Syn~
Business Positive

White paper

May 2007

Richard Barnes

Syn~ Scalability benchmark

Summary

This document describes an exercise to measure the scalability and performance of the Syn~Settlements application upon a Linux/HP platform.

The results recorded include:

- demonstrated scalability consistent with a linear model;
- peak throughput of the scalable system at 77K trades per hour;
- demonstrated efficacy of quad-core processor architectures as a means of scaling the database tier;
- maximum throughput recorded on six CPUs was 105k trades per hour.

Context

In Autumn 1997 a permanent working group was formed with the purpose of measuring the performance and scalability of Syn~ (then Project Aurora) and researching approaches to improve this. In addition to this steady ongoing work we are periodically requested by prospective users of Syn~ to provide evidence of STP rates given a particular hardware specification and business model. As production-scale hardware is not readily available in-house, Coexis has worked with their partners to achieve such benchmarks. Results from successive benchmarking exercises together with the results from the most recent such study, the subject of this paper, are summarised in table 1 below.

Some interesting trends are evident in the results displayed in table 1 [1-4]. Recall that the driver for any of these exercises is a request from a prospective client for an estimate of expected performance on a given hardware deployment. The emphasis for hardware has shifted from an initial concentration on large-scale "enterprise" SMP machines through mid-range Unix servers to larger numbers of smaller x86-based architectures and Linux-based operating systems.

The measured performance of Syn~ clearly demonstrates an upward trend, to the extent that the system deployed in the current exercise upon a handful of x86 Linux machines considerably exceeded the best previously recorded figures upon Solaris/UltraSparc IV. Whilst some of this improvement can be ascribed to increases in processor speed, it is likely that improvements at all levels – hardware, database, Java VM technology and enhancements to Syn~ itself – are contributing to the current performance.

The hardware, operating system and database were kindly made available to Coexis by AEMS London. Mention must be made of the sterling efforts on the part of Mark Howden and Philip Styles in getting everything set up and working, and thanks are also owed to Stephen Elkes for making the exercise possible.

Table 1: Platform specifications, operating system, and results from successive large-scale performance exercises.

	Hardware and OS			Storage	CPUs	Measured throughput			Scalability		
	Database	Server	Client			E	T	P	Model	E	T
Sun, Frimley May 2001	E10000 48 x 400 MHz Solaris 7			48 local drives	48	16	5.6	-	P2	57	12
Sun, Sale July 2003	F6800 24 x 900 MHz Solaris 8		E4500 8 x 400 MHz	8 x T3 Veritas 3.5 striped	32	20	41	-	P2	47	62
Sun, Sale March 2005	F6900 24 x UltraSparc IV Solaris 9		F4900 12 x UltraSparc IV Solaris 9	8 x 3510 striped	36	24	68	-	P2	61	108
AEMS Cannon St Oct 2006	HPDL585 4 x Opteron RHEL4u3	4 x HPDL360 Xeon dual core RHEL/WinXP		HP MSA1000 2 x 2 x 72 GB	8	4	38	53	L	U?	U?
AEMS Docklands May 2007	DL380G5 2 x Xeon quad core RHEL	4 x HPDL380G3 Xeon dual core RHEL		1 local drive	6	8	77	105	P2/L	U?	U?

Key:
Measured throughput records peak capacity for the largest size "on curve" system achieved, where E is the number of elements and T is the number of trades processed in thousand trades per hour; P is the peak throughput measured exploiting the hardware to the full and not therefore consistent with the scalability curve. Scalability is the extrapolation of the on-curve measurements to determine the maximum theoretical capacity of the system. *Model:* the model used to extrapolated where P2 is a 2nd order polynomial (quadratic) and L is a linear fit, E is the number of elements at the predicted maximum and T is the predicted throughput of the system at maximum, in thousand trades per hour. In the case where the data is equally consistent with a linear fit U indicates theoretically unbounded scalability.

Objectives

The previous Linux scalability exercise in October 2006 demonstrated a very effective (consistent with linearity) scaling of the system up to four elements. Whilst all previous data has suggested that the best model of Syn~ scalability is a second-order polynomial, the scaling observed was such that a linear fit to the scalability 'curve' actually gave a better fit than the quadratic. Scaling consistent with linearity is obviously the 'Holy Grail' for systems in which parallelisation is utilised to achieve higher processing volumes. This was the first time that behaviour consistent with linearity had been observed in Syn~ . However, this scaling was not sustained for larger systems of five to eight elements. A clear discontinuity was observed between the extremely effective scaling up to four elements and the markedly poorer scaling of larger systems. The likely cause of this effect was not known although it was felt that it may have been related to some resource contention within the database. A primary goal of the present exercise is to investigate this further and as far as possible to demonstrate scaling beyond four elements. To this end the specification of the database server has been enhanced compared to that used in the previous exercise.

A secondary goal is to assess the possible use of quad core processor architectures particularly as regards their suitability within the database tier and to what degree this allows the capacity of the database to be scaled up. The capacity of the database can be scaled up by 'horizontal' means but this requires investment in Oracle Real Application Clusters (RAC) and the prerequisite network infrastructure.

The use of quad core processors may provide a means of meeting the required demand in some systems without having to adopt the RAC approach.

The Syn~ element is a multi-threaded process which is capable of processing discrete units of the STP workload – individual trades – in parallel. This parallel processing ability is controlled by configuring the number of process queues within the element. Each process queue is a separate Java thread, with dedicated database resource. The Java VM is able to exploit multiple operating systems threads and softly map these to Java threads within the VM. An auxiliary benefit of the use of a dual quad-core machine was that this allowed the scalability of a single Syn~ element to be assessed, by measuring the throughput of a single element as the number of process queues is increased.

