AUTOMATED PERFORMANCE MANAGEMENT ADVICE USING COMPUTATIONAL INTELLIGENCE

Adam Grummitt
Metron Technology Limited
adam@metron.co.uk

This paper discusses the demand for a practical approach to automate performance analysis and capacity planning for typical enterprise-wide systems. Users of performance management tools now expect the exercise to be simple and readily automated. The full range of Computational Intelligence (CI) techniques available to achieve this is described. Previous attempted solutions exploiting expert systems are used to demonstrate the problems. A pragmatic approach is described, based on using combinations of established statistical and CI techniques in a particular case study.

1. PERFORMANCE MANAGEMENT

Performance Management is the computer-assisted control of all aspects of the performance of a computer system. Along with other functions, such as resource management, security management and privacy management, it forms a key part of System Management.

Performance Management has been defined in many ways over the years. A simple outline is that it is like a "MAP" to guide you to effective use of Information Technology (IT), in that it deals with the Measurement, Analysis and Planning of computer performance. The same "MAP" also enables you to Monitor the present, Analyze the past and Predict the future. Clearly key findings need to be broadcast effectively. Actions need to be taken in the light of the observations, interpretations and predictions, with tracking of the actuals to prove and refine the planning function.

The performance analyst may spend a certain amount of time tuning a system to overcome problems. Equally, rogue modules in application software can be optimized to avoid unnecessary resource demands. But there comes a point where these activities become self-defeating, with diminishing returns because the system itself is saturated.

Capacity planning, the art of designing and configuring a computer system to avoid saturation and achieve predetermined performance targets, is well established for conventional systems.

The performance analyst's role is to evaluate hardware configurations, analyze workload throughput, transaction response times and device utilizations, tune for optimal performance, define configuration updates to meet any immediate problems and to plan for potential expansion (or contraction) of workload or configuration in the light of business forecasts.

2. COMPUTATIONAL INTELLIGENCE

Computational Intelligence (CI) may be thought of as "Artificial Intelligence" (AI) but that is a somewhat discredited term. AI could have been better called synthetic intelligence. AI may be thought of in terms of expert systems, but successful expert systems have proved elusive in general domains. CI may be thought of as the application of Knowledge-Based Information Systems (KBIS) theory, but that too is more of an academic discipline rather than a proven commercial software technique.

CI is the application of computational techniques to create agents that behave in an apparently intelligent manner. The agent does what is appropriate for its goal, is flexible to changing environments, makes appropriate choices from any conflicting input and sometimes learns from experience [1].

The agent has to take on board knowledge about its world and past experience to learn from, as well as goals that it must try to achieve and observations about the current environment.

Intelligent agents in the past have tended to be of four main types – the robotic (factory or hazardous environment), the deductive (chess), the diagnostic (medical or other specialist expertise) and those that make use of natural language (translation or reference systems).

The most successful expert systems to-date have had the advantage of a "small world" or "tight domain" with a solid base of proven observations and expertise and limited goals.

The knowledge base for each application would typically be huge, requiring a formal means of defining the information. The instance data of given parameters for the case in hand would usually be constrained to be comparatively small.

3. PERFORMANCE MANAGEMENT & CI

There are some areas in which the environment has been so tightly defined that CI techniques can be exploited. This is in real terms, not in the fashionable use of the latest jargon that allows marketing spin to influence product descriptions. Old examples of such spin were "interfaces" described as "integration" or "linked files" suddenly becoming "relational databases". Likewise, knowledge-based systems are more than agents which "know" the environment and so "select" the relevant metrics (i.e. merely have a domain specific file layout). Neural networks are more than agents that merely happen to have their data collection metric definitions structured as tables. Expert systems and rule-based systems are still evolving and incorporate more than decision tables and domain specific information.

Computer performance might initially be thought of as having most of the properties that would suit it to the application of CI. Certainly the instrumentation and metrics are reasonably constrained and defined. However, the expertise and experience for each domain is changing so quickly that it is a challenge to create the knowledge base before the system has moved on to a new architectural release.

Furthermore, the instance dataset is huge, which makes the problem more difficult. The old monitor listing from a mainframe filling a desk has been replaced by numerous tables of data on the machine itself. The instrumentation is still voluminous and requires major interpretation skills. These are rare and so automated systems are in demand.

As applications are increasingly based on standard commercial packages, many sites now have no in-house development, little systems software expertise or awareness of systems management let alone performance management. Sadly there is no kudos associated anymore with acquiring such computer expertise. Thus sites in the distributed world, when they experienced performance problems in vital applications, had to find new performance specialists to apply performance management techniques.

These neophyte performance specialists brought a more business-driven approach. The challenge was not to try to produce the most detailed analysis of the characteristics and components of a computer system, but rather an indication of its behavior sufficient to give reasonable answers to business questions. Tolerances in accuracy of 10-20% became more acceptable, and the detail of particular machine components or application transactions less significant. What was more vital was a pragmatic approach with as much automation as practical, given the large numbers of nodes to be addressed and the small amount of time available for each.

Further, as hardware prices fall, and incremental upgrades apparently become easier, so there is an increasing superficial belief that performance management is less critical. Yet the true total IT expenditure is still increasing and the financial leverage for performance management is greater than ever.

The typical management request is for a shrink-wrapped solution, with minimized staff costs. The assumption is that performance can be built into the system itself or else provided automatically without requiring rare skills or expensive tools. The computer-literate manager might well also express the requirement that the system be expert so that it would make appropriate diagnostic analysis and provide intelligent advice.

4. ARTIFICIAL INTELLIGENCE

The performance management applications typically envisaged for AI techniques are in the areas of alerting, metric analysis and correlations, tuning and optimization and analytical modeling.

The intelligent selection of alerts is a good area for AI given that it is a constrained problem. It requires more than just Boolean combinations of threshold conditions and use of neural networks and pattern recognition over periods of time. It enables effective exception reporting regimes to be incorporated into automatic management reports with improving filters and better exception handling built over time.

Similarly, the intelligent analysis of metrics to reveal significant bottlenecks is a potentially good area for AI.

