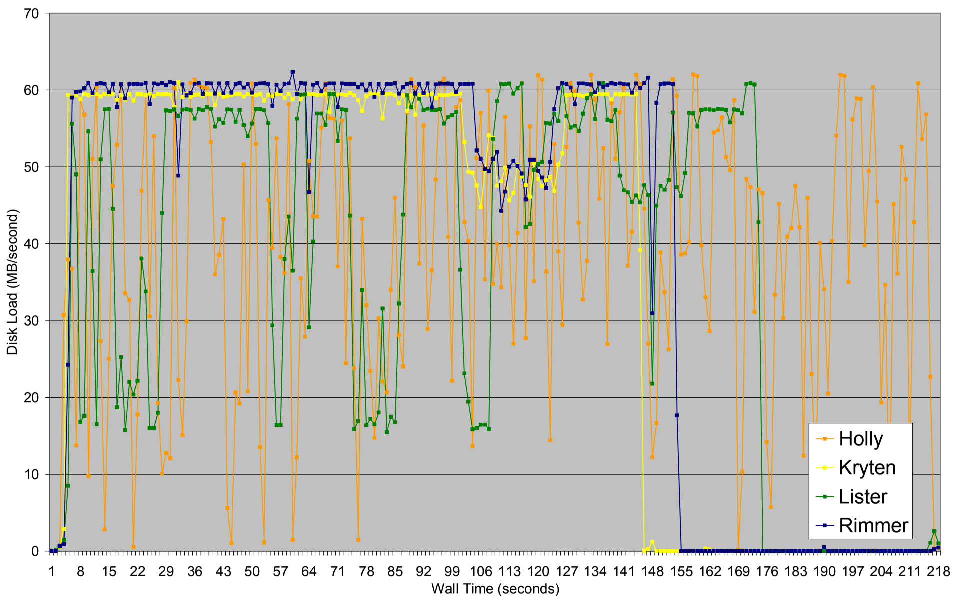


Figure 4: Nexus II distributed, disk performance. (4 computers)

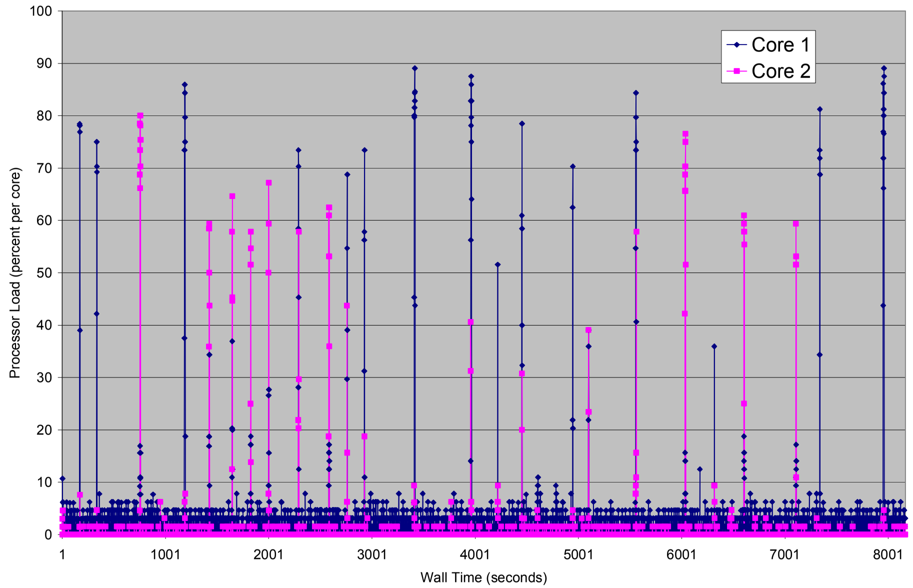


Figure 5: Nexus II centralised, processor performance.