Methodology

The methodology for studying scalability has been discussed at length in previous documents within the context of Syn~ [1, 2] and by theoreticians at a more abstract level [5] and will not be discussed here. The business model used for this testing consisted of the standard Syn~Settlements module with the addition of the optional Syn~Ledger module. Figures 1 and 2 show schematic views of the deployment of the system and of the sequence of events corresponding to the processing of a trade.

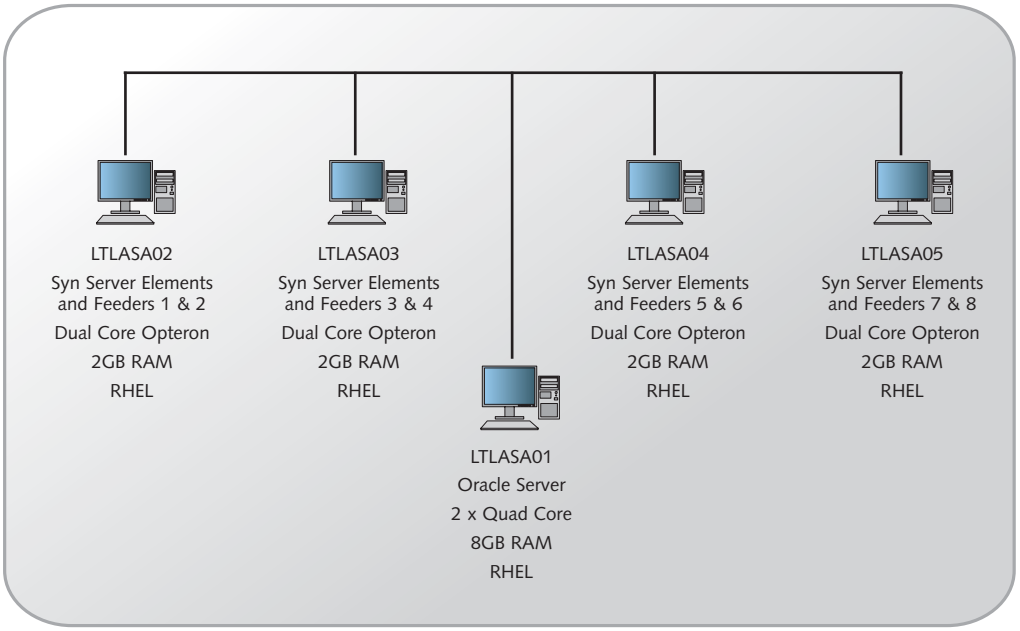

Figure 1: Deployment of the Syn~ Trade Feeder and Syn~ Server Element processes upon the AEMS HP hardware.

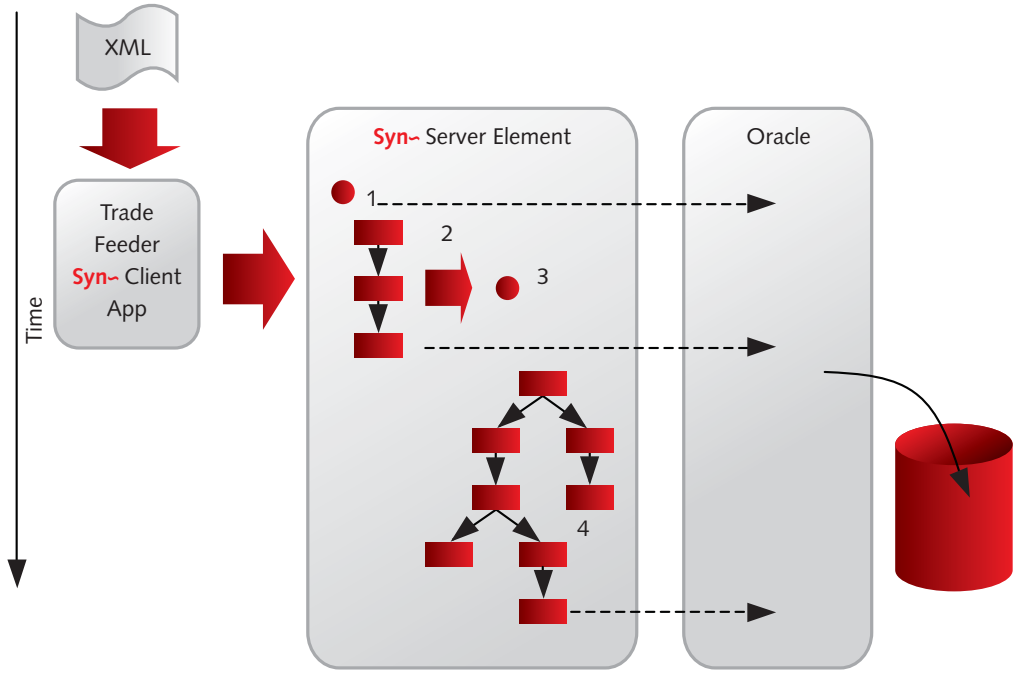

Figure 2: Schematic view of the processing of an XML trade message showing
(1) creation of the IncomingTradeMessage business object;
(2) business process executing on the IncomingTradeMessage;
(3) creation of the SecurityPurcSale business object (representing the trade itself); and
(4) business process executing on the SecurityPurcSale.
The flow of time is down the page; dashed arrows indicate typical commit points.