Intelligent advice for tuning and optimization is a dynamic area as systems evolve with new features and releases. It requires a huge library of up-to-date knowledge of the detail of a domain. Traditional approaches to this area tend to yield verbose pages of advisory text, which require the reader to already understand the problem in detail just to be able to comprehend the text. It may often be easier for such an expert reader to just look at the metrics directly without comprehensive automated advice. However, AI systems that focus on the key areas to address and highlight outstanding measures can save investigative time. Various commercial examples of automatic advice and guidance systems for tuning exist. ES/1 Advisor JP was the catalyst for this paper [2].

Experience of developing an AI system to review the results of analytical models and their interpretation with respect to defined Service Level Agreements (SLA) showed the difficulty of presenting results in a friendly way [3].

Figure 1 below shows typical expert advice for a particular node. It is based on assessing the results of an analytical model for a particular message pair in a particular transaction as part of the on-line workload. In the enterprise-wide system of today, many systems are viewed as application servers, without too much business interest in the finer analysis of the workload components. Thus such early efforts at exploiting expert systems were not widely adopted.

```
Metron's ATHENE    V1.0       Decision Support                                        3900

Component  Type    Required       Desirable      Baseline       Projected Response Times <secs>
                   Resp.Time      Resp.Time      Resp.Time      1              2              3
                   <secs>         <secs>         <secs>         ONE            TWO            THREE

SYSTEM     tp        -              -              0.05           0.05           0.05           0.05
SPOOLERS   tp        -              -              0.44           0.44           0.44           0.46
TP1        tp      0.80           0.70           0.64           0.70           0.80           1.09
TP2        tp      4.50           4.00           2.25           2.56           3.20           5.28
BATCH      ba    725.00         700.00         574.10         624.15         682.91         762.70

                              Explanation of Causes

Increase in arrival rate of TP1 affected
    Itself by increasing:
        Queueing at the CPU    by  10.05%
        Queueing at DA20       by  23.11%
        Queueing at DA22       by  17.85%
    TP2 by increasing:
        Queueing at the CPU    by  14.28%
        Queueing at DA20       by  21.94%
        Queueing at DA22       by  19.58%
Increase in MPL of BATCH affected
    Itself by increasing:
        Queueing at DA23       by   9.42%

                              Summary of Recommendations

1   Move 100% of the I/O generated by TP1 from DA20 to DA21 to reduce:
        Queueing at DA20, reducing the response time of itself and TP2
2   Move 100% of the I/O generated by TP2 from DA22 to DA12 to reduce:
        Queueing at DA22, reducing the response time of itself, TP1 and BATCH
3   Upgrade the CPU from 3980 to SX450-10 to reduce:
        Busy time at the CPU, reducing the response time of TP1, TP2 and BATCH
    This expensive operation is only necessary if you feel that the response time of
    BATCH really must be better than within 5% of the target.

                              Benefits of Recommendations

Workloads that     Target             Excess             New                % Reduction of
Exceeded Targets   Response <s>       Response <s>       Response <s>       Excess Response

TP1                  0.70               0.40               0.67               107.5
TP2                  4.00               1.28               3.18               164.1
BATCH              700.00              62.70             718.29                70.8
```

Figure 1. A typical piece of advice based on the results of a model of current behavior, with precise definition of the reasoning attached. Although the advice is sound and the presentation logical, it requires an experienced user to interpret it carefully - and such a user would possibly prefer to control the model making and interpretation directly.

The user interface for analysis of newer distributed systems is now also required to present a high-level overview with the facility to drill down into the detail of a particular node or sub-system as required. However,

the numbers of processors and disks on some newer machines make the presentation of the results in an accessible way somewhat challenging. Perhaps the adoption of huge screens, 3D graphics, surface envelope plots or even virtual reality will eventually help to meet the challenge. In the meanwhile, other techniques such as wizards are emerging to help.

Wizards can be useful training aids and also good for rarely used functions, but tend to "get in the way" of the experienced user. The best wizards allow for this by enabling both "beginner" and "expert" modes of working, or else allow the user to move from the wizard to a "tab & form" screen-fill style of entry.

Wizards can ease the learning curve for new packages, as can some expert advice and guidance built into hypertext help and improved forms of computer based training exploiting multi-media and animation.

Clearly the success of Big Blue in chess has improved the AI system image - but that is a highly constrained domain, where the rules are clear and there is a vast library of experience embodied in classical games. So the ability of the program to look at numerous options and consider so many moves ahead and assess them in the light of preferred patterns and known situations is confinable. Furthermore, the presentation of the result is embodied in a single simple command to move a selected piece to a given square.

5. EXPERT SYSTEMS

Chess is a highly specialized use of expert systems. Other uses where success has been achieved, are somewhat rare and tend to lie in the medical world. Even there most systems fail to be reliable. They are typically described as being "brittle", requiring expert interpretation and serving essentially as prompters to the diagnostician. Yet they have the advantages of typically a single problem or condition, with a small number of observable metrics and a larger number of subjective signs and symptoms, to be compared with a vast library of known case histories.

Expert systems normally have a large knowledge base of known facts and a small number of precise and/or qualitative observations. A significant rule base is used for the interpretation of one against the other with an inference engine selecting the most likely diagnosis.

Past experience of building an expert system for Software Performance Engineering (SPE) showed the challenge of knowledge acquisition and also the difficulty when a large fact base has to be presented to an inference engine [4].

This is indicated in figure 2 below.

```
Metron's PERSEUS       8.00    TP Message

Application    : IND1TLM3
Message Pair   : E-ADD-A        Type: Query

Data Description                User/Default
                                Values

Significance                    H
Multiphase                      Y
Computing Intensity             H
Programming Language            COBOL
Av.Day Think/Type Time (s)      30
Av.Day Message Pairs/Hr         1
Av.Day No. Active Terminals     1
Av.Day Desired Response (s)     0.5
Av.Day Maximum Response (s)     1.0
Av.Day Interval (s)             3600
```

```
Pk.Day Think/Type Time (s)      30
Pk.Day Message Pairs/Hr         8
Pk.Day No. Active Terminals     1
Pk.Day Desired Response (s)     1.0
Pk.Day Maximum Response (s)     2.0
Pk.Day Interval (s)             450
Conv.I/O Logical Reads          0
Conv.I/O Logical Writes         0
Conv.I/O CAFS searches          1
IDMS I/O Keyed Find/Obtain      0
IDMS I/O Keyed Store            0
IDMS I/O Keyed Modify           0
IDMS I/O via Find First         0
IDMS I/O via Oth Find/Obtain    0
IDMS I/O via Store              0
IDMS I/O via Modify             0
IDMS I/O via Erase              0
IDMS I/O via Connect/Disconn    0
IDMS I/O CAFS Searches          0
Total DML statements obeyed     0
```

Figure 2. A typical pair of input screens to define part of the logical specification of an application for SPE.