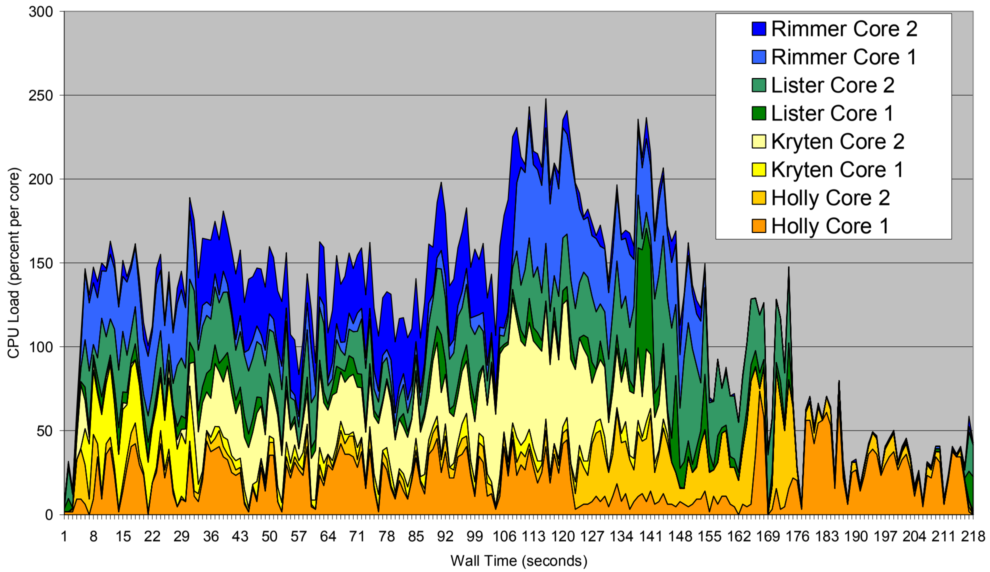


Figure 6: Nexus II distributed, processor performance. (4 computers)

The test platform consisted of five of the following computers connected by 100Mbit LAN

- Microsoft Windows 2003 Enterprise Edition operating system
- dual core Pentium V 3.2GHz processor
- 2 GB main memory
- 7200RPM 80 GB *operating system* hard drive, 7200RPM 250GB *data* hard drive

The computers are named Holly, Kryten, Lister, Rimmer and Cat. Cat was used for testing related to the centralised database configuration and Holly, Kryten, Lister and Rimmer computers were used as the platform for the distributed database.

The expectation before testing was that distribution by ExtenDB default hash would deliver the best performance; this configuration was expected to maximise parallelism. The alternative, distribution by placing unfragmented relation partitions on each node was expected to perform worse but still better than the centralised version. This was because no parallelism was possible for intra month operations.

The first test query calculated the number of times a detector alarm was recorded at five minute intervals on a selection of intersections in the available data. The second determined the number of times no data was recorded for each five minute period on a selection of intersections in the available data. The third query returns the average volume of traffic for each five minute period over the available data across selected intersections. Figure 3 to Figure 6 graph performance of the first query on the centralised and distributed by hash database versions.

Note that there are no results included for the unfragmented distributed implementation of the system. This portion of testing was abandoned after the first query had not finished after nineteen hours of processing. From a cursory analysis of the performance logs the system had become completely unresponsive at this stage.

Consider Figures 3 and 4: there is a significant difference in the disk utilisation between the centralised and the distributed systems. With execution time also shown as appreciably shorter, over 2 hours (8166 seconds) on the centralised system and about 3 and a half minutes (219 seconds) on the distributed system. The size of the result set was 620 records.

344 million records were processed in 3.5 minutes

Figure 4 clearly indicates Kryten, Lister and Rimmer were bound by hard disk performance for most of the distributed query. During testing the Holly computer hard drive was registering a SMART error so it was felt Holly's performance was also disk bound, though Figure 4 does not make this readily apparent. Even so, Figure 4 shows the distributed system was processing 200 MB of data per second for most of the query.

Figure 3 shows the centralised implementation could only consider on average 2 MB of data per second. It was felt the smaller cache (2 GB compared to 4 * 2 GB = 8 GB) of the centralised system meant the distributed system was forced to repeatedly page to disk through the entire operation. This crippled disk performance. When comparing Figure 3 to Figure 5 it is apparent that when the hard disk could attain a reasonable transfer rate the processor was able to take advantage.

Figure 6 makes it apparent that the distribution of the 6 GB data set to four computers (1.5 GB per computer) meant each system was able to cache the entire data set and so load achieve a better work rate from each processor during execution. It is likely that has the centralised system also had 8GB of memory for caching the performance difference between centralised and distributed configurations would be less dramatic.

The expectation is that for queries that are significantly larger than cache sizes performance of the distributed system would increase linearly, so performance would be roughly four times that of the centralised system. (DeWitt & Gray 1992)

Performance monitoring was comprehensive, network utilisation was measured to record row shipping. None occurred other than for the small result set.

Future testing will involve more complex operations that require row shipping. This can reduce performance if large amounts of data must be transferred between computers. (Thomas et al. 1990) Although unlikely, the authors note that future testing is needed to confirm the performance monitoring utility used, 'perfmon' had no appreciable effect on performance.