Elimination of systematic variation

Considerable variation in recorded throughput was observed over the course of the exercise. This was not correlated with changes to the configuration of the application or the database. Identical test scenarios could yield considerably different throughputs when repeated under identical conditions with respect to the database and application

Figure 3 below shows the variation of throughput for an identical eight-element test run repeatedly over a number of days. The throughput shown for each machine is the aggregated throughput for the two elements running on that machine. As can be seen the variation applies uniformly across all machines. This indicates a systematic, rather than statistical, variation.

Figure 4 shows the variation in mean time to process a trade for four of the elements comprising a ten-element system set to process 500,000 trades. The performance is uniform up until midnight at which point 'something' clearly happens which severely impacts the performance of the system as a whole. Elements one to four are shown in the figure, although the degradation is seen across all the elements for the same period of time. As the **Syn~** trade feeder client and element to which it is connected operate in isolation from all other client and element pairs this effect is unlikely to originate in **Syn~** itself. The effect must be caused by some shared resource upon which all the **Syn~** processes are dependent. Possible candidates include the network infrastructure over which the **Syn~** elements communicate with the database; the database itself; or the hardware upon which the database is deployed. Extensive checking of the database log files showed no evidence of anomalous activity during the time period after midnight. Given the timing we strongly suspect that some regular scheduled batch process such as a system backup is occurring at this time and that this is affecting network traffic.

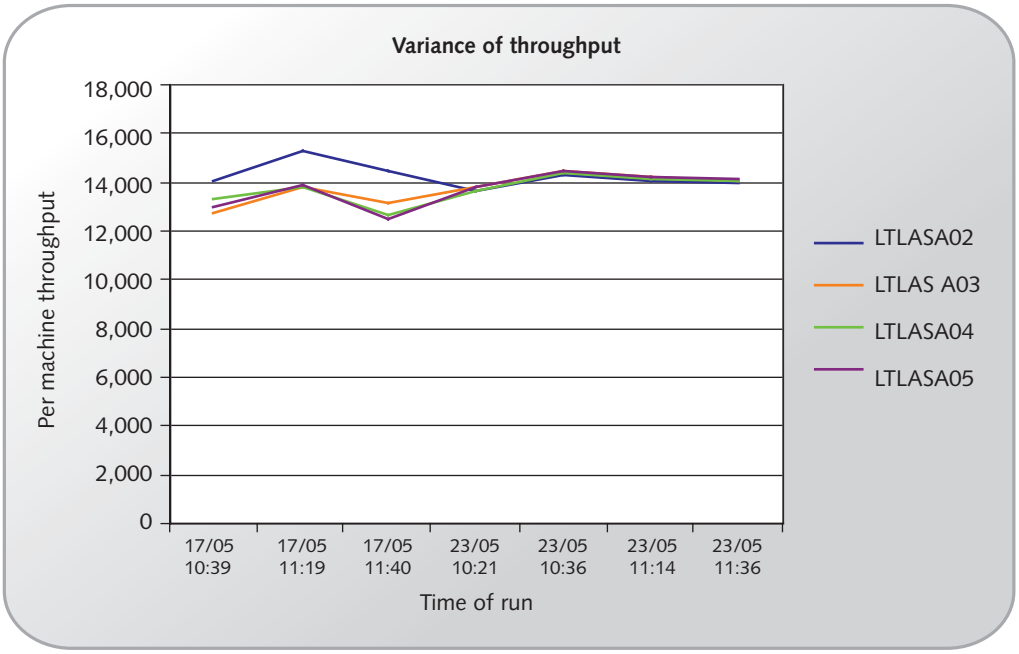

Figure 3: The throughput for the two elements running on each of the application hosts was found to vary with the time of day uniformly across all machines.

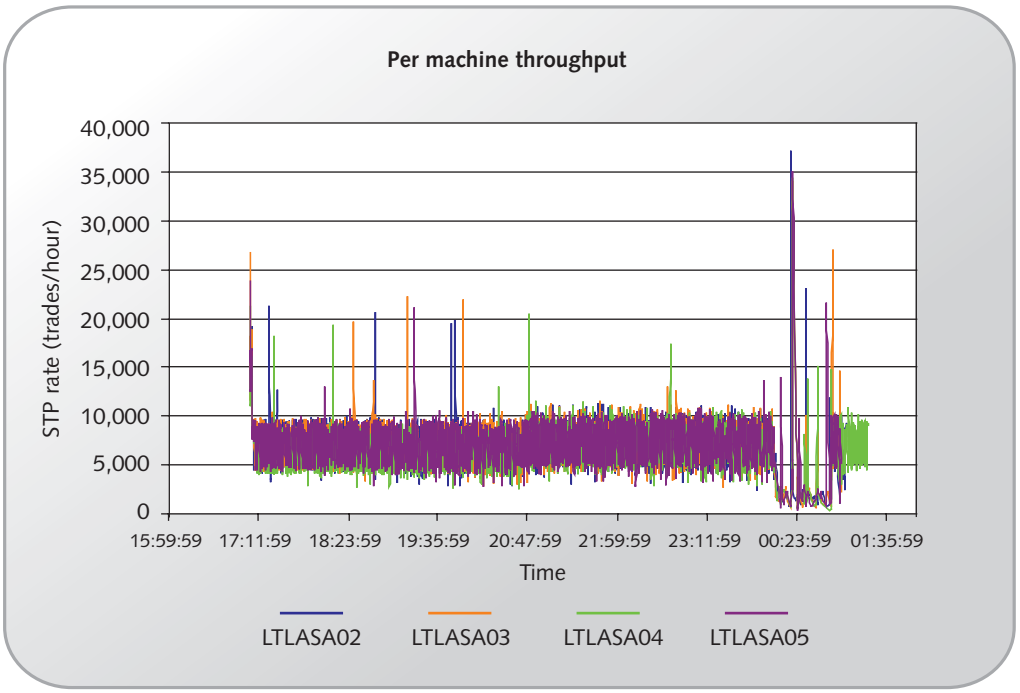

Figure 4: Throughput recorded during an extended data-taking run intended to process 500,000 trades. 'Something' is clearly happening at midnight …

In general, the occurrence of uniform performance variation across all elements has been interpreted as an indicator that some external factor is influencing the behaviour of the system. Care must be taken to avoid such conditions where results are being recorded which are intended to contribute to a broader interpretation such as a scalability graph. Above all a series of results must display coherence and self-consistency – rather than maximised throughput – in order to draw meaningful conclusions regarding scalability.