The advantage we have in performance management is that it is a constrained domain, usually with precise and voluminous instrumentation (although not always including the key metrics). The knowledge acquisition phase for SPE clearly requires the extended presence of experts in the area. Also in the case of SPE, the facts to be presented for interpretation are also unusually large.

Figure 2 above shows parts of the first two of three sample screens to be completed for each key message pair. Similar detail is required of the database and other aspects of the system. This is a significant demand on the quality of a typical logical specification and is not always readily available

6. KNOWLEDGE-BASED INFORMATION SYSTEMS

KBIS have developed to exploit the structured representation of knowledge. This has moved in the same direction as software in general, towards object definition. Each object has attributes, relationships with other objects and is part of a higher group or class. The same object approach is adopted for data modeling and object oriented programming. In data modeling, entities are defined via normalization techniques. An entity is an object. The type of entity is the class of object. An object-oriented approach has four main features:

- Data abstraction
- Attribute inheritance
- Data encapsulation
- Dynamic binding

Data abstraction extracts the data definitions into classes and objects. It thus maps the problem domain (data) and the system's responsibility (procedures) from the real world.

Attribute inheritance allows for the definition of two different classes in terms of their common base set and then two new classes just logging the defined differences. Inheritance allows representation of commonality to maximize code re-use.

Data encapsulation is the incorporation of both the object definition (instance attributes) and the methods that define an object. This assumes that the methods and attributes are unique (effectively that data modeling and entity analysis have taken place). Encapsulation facilitates resilience to change.

Thus data and code are in the same object, sometimes called "information binding" with the private properties controlled by the object itself. Dynamic binding is the creation of derived classes during run-time (as opposed to compilation time).

Figure 3 below shows the typical structure of a KBIS. Essentially the input data has to be fed into an inference engine where it is assessed against knowledge in the rulebase. An explanation module then presents the conclusion in a report.

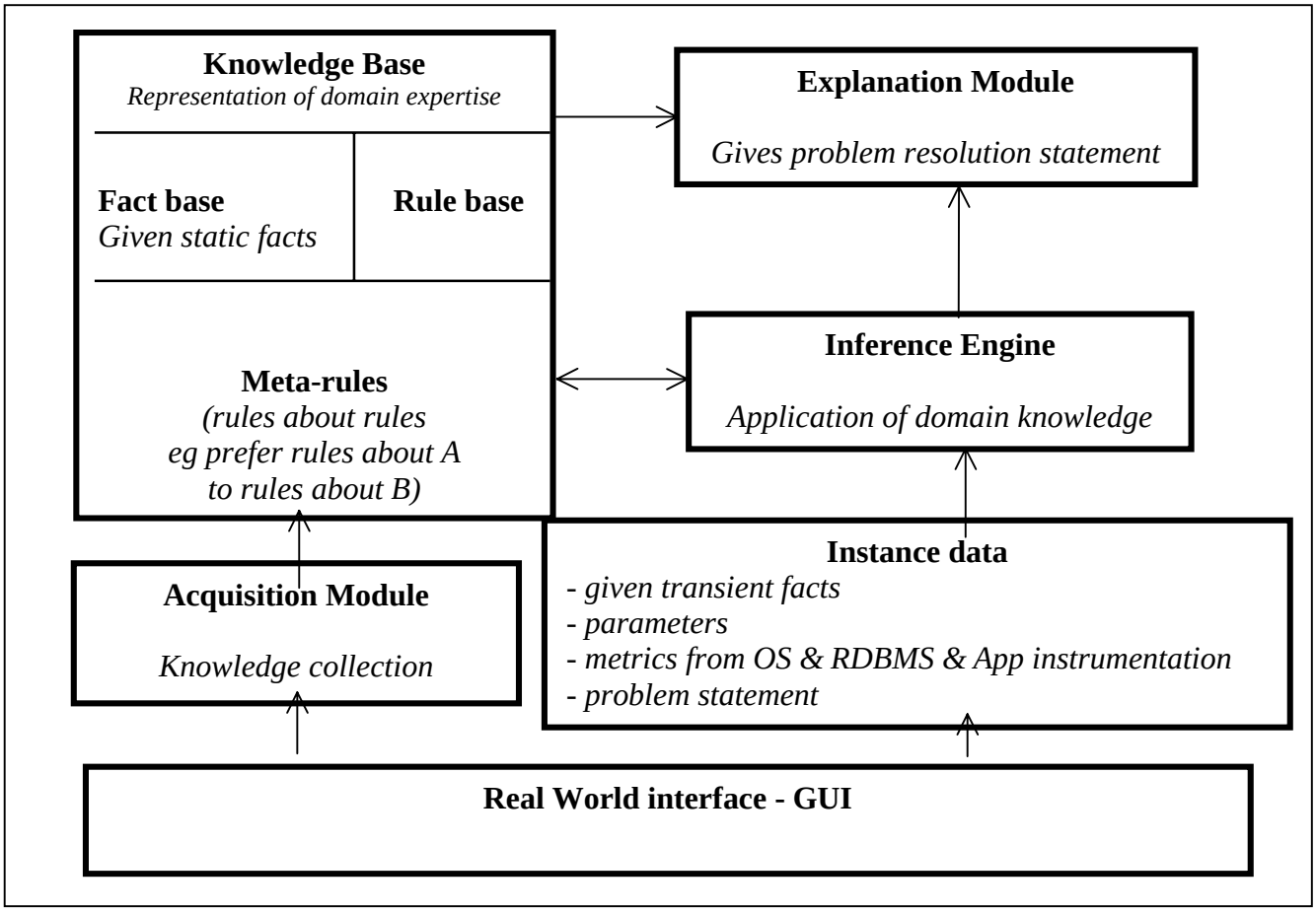

Figure 3. Knowledge-Based Information System Structure

A knowledge base is a set of production rules with variables (also referred to as assignments or unification) – typically:

IF <condition> THEN <conclusion>
IF the tap on the barrel is open THEN wine flows

For a particular problem:

- Select the available rules
- Determine the applicable set
- Resolve any conflicts
- Select a rule to "fire"

Uncertainty lies in the rule itself, the evidence offered or the use of vague language. "The bottle is large" is vague. The real question is "How many glasses can I pour?" Thus the wrong statement and the wrong set were selected.