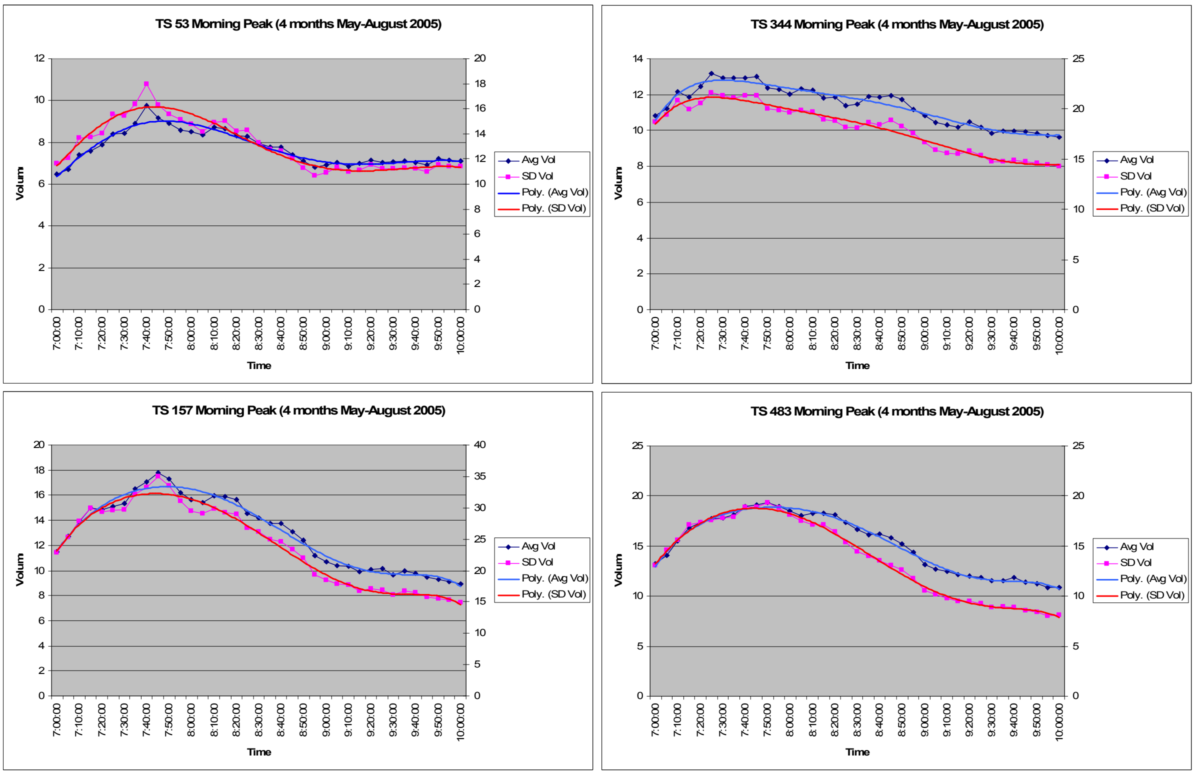


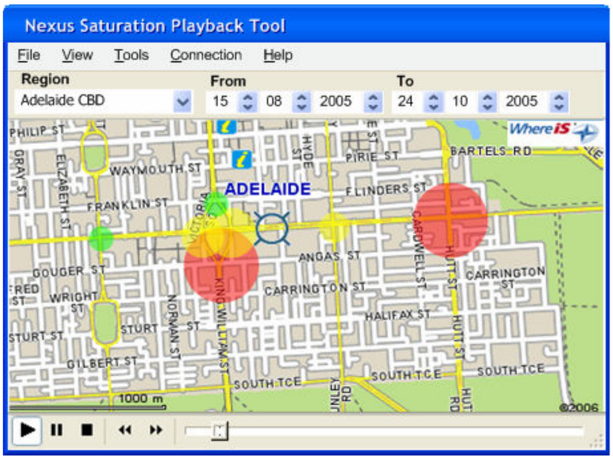
Figure 7: Four examples of intersection profile analysis obtained from the queries performed on the computational cluster (Vogiatzis and Taylor 2006).