Results

Scalability versus 'off curve' peak throughput

A meaningful study of scalability requires that the performance of the system be measured as the capacity of the system is increased in identical increments. So for example the throughput of the system may be measured with a single element and trade feeder deployed upon one machine in the application tier. The next point on the scalability curve is achieved by deploying two elements and their corresponding trade feeders, each pair upon its own machine. As four application tier hosts were available this allows a scalability graph to be drawn with points corresponding to one to four elements (figure 5).

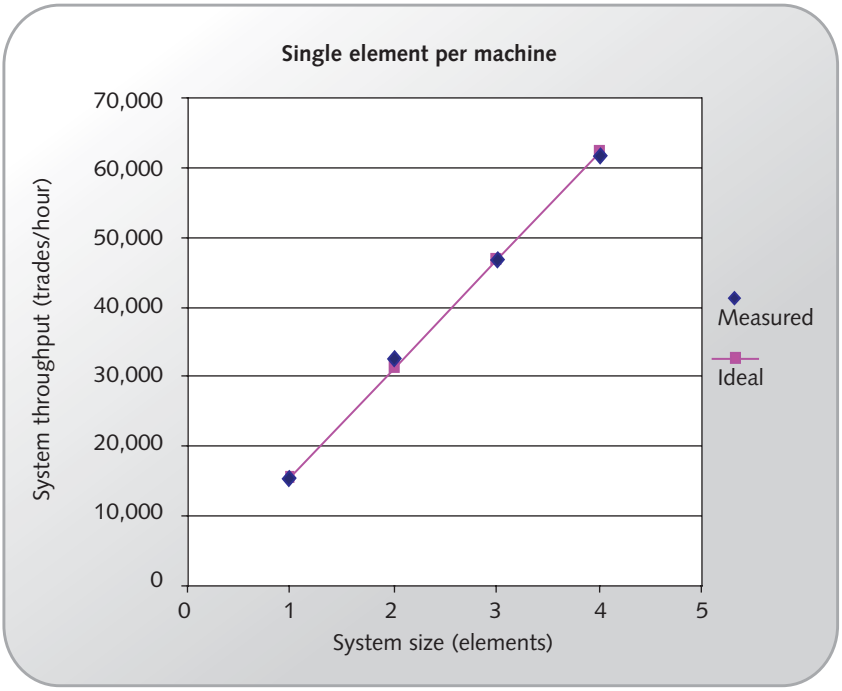

Figure 5: Measured and ideal (linear) throughput for one to four elements

Figure 6 shows the scalability graph for two, four, six and eight elements. This is achieved by running two elements and two feeders upon each application tier host. However this graph is not directly comparable to figure 5. Whilst the throughput processed by a machine in the application tier is increased by running two elements and feeders upon it, it is not double the volume processed if a single element and feeder were running upon it, nor is it the same throughput as achieved by running two elements et al upon two machines. Hence this mode of deployment in which server elements are 'doubled up' on the application hosts must be regarded as yielding a different and separate scalability to the single element per host model. At present the reason for this difference is not understood. Clearly there is resource contention of some kind at the machine level but time constraints prevented investigation in detail as to the source of this.

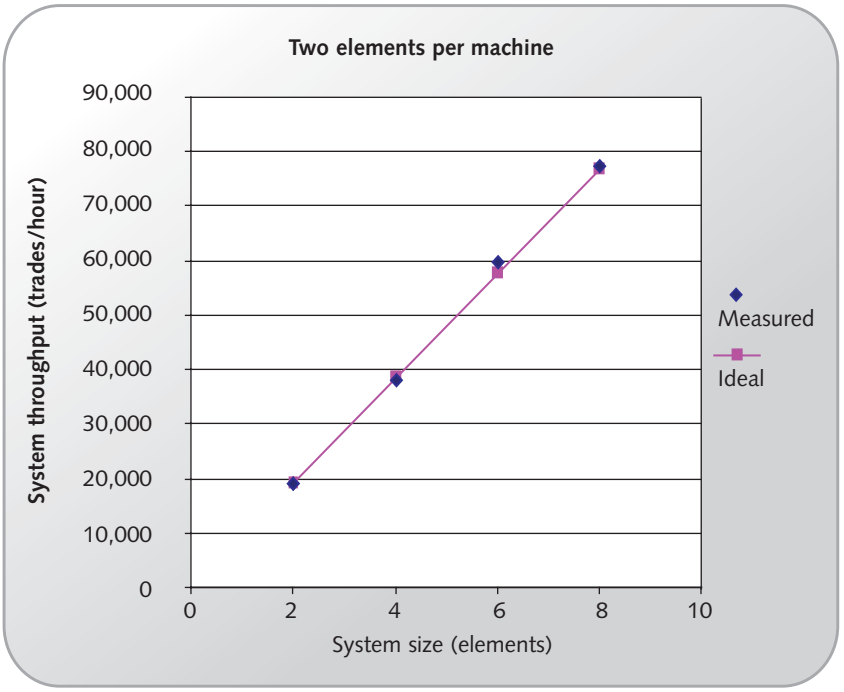

Figure 6: Measured and ideal (linear) throughput for two to eight elements

This observation allows the unexplained loss of scalability noted in the previous study to be explained. As described in the section 'Objectives' above the previous study showed linear scalability up to four elements but an apparent loss of scalability for five to eight elements. It is not valid to combine the results for five to eight elements upon the same graph as those for one to four as these are two separate systems displaying different scaling characteristics and to treat them as a single model is incorrect.