The logic to process such knowledge falls into either of two main classes, Bayesian updates or fuzzy logic.

Bayesian updating is a means of handling uncertainty by assessing the probability of an assertion when evidence for or against the assertion is provided. Thus it assesses the likelihood that a hypothesis arises either from statistical variations or from randomness. Bayesian updating enables combinations of dependent probability to allow for combining evidence [5].

Possibility theory leads to fuzzy logic and fuzzy sets. This addresses vagueness in the use of language and looks at the meaning of a hypothesis. Consider set theory with a variable (say wine temperature) and mutually exclusive sets "high", "right" and "low". "The temperature is high" is a fuzzy statement, with temperature being the fuzzy variable and high the fuzzy set.

7. MACHINE LEARNING

This discipline tends to be split into two areas, symbolic and numerical. The symbolic covers the formulation and modification of explicit rules, facts and relationships. The numerical applies where the system can be numerically instrumented.

Machine Learning can be either supervised (by a "teacher") or explorative (unsupervised). The latter is where the system continuously searches for patterns/ relationships in the input data. The former is more relevant to this paper with a variety of approaches:

- Rote – spoon-fed correct decisions
- Advice – general advice with likelihood attached
- Induction – specific examples and conclusions are used as a basis for improvement
- Analogy – specific examples and conclusions are used as a basis for extrapolation in similar tasks
- Explanation Based Learning – where analysis of a set of given examples is used to guide future problem solving
- Case Based Reasoning – previous experiences are filed with outcomes and the system recalls and adapts them to resolve new circumstances
- Model Based Reasoning – where models of how components operate and combine to form a device are constructed from both first principles (where the basic laws of physics determine apparent behavior) and second principles (where reasoning is based on consideration of component behavior)
- Neural Networks – where each node performs a simple computation operation independently of the other nodes. This facilitates a parallel structure that is clearly good for massively parallel processing. It is good for yielding approximate answers within the range of examples, but not for extrapolation outside. For that, fuzzy logic is required, or an appropriate algorithm (if a formal relationship can be defined).

8. STATISTICAL METHODS

Using the term CI loosely to imply the illusion of intelligence, traditional statistical methods can be exploited to extract the key information from large sets of data. Early approaches to computer guidance were based on the use of statistical methods. These methods are well established and yet still provide good "advice" based on simple mathematical algorithms [6].

Statistics start with the observation of a variable, which is either a Category or an Ordinal. A category variable is nominal (e.g. a Mondavi bottle) whereas an ordinal variable is a metric (such as 75 cl of Merlot).

Variable

Sample observations of a single variable form a set of values. Consider a set of data x_1, x_2, x_3...x_N where N is the number of data items.

More practically, consider a group of seven at a dinner estimating how many centiliters of wine there are in a particular glass. The 7 estimates are 12, 15, 15, 17, 20, 22 and 25.

The basic information for a single variable is:

- N The number of observations in the analysis (7)
- The sum of the data (126)
- The minimum value of the data (12)
- The maximum value of the data (25)
- $\bar{x}$ The mean of the data (18) where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

- σ^2 The variance of the data (17.71) where $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$
- σ The standard deviation of the data (4.21) where $\sigma = \sqrt{\sigma^2}$
- AD The average deviation of the data (3.7) where $\mathrm{AD} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \bar{x}|$

The mean $\bar{x}$ is equivalent to the average and is defined by the sum of the values divided by the number of samples. The other three statistics below the mean on this list are measures of variation. As a reminder, the other expressions for the "middle" of a set are the Mode, which is the most frequent observation (15), and the Median, which splits the distribution into equal halves in terms of observations (17).

The variance σ^2 is a measure of the scatter around the mean and is defined as the mean of the squared deviations about the average. Variance σ is useful for comparing two distributions. The Standard Deviation is another measure of the same scatter, and is defined as the square root of variance. The average deviation AD is the mean of the deviations about the average.

The following examples are all based on a case study. A site installed and ran performance agents on their system for a couple of weeks and returned the collected data for technical investigation and a consultancy report. This was undertaken both manually and automatically.

The manual investigation had the advantage of being able to exploit both spreadsheet and graphical options interactively. The first step was to review the entire dataset presented.

Figure 4 below shows a typical set of such data presented for the CPU utilization of a single node over the given period of time. Figure 5 shows the drill-down to the selected day and busy period for analysis, including a set of observation statistics.

Figures 6 & 7 show extracts from the automatic reports generated against the same data.