Of significant interest is also the actual analysis of the intersections. As can be seen in Figure 7, morning peak profiles for four (of a total of twelve) intersections are shown (see Figure 0 for the geographic location of the intersections). Following the data we can see that depending on the 'closeness' of the intersection to the CBD affects the time at which the intersection peaks. Additional analysis performed by Vogiatzis and Taylor (2006) showed that potentially there is a mix of influence between the location of the intersection in relation to a major attractor (such as a CBD), route choice, link purpose (such as if the link is freight corridor) and land use. Vogiatzis and Taylor concede that the initial analysis conducted as a result of the experimental queries being performed was not sufficient to come to any conclusions, rather that the availability of the computational cluster has provided an opportunity to further explore the link in question and the addition of more data sources will provide for a clearer picture of the reasons for the results through the development of an expert system. Without the ability to compute solutions for large datasets connecting disparate data sources, the analysis is likely to take significantly longer and would need to be simplified for expediency.

8 Nexus II Future Directions

The Nexus II development effort has so far emphasised a data warehousing approach. As such the architecture, schema design and proof of concept testing have been conducted with this in mind.

When loaded with historical data for a given region, Nexus II not only supports complex data mining queries but is intended to eventually be a large traffic data repository. This repository could then be used to support a traffic data processing framework. The goal is to offer excellent processing capacity over all traffic data for a given region of interest. This data and processing capability could then be used as needed by applications through a simple set of interfaces. Some concept examples will now be presented.


Region: Adelaide CBD

From: 15 08 2005

To: 24 10 2005

Figure 8: Mock-up of a saturation playback tool, road network image sourced from WhereiS. (2006)

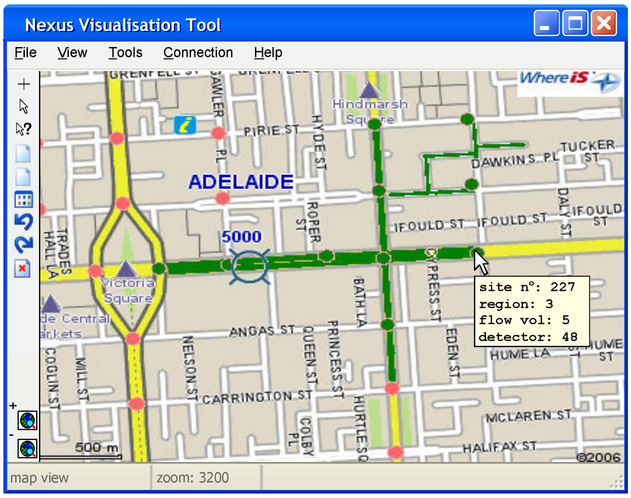
The obvious application would be to better inform real-time traffic signalling. Signalling decisions made in real-time by traffic signal control devices could reference a massive repository of traffic data for signalling decisions. Decisions made by a signalling device could account for traffic conditions in an entire metropolitan district if necessary.

Nexus II design has so far only been considered from a single user data warehousing standpoint. If Nexus II were to be used in a transactional sense (simultaneously by many users) issues such as concurrency control and multi user performance would require investigation.

Nexus II could comfortably support several traffic visualisation and analysis tools. Figure 8 presents a traffic network saturation playback tool. The tool might play an intersection saturation timeline as recorded by selected devices. The tool would allow researchers to instantly view traffic patterns over a given region and time period.

Another straightforward extension would be a report generation tool that allows researchers to query the Nexus II database from a remote GUI. Reports could be created without the researcher needing to devise the SQL request and in the extreme cases, the researcher may not need to know SQL.

More work is required to develop support technology, for example a flexible data loader that loads data from either source databases or from data files. Presently Nexus II relies on manual copying of data directly from source schemas, the exception is SCATS data for which an existing parser has been suitably modified. Ideally a single loader that supports all data types should be developed.


site n°: 227
region: 3
flow vol: 5
detector: 48

map view | zoom: 3200

Figure 9: Mock-up of a Nexus II visualisation tool, road network image sourced from WhereiS. (2006)

Figure 9 is a mock-up of a visualisation tool that might allow a traffic researcher to specify a desired path through a traffic network and to indicate signalised intersections of interest with point and click interaction.

Currently Nexus II can only account for points of interest on the road network, such as a change in speed limit and road intersections. However there are also entities that do not directly lie on the road network that impact road traffic, for example malls, schools or theatres.

Ultimately Nexus II will be used for long term traffic data storage. The system is currently considered *experimental* so redundancy is only exploited for performance reasons.

9 Conclusion

The Nexus II schema design is logically complete but has not been exercised with any comprehensive testing. Preliminary testing is however promising. Data has been successfully parsed and loaded into the portion of the schema used for traffic network definition and the section used for recording data from signalised intersections.

Proving the schema is logically sound will be time consuming. Developing the required parsers and the mere logistical aspect of loading large amounts of data is very resource intensive. Actual generation of previously mentioned linking data has not yet been attempted. Ideally though if the data is loaded once it could then be stored indefinitely.