For the purpose of the present exercise the 'doubled up' mode of deployment was adopted as this gave maximal throughput and presented the greatest loading upon the database tier.

The *peak* throughput recorded for the system was 105,300 trades per hour. This was recorded in an ad-hoc deployment of the system in which a further two elements and trade feeders were deployed upon the database host, in addition to the eight elements running upon the application tier hosts. This figure cannot be regarded as a fifth point on figure 6. The abundance of CPU resource on the database host enabled the additional two elements to process trades considerably faster than elements one to eight. Therefore these two new elements do not in this scenario represent identical increments of processing capacity and therefore this peak throughput must be regarded as 'off curve' with respect to the scalability graph.

Loading on the database machine

It readily became apparent that with 'only' four dual cores in the application tier there was little hope of applying any serious load upon the dual quad-core database host. CPU usage on the database host was imperceptible with a single element processing trades, rising to maybe 30% busy with eight elements processing trades concurrently.

Scaling of the Syn~ element

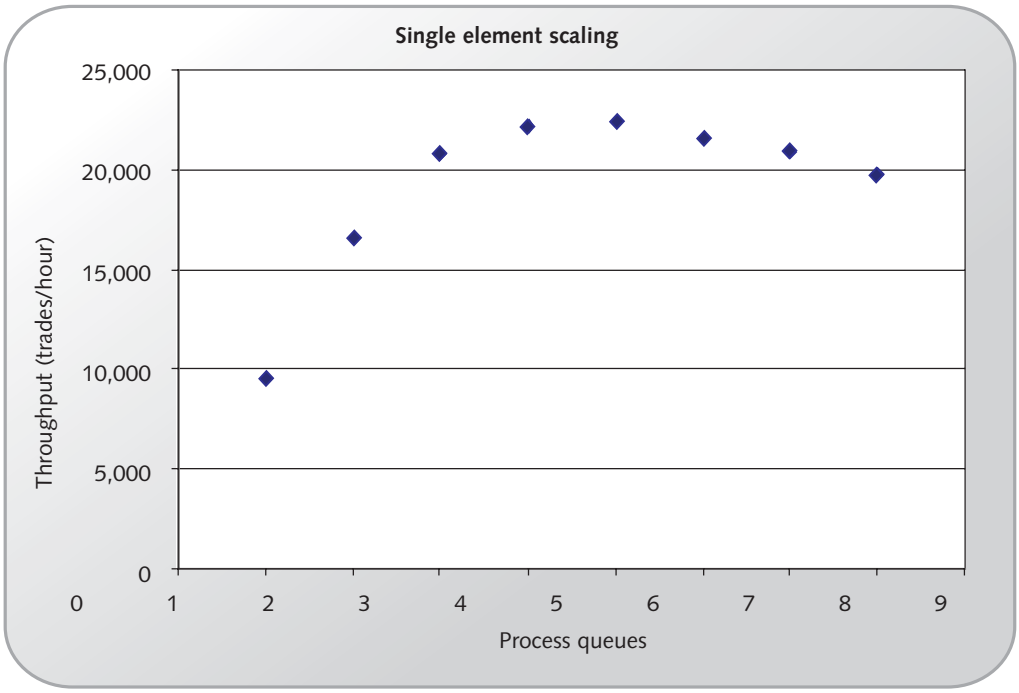

Figure 7: Subject to sufficient resource on the host machine, the element may be scaled effectively by configuring up to four or five process queues.

Figure 7 shows the throughput of a single element versus the number of process queues configured within that element. In this case the process was deployed upon the dual quad-core database host to enable the scalability to be assessed without encountering thread starvation. This would appear to indicate that the element cannot be effectively scaled above four to five process queues. The reason for this is that the Java VM is itself inherently unscalable particularly with respect to memory management and garbage collection.

The surface graphs figures 8 and 9 show the performance of the system for one to four, and two to eight 'doubled up' elements, against the number of process queues configured per element. This allowed the number of process queues to be set at three for optimal throughput in the 'doubled up' deployment mode.

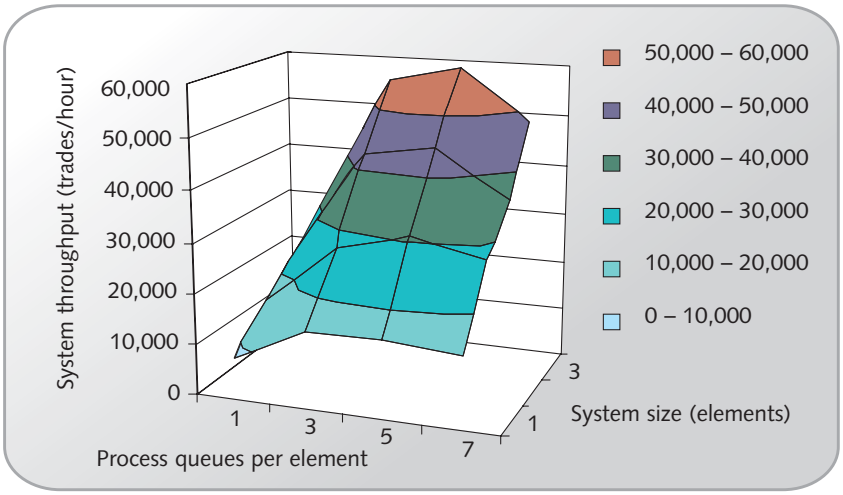

Figure 8: System size versus process queues per element (1-4 elements)