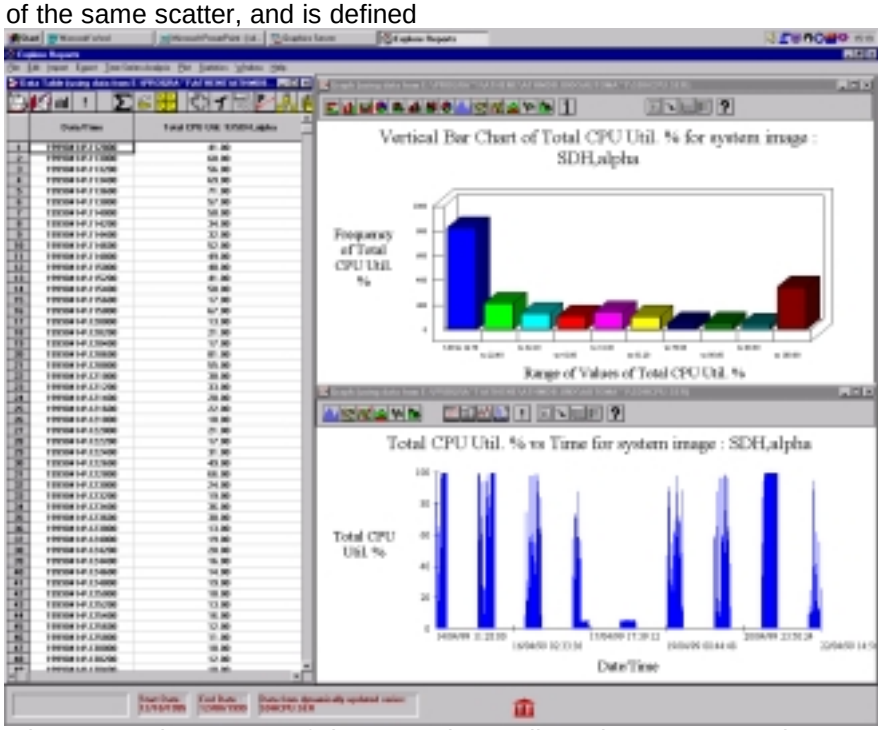

Figure 4. Shows part of the CPU data collected over two weeks as a spreadsheet (left). The CPU series and its date/timestamp have been selected for plotting both as a frequency distribution or histogram (top right) and a time series plot (bottom right). The frequency distribution shows a typical decay from low utilization to occasional high utilization but with an unusual peak at 90-100%. The time series clearly identifies 20 April as the busiest day.

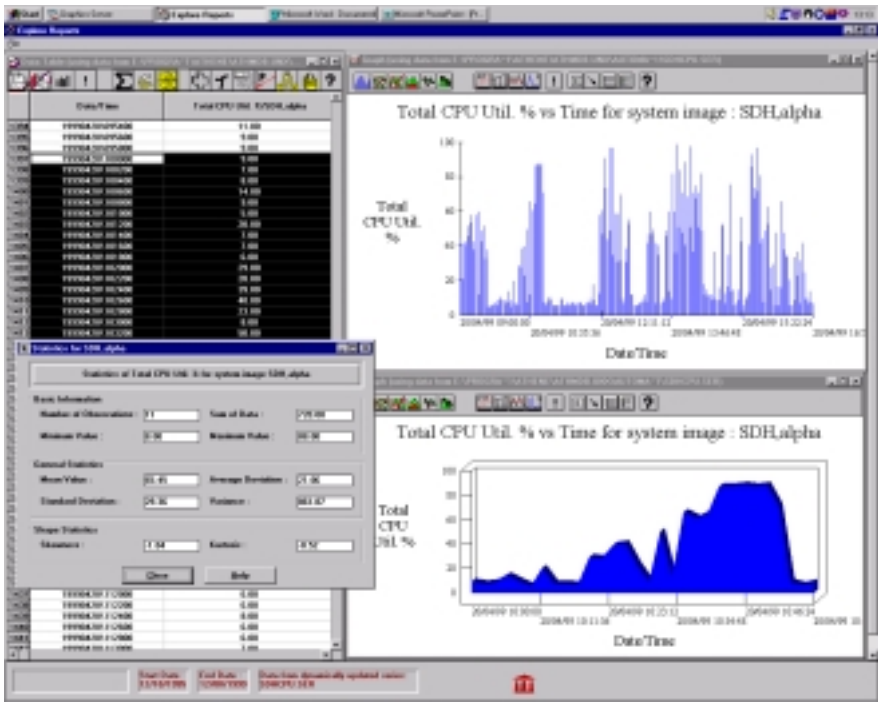

Figure 5. Shows the selected day's CPU utilisation spreadsheet (top left). A plot over the period 9-5 is at top right, showing 4 apparently busy peaks (at tea-breaks at 10.30, 12, 2 & 3.30?). The period from 10-11 was selected and plotted (bottom right), together with the statistics for the observations during that period (bottom left).

SYSTEM PERFORMANCE REPORT
FOR INSTALLATION SDH, SYSTEM IMAGE ALPHA
ANALYSIS PERIOD 20/04/99 09:00 - 20/04/99 17:00

CPU SUMMARY

Total CPU Util. (%)
Highest value of 99.00 (%) occurred at 20/04/99 13:54
Lowest value of 5.00 (%) occurred at 20/04/99 09:38
Average Value of 31.50 (%)

I/O SUMMARY

Physical Read/Writes (/sec)
Highest value of 62.00 (/sec) occurred at 20/04/99 15:40
Lowest value of 2.00 (/sec) occurred at 20/04/99 11:08
Average Value of 7.08 (/sec)

USER SUMMARY

CPU Total (secs): Busiest Users over the Measurement Period
Total of CPU Total (secs): 8347.53

UNIX User	CPU Total (secs)	% of CPU Total (secs)	Cumulative %
oracle	4153.00	49.75%	49.75%
delphi	2699.94	32.34%	82.09%

COMMAND SUMMARY

CPU Total (secs): Busiest Commands over the Measurement Period
Total of CPU Total (secs): 8348.54

UNIX Command	CPU Total (secs)	% of CPU Total (secs)	Cumulative %
oraclelive	4021.00	48.16%	48.16%
oracle	846.35	10.14%	70.09%
dfrunreport	829.62	9.94%	80.03%

Figure 6. Extract from the Automatic System Performance Report Highlights only are shown. The busiest interval is also automatically presented with similar statistics The top items contributing 50% were selected. This report also highlights 20 April and draws attention to oracle and delphi users and the commands oraclelive, oracle and dfrunreport.