With regard to performance, initial results are promising, distributing the Nexus II schema across four computers resulted in completion of queries across ‘small’ 6 GB data sets forty times faster when compared to a traditional implementation. However, for queries significantly larger than the available cache, performance gains are however expected to be closer to linear.

10 Acknowledgements

The authors wish to thank the following people for their ongoing support and for generously offering their expertise. This paper would not have been possible without them.

Ms Kirsten Wahlstrom
University of South Australia, AUSTRALIA
http://people.unisa.edu.au/Kirsten.Wahlstrom

Ms Sue Tyerman
University of South Australia, AUSTRALIA
http://people.unisa.edu.au/Sue.Tyerman

Mr Mason Sharp
ExtenDB Inc, USA
http://www.extendb.com

Mr Tai Dinh
Department of Transport, Energy and Infrastructure
http://www.transport.sa.gov.au

11 References

DeWitt, D & Gray, J 1992, 'Parallel database systems: the future of high performance database systems', *Commun. ACM,* vol. 35, no. 6, pp. 85-98.

Dineen, M, & Cahill, V 2002, Towards an Open Architecture for Real-Time Traffic Information Management. viewed June 2006, <http://www.cs.tcd.ie/publications/tech-reports/reports.02/TCD-CS-2002-01.pdf>.

ExtenDB 2006, ExtenDB - Cost-effective Data Warehousing and BI, viewed May 2006, <http://www.extendb.com/>.

Gardner, S 1998, 'Building the data warehouse', *Commun. ACM,* vol. 41, no. 9, pp. 52-60.

Ikeda, H, Vogiatzis, N, Wibisono, W, Mojarrabi, B and Woolley, J 2004,Three Layer Object Model for Integrated Transportation System, In 1st International Workshop on Object Systems and Systems Architecture.

ITS Australia 2006, Home, viewed July 2006, http://www.its-australia.com.au/kmxserver3/.

ITS Japan 2006, ITS Japan, viewed July 2006, http://www.its-jp.org/english/.

Marek, R & Rahm, E 1993, 'On the Performance of Parallel Join Processing in Shared Nothing Database Systems', *5th International PARLE Conference*, pp. 622-633.

McCabe, G 2004, SCATSSIM: Interfacing SCATS and Traffic MicroSIMulations, In Workshop on Traffic Simulation,

Roozemod, D & Rogier, J 2000, Agent controlled traffic lights, Aachen, Germany, viewed April 2006, <http://www.erudit.de/erudit/events/esit2000/proceedings/AD-01-3-P.pdf>.

Skabardonis, A, Bertini, R & Gallagher, B 1998, Development and application of control strategies for signalized intersections in coordinated system, Transportation Research Record, National Research Council, Washington DC, viewed April 2006 <http://web.pdx.edu/~bertini/981271.pdf>.

Stonebraker, M 1986, ‘The Case for Shared Nothing’, *IEEE Database Eng. Bull.*, vol. 9, no. 1, pp. 4-9.

Tamura, T, Oguchi, M & Kitsuregawa, M 1997, 'Parallel database processing on a 100 Node PC cluster: cases for decision support query processing and data mining', *Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)*.

Thompson, C & Kung, H 1977, 'Sorting on a mesh-connected parallel computer', *Commun. ACM,* vol. 20, no. 4, pp. 263-271.

Vogiatzis, N and Taylor, MAP 2006, ‘On Concurrent Holistic-Component Dynamics (Abstract Submitted)’, In ‘Australasian Transportation Research Forum 2006’, Gold Coast, Australia

Vogiatzis, N, Zito, R, Ikeda, H, Primerano, F, Vanderlinde, B & Wibisono, W 2005, 'Techniques for Analysing Large Volumes of Data Generated by Traffic Research', Proceedings of the Eastern Asia Society for Transportation Studies, vol. 5, pp. 1522 - 1532.

Vogiatzis, N, Ikeda, H, Woolley, J & He, Y 2003, ‘Integrated Multi-Nodal Traffic Network Systems’, *Journal of the East Asia Society for Transportation Studies, vol. 5, pp. 2092-2107.*

Vogiatzis, N 2003, ‘On Improving Traffic Micro-simulation Techniques: A Research Proposal’, paper presented at the 25th Conference of Australian Institutes of Transportation Research Adelaide, Australia.

WhereiS 2006, whereis.com - Search Australian Maps, viewed June 2006, <http://whereis.com.au>.