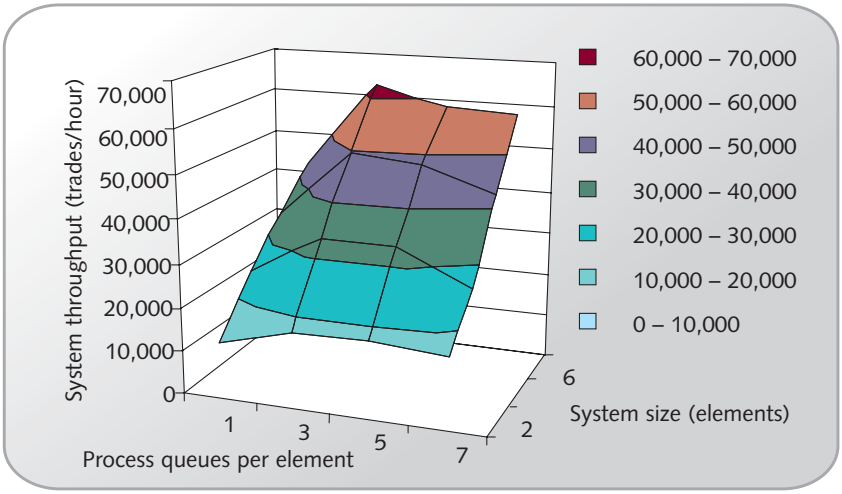

Figure 9: System size versus process queues per element (2-8 elements)

Comparison of position-keeping models

For the majority of figures recorded in this exercise a prototype position keeping model was utilised which attempts to minimise contention upon the position. This model is currently being implemented by Coexis and will feature in future deployments. However existing Syn~ deployments there is considerable contention as a result of the way in which positions are modelled.
Figure 10 shows the effect of this compared to the new prototype model.

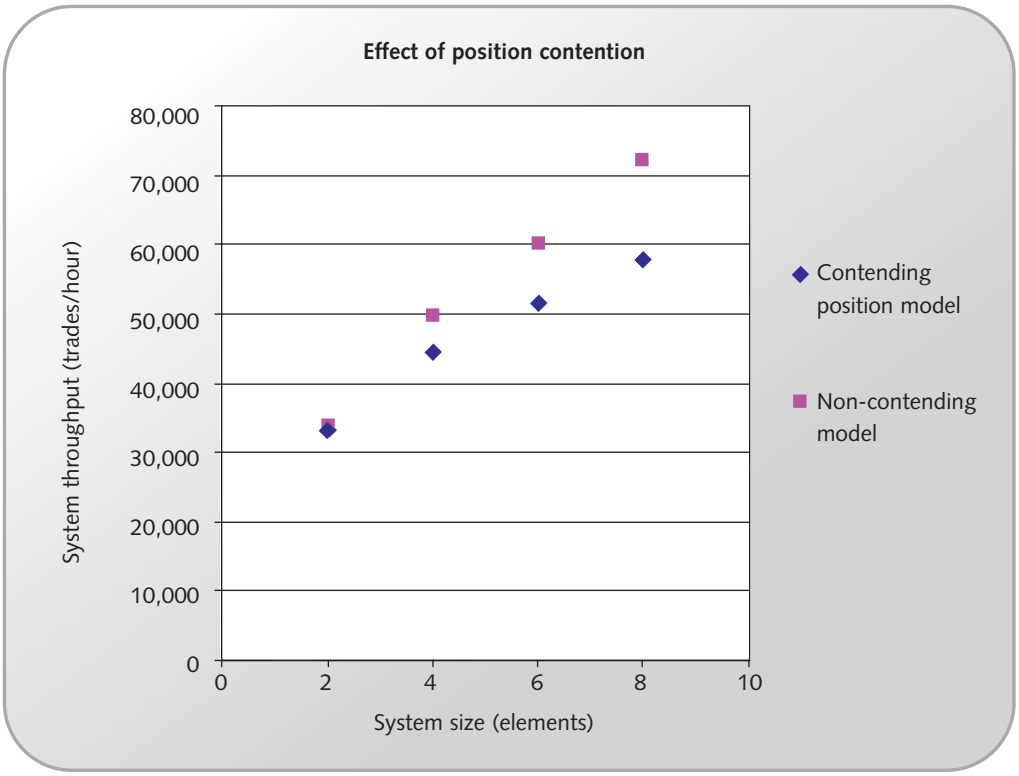

Figure 10: comparison of alternative position keeping models with different contention characteristics

Interpretation

Previous experience has shown that the best fit to the scalability characteristics of Syn~ is a second-order polynomial [2]. Published theories of scalability such as Amdahl's Law and the Gunther Super-seriality model [6, 7], which model scalability in terms of parameters which express contention and the effect of 'cache thrashing' have been found to fit less well to the observed behaviour of Syn~. A least-squares regression fit to the data gathered in the present exercise yield R^2 values of 0.9988 for a linear fit and 0.9989 for a second-order polynomial. The data fits poorly to the Amdahl and Gunther models and indeed in the latter case yields a nonsensical negative value for the super-seriality factor (which describes the effect of 'thrashing').