ORACLE SYSTEM PERFORMANCE REPORT
FOR INSTALLATION SDH, SYSTEM IMAGE WORLD+LIVE
ANALYSIS PERIOD 20/04/99 09:00 - 20/04/99 17:00

SYSTEM SUMMARY

Total CPU Util. (%)

Highest value of 92.30 (%) occurred at 20/04/99 15:42
Lowest value of 0.60 (%) occurred at 20/04/99 09:38
Average Value of 15.37 (%)

FILE I/O SUMMARY

Physical Reads/Writes (/sec): Busiest Files over the Measurement Period
Total of Physical Reads/Writes (/sec): 3105847.84

Oracle File	*Physical Read/Writes (/sec)*	*% of Physical Read/Writes (/sec)*	*Cumulative %*
/midland/oracle/live/data/data_ts02.dbf	100.69	93.37%	93.37%
/midland/oracle/live/data/data_ts01.dbf	6.41	5.94%	99.31%

SESSION SUMMARY

Session Logical I/O (/sec): Busiest Sessions over the Measurement Period
Total of Session Logical I/O (/sec): 13700.09

Oracle Session	*Session Logical I/O (/sec)*	*% of Session Logical I/O (/sec)*	*Cumulative*
DELPHI, 16, 317, delphi, 9482, oracle@ALPHA (TNS V1-V2)	845.77	6.17%	6.17%
DELPHI, 15, 1521, pridpath, 23366, oracle@ALPHA (TNS V1-V2)	501.68	3.66%	9.83%
DELPHI, 16, 16108, delphi, 525, oracle@ALPHA (TNS V1-V2)	482.24	3.52%	13.35%
DELPHI, 14, 705, delphi, 7557, oracle@ALPHA (TNS V1-V2)	472.22	3.45%	16.80%
DELPHI, 21, 1019, pridpath, 28306, oracle@ALPHA (TNS V1-V2)	461.32	3.37%	20.17%
DELPHI, 15, 1976, ajames, 13905, oracle@ALPHA (TNS V1-V2)	452.74	3.30%	23.47%
DELPHI, 15, 1736, pridpath, 29371, oracle@ALPHA (TNS V1-V2)	447.27	3.26%	26.73%
DELPHI, 6, 601, apratt, 30825, oracle@ALPHA (TNS V1-V2)	446.17	3.26%	29.99%
DELPHI, 21, 346, jsheen, 15725, oracle@ALPHA (TNS V1-V2)	429.79	3.14%	33.13%
DELPHI, 16, 14207, ajames, 6380, oracle@ALPHA (TNS V1-V2)	420.20	3.07%	36.20%
DELPHI, 21, 884, jsheen, 6425, oracle@ALPHA (TNS V1-V2)	419.34	3.06%	39.26%
DELPHI, 20, 318, pridpath, 30858, oracle@ALPHA (TNS V1-V2)	405.10	2.96%	42.22%
DELPHI, 15, 1687, apratt, 29957, oracle@ALPHA (TNS V1-V2)	386.82	2.82%	45.04%
DELPHI, 21, 906, parcher, 225, oracle@ALPHA (TNS V1-V2)	385.58	2.81%	47.85%
DELPHI, 6, 1579, jsheen, 13718, oracle@ALPHA (TNS V1-V2)	378.32	2.76%	50.61%

Figure 7. Extract from the Oracle System Performance Report. Again selecting 20 April and highlighting delphi and other users' busy sessions.

Distribution Statistics

Standard Deviation is the statistic most often used to describe variation in data because of its relationship with the normal distribution. If the data describes a normal distribution (i.e. bell shaped or Gaussian as first derived by de Moivre) then it has a standard shape. Approximately 68% of the data lies in the range mean ± one standard deviation; approximately 95% lies in the range mean ± 2 standard deviations and approximately 99.7% lies in the range mean ± 3 standard deviations. In this case, the three ranges are 14-22, 10-26 and 6-30.

The "average deviation" is a measure used in some cases and is more robust than the standard deviation in the case of a distribution with a number of outlying points, because it is less affected by unusual values.

Shape statistics give an indication as to the shape of a given distribution being analyzed.

Skewness is a measure of asymmetry with positive skewness indicating a tail towards the right (positive x-direction) and negative skewness indicating a tail towards the left (negative x-direction).

- Kurtosis is a measure of the degree of peaking of a distribution, positive values indicating a sharp peak and negative values indicating a flat-topped distribution.

Significance

The significance of a finding about two variables can be assessed statistically using probability theory. This is usually approached obliquely via the Null Hypothesis. Ask not "Are they related?", ask rather "What is the statistical significance of the reliability of the assumption that they are not related?". So if we are to compare two variances to see if they have a significant correlation, we put forward the null hypothesis that they are not related and define the probability of that statement being rejected [7].

Correlation

Correlation concerns the strength of the relationship between the values of two variables. It only implies the fact that there is an apparent similarity in the two series, not that there is any causal link between them. The aim of multiple correlation is to find a suitable equation to explain two or more variables. In the general case the equation is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \mathrm{K} + \beta_n x_n$$

The x_i's are the predictor variables whereas *y* is the dependent variable. Multiple correlation is the process by which the *linear correlation coefficient* between each constituent variable is calculated. This results in a correlation matrix.

Figure 8 below shows the initial manual correlation identified by selecting the key users and plotting their CPU utilizations against the total system CPU utilization. There is a clear relationship in terms of being synchronized such that the peaks in the predictor variable contribute to the peaks in the dependent variable.

These series can be investigated using statistics, but it is always essential to use the right tools. Comparisons of means and variances in this case will show the apparent lack of a relationship (see figure 9). Comparison of spectral analyses will highlight the clear relationship between total CPU utilization and the CPU utilization of one of the key users (see figure 10).

Performance Analysis can be undertaken automatically using a "correlation engine". This undertakes correlation tests between all pairs of key metrics to identify any significant associations. After sorting on the significance values found, it presents a report with the key findings automatically highlighted, generating text based on the metrics involved.

Figure 11 below shows an automatic report where a correlation engine has been used to compare values of key metrics to search for any significant correlations. The primary application is the use of a correlation engine for highlighting areas as tuning targets. It is sorted into levels of significance to bring the tuner's attention to the key areas first. It reveals the same key relationships as identified manually.

Note that the significance tests on the correlations found have been banded into three groups (high, close and reasonable correlation).

Regression

Regression analysis determines the nature of a relationship between two variables and enables interpolation and extrapolation from it. This relationship can be assumed to be linear or polynomial. Figure 10 shows linear regression being applied to two series with confidence limits shown. The confidence limits can be set to 90%, 95% or 99% and expressed in terms of the predicted value of the next point in the series lying within a certain range or the confidence that a range of new points will have a mean within that certain range.

This can be tested by four main techniques:

- Linear correlation coefficients
- Spearmans rank correlation coefficient
- Sum squared differences of ranks
- Confidence intervals

The data series as presented may first require some manipulation. If it is excessively "noisy" then smoothing techniques can help reveal the underlying nature. Typically this is done by "moving average" or by Savitzky-Golay smoothing. Also, the data may have a few spurious values which can be neglected by "robust" curve fitting. If the data has missing points or requires extra values, then interpolation and extrapolation are used, typically "linear" or better by the "natural cubic spine" method.

Significance Testing

There are three main methods available to compare two data sets, to see whether the data are similar:

- t-test to compare means
- F-statistic to compare variances
- Kolmogorov-Smirnov (KS) statistic for both

Suppose we are working with two data sets. Set A is a_1, a_2, a_3 … a_N where N_A is the number of data in set A and set B is b_1, b_2, b_3…b_N where N_B is the number of data in set B. The variances are calculated and represented as σ_A^2 and σ_B^2 respectively.

The t-statistic (Student's t-test) is a measure of equivalence of the mean of the two data sets. Exactly equal means give a near zero value. This statistic is interpreted via the probability that two distributions with equal means could have a t-statistic as large as the one given by chance.

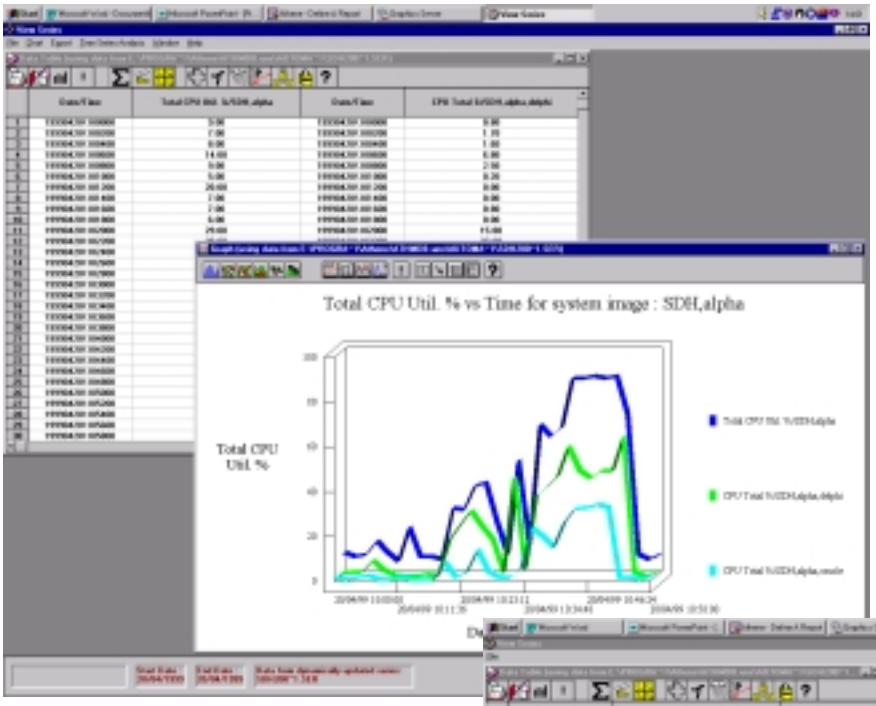

Figure 8. Manual correlation – looking at the plots for the three key CPU utilization columns (total system, total oracle, total delphi. There is a clear relationship with the contributions of the two key users to the total CPU readily seen.

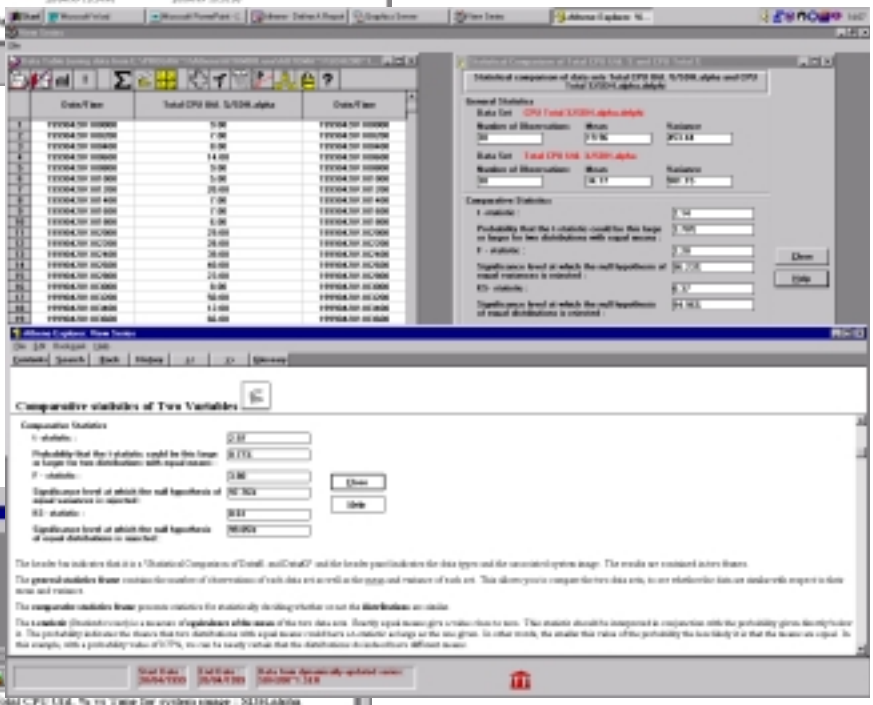

Figure 9. However, standard tests for means and variances will not show the relationship. The two series are shown in the spreadsheet on the left. The table on the right shows the three tests used, showing t approx 0 and F & KS approx 1 . The Help at the bottom reminds us that this implies they are not similar

Figure 10. What is necessary is a spectral analysis. This is only undertaken after first using linear regression to find a trend, detrending to get an oscillating series and then applying Fourier Analysis. In this case the two spectral analyses are clearly similar, with the statistics summary showing the two harmonic frequencies revealing patterns every 23/46 minutes (tea breaks again – perhaps dfrunreport pseudo-batch jobs initiated at brew-

CORRELATION REPORT
FOR INSTALLATION SDH, SYSTEM IMAGE ALPHA
ANALYSIS PERIOD 20/04/99 09:00 - 20/04/99 17:00

HIGHLY CORRELATED METRICS (100% - 95%)
No metrics were highly correlated.