In general the best model to adopt is the simplest which accurately describes the data and in this case this would be a linear fit. In the light of previous experience this may be interpreted as a system whose behaviour is described by a more complex model (such as the quadratic model) which has only been measured at the close-to-linear 'early' part of the scalability curve, far from the point at which any deviation from linear behaviour becomes apparent.

Conclusions

The following conclusions are drawn:

- on the hardware specified Syn~ demonstrated scalability consistent with linearity, interpreted as being the near-linear "early" stage of a quadratic scalability curve;
- the dual quad-core database host had ample capacity to support an eight element system and had further application hosts been available the recorded scalability graph could have been extended to greater numbers of elements and showing increasing capacity of the system;
- previous conclusions which appeared to show a 'broken' scalability model for systems with greater than four elements were incorrect and arose through an incorrect interpretation of the results (as shown in figure 11);
- limited scaling of the individual element may be achieved by configuring additional process queues, subject to available host resource.

Future directions

Having proved the suitability of the dual quad-core architecture as a database server for Syn~ it is the intention of Coexis to invest in a single such machine and use this in combination with a large number of development desktop machines as the application tier to test larger systems with significantly more than eight elements.

The results have confirmed the importance of developing a model for position-keeping which features a lower degree of contention than that currently used in production Syn~ systems and work is in progress to produce this.

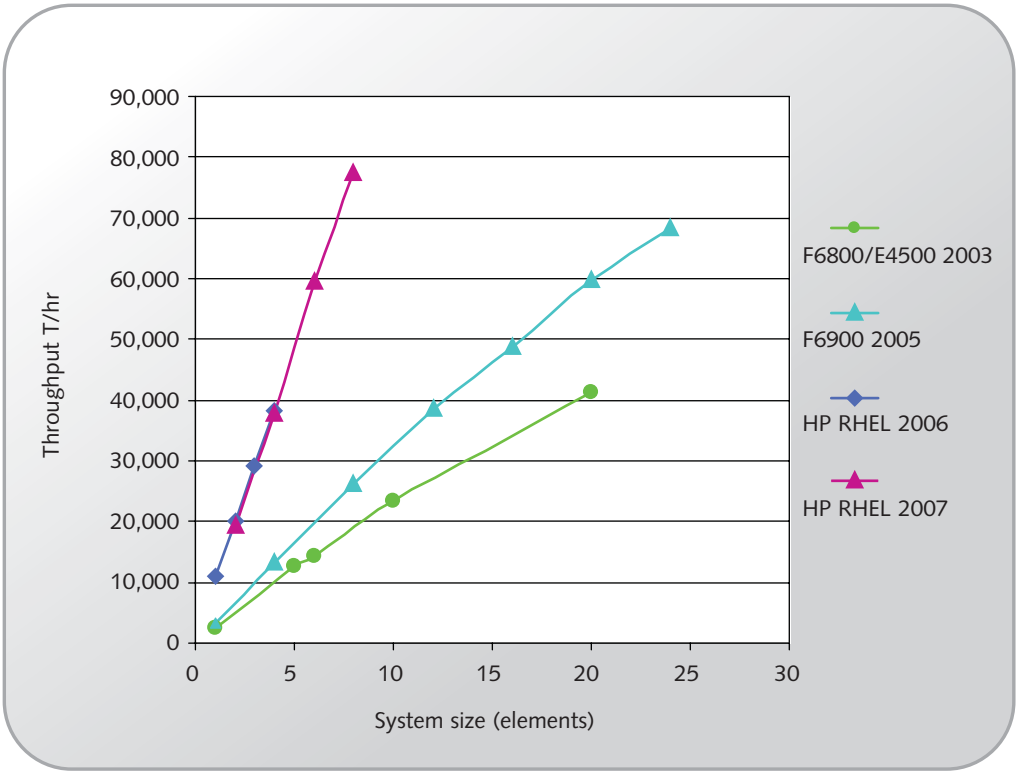

Figure 11 shows the current scalability results compared with previous recorded scalability curves.

References

[1] Barnes, R. P., 2001. 'Report on Scalability Testing on E10000', Coexis Ltd
[2] Barnes, R. P., 2003. 'Syn~ Scalability Testing', Coexis Ltd
[3] Barnes, R. P., 2005. 'Syn~ STP Benchmark', Coexis Ltd
[4] Barnes, R. P., 2006. 'Extract from Syn~ STP Benchmark', Coexis Ltd
[5] Gunther, N. J., 2001, 'How to measure an Elephant', at http://www.teamquest.com/html/gunther/elephant.shtml, Teamquest Corporation
[6] Gunther, N. J., 1995, 'Parallel Processing and OLTP Scalability', at http://www.pha.com.au/arch/neil_gunther/ng3/sld001.html, reprinted from The TPC Quarterly 1995
[7] Gunther, N. J., 2001, 'Commercial Clusters and Scalability', at http://www.teamquest.com/html/gunther/scalability.shtml, Teamquest Corporation
[8] Gunther, N. J., 2001, 'Evaluating Scalability Parameters: A Fitting End', at http://www.teamquest.com/html/gunther/fitting.shtml, Teamquest Corporation

Appendix A: Gunther interpretation of single-element scalability

Appendix A: Gunther interpretation of single-element scalability

The scalability curve given in figure 7 for the single element with respect to number of process queues bears a strong resemblance to the scalability curve for a system whose scalability adheres to the Super-seriality model of scalability.