CLOSELY CORRELATED METRICS (95% - 90%)
No metrics were closely correlated.

REASONABLY CORRELATED METRICS (90% - 70%)

Metric 1	Metric 2	Correlation Coefficient
Total CPU Util. (%)	CPU Total (secs) for UNIX User oracle	84.5%
Total CPU Util. (%)	CPU Total (secs) for UNIX Command oraclelive	83.62%
Total CPU Util. (%)	CPU Total (secs) for UNIX User delphi	70.75%

Fig 11. Automatic Correlation shows the three most significant correlations for this installation. Again it draws attention to the oracle and delphi users and the command oraclelive.

The first step in the calculation of t is to derive the standard error of the differences of the means S_D. The t-statistic is then:

$$t = \frac{\bar{a} - \bar{b}}{S_D}$$

The F-statistic is a measure of the equivalence of the variance of two data sets. It is essentially the ratio of the two variances and hence values near 1 indicate equal variances. The F-statistic is then:

$$F = \frac{\sigma_A^2}{\sigma_B^2}$$

The KS statistic is a measure of the overall equivalence of the data sets in two dimensions. The mean, variance and time are all taken into account. Equal data sets will have values close to zero. The probability associated with this statistic indicates the significance level at which the null hypothesis of equal distributions is rejected. High values indicate that the data sets are indeed unequal. Suppose we have a set of data representing CPU % Idle measurements numbered from 1 to N. Then, if the data are sorted into ascending order, a cumulative distribution function can be defined for the data which indicates the fraction (or percentage) of data to the left of a given value. The KS statistic is simply the maximum difference between two such cumulative distributions of two data sets. The advantage of this statistic is that significance levels may be calculated for it.

Other statistical techniques are important to time-based series, such as signal or spectral analysis. The primary technique is Fourier analysis, which can be used to identify any seasonal variations in a series of values. Figure 10 above shows the power spectrum with the dominant cycles and their wavelengths for the two series concerned.

9. PRACTICAL APPLICATION OF CI

It is useful to select from all the above options in CI to find the most practical to apply to the required areas of performance management. Ideally in any area, it would be possible to apply both deduction and abduction (deduction predicts an effect from a given cause; abduction finds a cause given a particular effect).

New tools need to be intelligent, provide advice, and attempt to be as automated as practical. This does not suggest that tuning, optimization and capacity planning can be handled totally automatically. Experience of expert systems in the computer world and elsewhere seem to indicate that they are best used as support tools for the educated user. This is somewhat paradoxical, given their objective was originally seen as being able to incorporate the expertise in the tool itself, so that “expert” domains are made accessible to the non-expert.

A distinction has to be made between those systems where the marketing material makes veiled references to knowledge bases and expertise and true expert systems which claim the formal use of AI techniques or knowledge-based information systems. The former may be an exaggerated description of procedural programming techniques where the result is context sensitive and considerable expertise has gone into the logic to define the conditions and filters used. The latter typically incorporates an inference engine.

Computers seem on the surface to obey chaotic behavior. However, taking larger populations and longer time-spans, it can be treated adequately as ordered. So statistical methods can be used with some validity, although boundary conditions can affect the results. There are two main approaches to handling this data. The stochastic approach assumes that all the parameters (e.g. the inter-arrival time of transactions and their service times) are random variables which can be characterized by given distributions and hence definable means. The

operational analysis approach relies on all quantities being measured or otherwise known data.

Automated alerts, interpretation of monitor logs, application of error diagnostics and many other areas have been addressed in the past. The use of KBIS in SPE & SLA has been mentioned. Yet few of these more advanced approaches have become major commercial applications. A more pragmatic approach is that of using statistical methods. Workload growth and current performance analysis can be addressed in this way. (Clearly for service level prediction, where contention and queuing can dominate, different methods such as analytical modeling are required.) Simple trends and regression analysis can be used to review workload growth, with confidence limits shown on the plot.

A few sample screens have been shown above. They demonstrate that with a standard statistical approach it is feasible to automatically isolate the exceptions that merit reporting and so focus the attention of the performance analyst on key areas.

10. SUMMARY

The traditional approach to performance management and capacity planning has to be modified to cater for new demands. The need is for fast tools to provide adequate detail to answer the technical questions arising from the need to maintain required service levels in the light of changing business circumstances. The tools need to incorporate expert systems to save new users' time and to act as a reminder to experienced users to check for possible alternative interpretations.

The tools need to cater for the complexities of client-server distributed systems and other enhancements to both hardware and software. Applications and frameworks need to provide the instrumentation to feed the data capture so that the end-user can see the results with minimal system interference.

The statistical analyses outlined in this paper can play a valuable role in isolating significant correlations and patterns of events. Disparate data can then be integrated into a format and language appropriate to the typical technical user. Thus it can enhance the timeliness of performance management and control. Further, such systems can be improved either by learning from its own experience or by experts applying their expertise to improve it, so that the skills shortage issue can be addressed.

Performance management is alive and well, but needs to heed new requirements. The objective is to provide fast analysis tools that can be applied automatically where accuracy is less important than gaining a general sense of direction. The evolution of statistical, intelligent and expert systems will gradually provide a sensible degree of automation and intelligent support for Performance Management and Capacity Planning across the enterprise.

ACKNOWLEDGEMENTS

I would like to thank Kawano San for his inspiration on this theme, Dr Uli Harder for trying to explain some of the maths to me, Andy Mardo for clarifying its significance, Liz Norris for creating the new CI code and Tim Foxon for providing the case study.

REFERENCES

[1] Computational Intelligence, Poole, Mackworth & Goebel ISBN 0-19-510270-3 OUP 1998

[2] Methodology for System Performance Evaluation, Kawano, CMG97

[3] Knowledge Based Decision Support for Capacity Management Tools, Lowe, CMG91

[4] Performance Engineering – Better bred than dead, Wilson, CMG91

[5] Knowledge-Based Systems for engineers and scientists, Hopgood, ISBN 0-8493-8616-0 CRC 1993

[6] Statistics without tears, Rowntree, ISBN 10-9-8 Penguin 1981

[7] Introduction to Statistical Modelling, Krzanowski, ISBN 0-340-69185-9 Arnold 1998