Gunther's Super-seriality model [7] may be expressed as below [8]:

$$C(N) = \frac{N}{1+\sigma((N-1)+\lambda N(N-1))}$$

Here $C(N)$ is the relative capacity of the system at N users. The parameter σ represents the level of contention in the system. The parameter λ represents the degree of coherency of the system – that is, the level at which updates in other processes cause cache misses and hence the phenomenon sometimes termed 'thrashing'. In the case where the super-seriality factor $\lambda = 0$, the equation reduces down to the conventional formulation of Amdahl's Law.

The equation above may be refactored as

$$Y = \sigma\lambda X^2 + (\sigma\lambda + \sigma)X$$

by making the substitutions

$$Y = N/C - 1$$

and

$$X = N - 1$$

to give a quadratic in X as shown in figure 12.

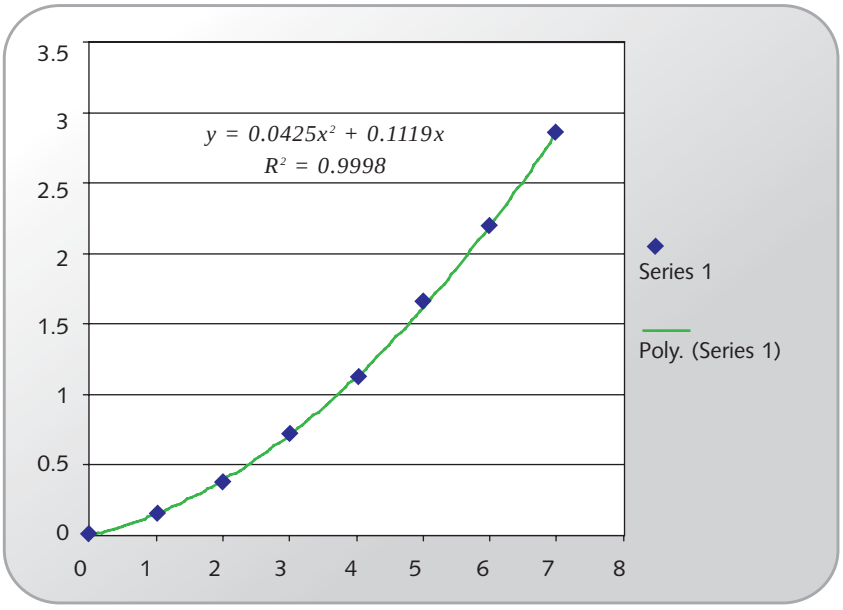

Figure 12: Quadratic fit of the recorded data to the refactored expression of Gunther's Super-seriality model.

Given a regression fit of a model to a set of data, the R^2 value – the 'coefficient of determination' – is a relative indication of how well the model describes the data, and therefore the degree of confidence with which the model may be used to predict values outside the range of those measured. A value of 1, very rarely encountered in real-world scenarios, indicates a perfect model, whereas a value of zero indicates that there is no correlation between the model and the observations.

The extremely high R^2 value of 99.98% indicates that 0.02% of the recorded effect is unaccounted for by the model. This is an extremely high degree of correlation between the recorded statistics and a theoretical model.

This allows the values of the parameters σ and λ to be calculated as:

Contention parameter	σ	0.0694
Coherence parameter	λ	0.612392

Finally the original recorded data may be compared directly with the original expression of the Super-seriality model in figure 13 below.

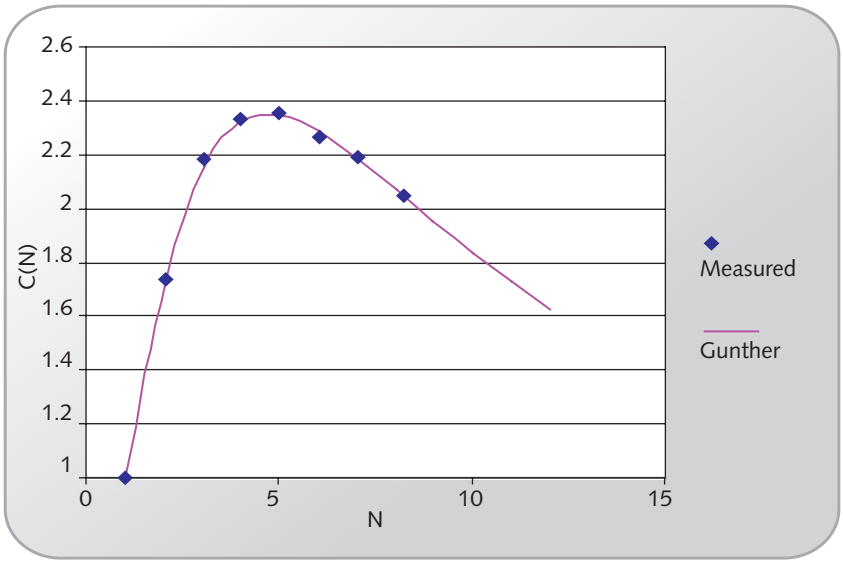

Figure 13: Comparison of recorded data with Super-seriality model for a single server element.

Syn–
Business Positive

London

Coexis Limited
Victoria House
Second Floor
64 Paul Street
London EC2A 4NG
Tel: +44 (0) 20 7613 8800
Fax: +44 (0) 20 7033 1965

New York

Coexis Inc
75 Broad Street
New York
NY 10004
USA
Tel: +1 646 649 9380
Fax: +1 646 649 9381
www.syn.com
Email: syn@coexis.com

Partner

Serisys Solutions Ltd
1201 Jubilee Centre
18 Fenwick Street
Wanchai
Hong Kong
Tel: +852 (2376) 3232
Fax: +852 (2376) 3030
www.serisys.com