

1 **METHOD AND SYSTEM FOR DETERMINING COMPATIBILITY**
2 **OF COMPUTER SYSTEMS**

3 **[0001]** This application claims priority from U.S. provisional patent application
4 no. 60/745,322 filed April 21, 2006.

5 **FIELD OF THE INVENTION:**
6

7 **[0002]** The present invention relates to information technology infrastructures and has
8 particular utility in determining compatibility of computer systems in such infrastructures.

9 **DESCRIPTION OF THE PRIOR ART**

10 **[0003]** As organizations have become more reliant on computers for performing day to day
11 activities, so to has the reliance on networks and information technology (IT) infrastructures
12 increased. It is well known that large organizations having offices and other facilities in different
13 geographical locations utilize centralized computing systems connected locally over local area
14 networks (LAN) and across the geographical areas through wide-area networks (WAN).

15 **[0004]** As these organizations grow, the amount of data to be processed and handled by the
16 centralized computing centers also grows. As a result, the IT infrastructures used by many
17 organizations have moved away from a reliance on centralized computing power and towards
18 more robust and efficient distributed systems. Distributed systems are decentralized computing
19 systems that use more than one computer operating in parallel to handle large amounts of data.
20 Concepts surrounding distributed systems are well known in the art and a complete discussion
21 can be found in, e.g., "Distributed Systems: Principles and Paradigms"; Tanenbaum Andrew S.;
22 Prentice Hall; Amsterdam, Netherlands; 2002.

23 **[0005]** While the benefits of a distributed approach are numerous and well understood, there
24 has arisen significant practical challenges in managing such systems for optimizing efficiency
25 and to avoid redundancies and/or under-utilized hardware. In particular, one challenge occurs
26 due to the sprawl that can occur over time as applications and servers proliferate. Decentralized
27 control and decision making around capacity, the provisioning of new applications and hardware,
28 and the perception that the cost of adding server hardware is generally inexpensive, have created
29 environments with far more processing capacity than is required by the organization.

21562813.1

1 **[0006]** When cost is considered on a server-by-server basis, the additional cost of having
2 underutilized servers is often not deemed to be troubling. However, when multiple servers in a
3 large computing environment are underutilized, having too many servers can become a burden.
4 Moreover, the additional hardware requires separate maintenance considerations, separate
5 upgrades and requires the incidental attention that should instead be optimized to be more cost
6 effective for the organization. Even considering only the cost of having redundant licenses,
7 removing even a modest number of servers from a large computing environment can save a
8 significant amount of cost on a yearly basis.

9 **[0007]** As a result, organizations have become increasingly concerned with such
10 redundancies and how they can best achieve consolidation of capacity to reduce operating costs.
11 The heterogeneous nature of distributed systems makes consolidation ever more difficult to
12 achieve.

13 **[0008]** It is therefore an object of the following to obviate or mitigate the above-described
14 disadvantages.

15 SUMMARY OF THE INVENTION

16 **[0009]** In one aspect, a method for determining compatibilities for a plurality of computer
17 systems is provided comprising generating a configuration compatibility score for each pair of
18 the plurality of systems based on configuration data obtained for each of the plurality of systems;
19 generating a workload compatibility score for each pair of the plurality of systems based on
20 workload data obtained for each of the plurality of systems; and generating a co-habitation score
21 for each pair of the plurality of systems using the respective configuration compatibility score
22 and workload compatibility score, the co-habitation score indicating an overall compatibility for
23 each system with respect to the others of the plurality of systems.

24 **[0010]** In another aspect, a computer program is provided for determining compatibilities for
25 a plurality of computer systems. The program comprises an audit engine for obtaining
26 information pertaining to the compatibility of the plurality of computer systems; an analysis
27 engine for generating a compatibility score for each pair of the plurality of systems based on the

1 information that is specific to respective pairs; and a client for displaying the compatibility score
2 on an interface.

3 **[0011]** In yet another aspect, a method for determining configuration compatibilities for a
4 plurality of computer systems is provided comprising obtaining configuration data for each of
5 the plurality of computer systems; assigning a weight to one or more parameter in the
6 configuration data indicating the importance of the parameter to the compatibility of the plurality
7 of systems; generating a rule set comprising one or more of the parameters; and computing a
8 configuration score for each pair of the plurality of systems according to the weights in the rule
9 set.

10 **[0012]** In yet another aspect, a method for determining workload compatibilities for a
11 plurality of computer systems is provided comprising obtaining workload data for each of the
12 plurality of systems; computing a stacked workload value for each pair of the plurality of
13 systems at one or more time instance according to the workload data; and computing a workload
14 score for each pair of the plurality of systems using the stacked workload values.

15 **[0013]** In yet another aspect, a graphical interface for displaying compatibility scores for a
16 plurality of computer systems is provided comprising a matrix of cells, each the cell
17 corresponding to a pair of the plurality of computer systems, each row of the matrix indicating
18 one of the plurality of computer systems and each column of the matrix indicating one of the
19 plurality of computer systems, each cell displaying a compatibility score indicating the
20 compatibility of the respective pair of the plurality of systems indicated in the corresponding row
21 and column, and computed according to predefined criteria.

22 BRIEF DESCRIPTION OF THE DRAWINGS

23 **[0014]** An embodiment of the invention will now be described by way of example only with
24 reference to the appended drawings wherein:

25 **[0015]** Figure 1a is a schematic representation of a system for evaluating computer systems.

1 **[0016]** Figure 1b is schematic representation of a network of systems analyzed by a
2 compatibility analysis program.

3 **[0017]** Figure 2 is a schematic block diagram of an underlying architecture for implementing
4 the analysis program of Figure 1b.

5 **[0018]** Figure 3 is flow diagram illustrating a system consolidation analysis.

6 **[0019]** Figure 4 is a flow diagram illustrating strategies for a server consolidation analysis.

7 **[0020]** Figure 5 is a table illustrating data enablement for system consolidation and
8 virtualization.

9 **[0021]** Figure 6 is a system compatibility index (SCI) matrix.

10 **[0022]** Figure 7 is a workload compatibility index (WCI) matrix.

11 **[0023]** Figure 8 is a co-habitation index (CHI) matrix for the SCI and WCI matrices of
12 Figures 6 and 7.

13 **[0024]** Figure 9 is a graphical display showing server consolidation information.

14 **[0025]** Figure 10 is an audit request template.

15 **[0026]** Figure 11 is a detailed configuration report containing audited configuration data.

16 **[0027]** Figure 12 is a table containing workload data.

17 **[0028]** Figure 13 is a table containing a rule set used in generating an SCI matrix.

18 **[0029]** Figure 14 is a screenshot of a program for generating compatibility reports.

19 **[0030]** Figure 15 is an SCI matrix for an example environment having four server systems.

20 **[0031]** Figure 16 is a table containing a summary of differences between a pair of systems in
21 the environment.

- 1 **[0032]** Figure 17 is a table containing details of the differences listed in Figure 16.
- 2 **[0033]** Figure 18 is a WCI matrix for the environment.
- 3 **[0034]** Figure 19 is a workload compatibility report for a pair of systems being analyzed.
- 4 **[0035]** Figure 20 is a CHI matrix for the environment.
- 5 **[0036]** Figure 21 is a flowchart illustrating a system compatibility analysis procedure.
- 6 **[0037]** Figure 22 is a flowchart illustrating a configuration data extraction procedure.
- 7 **[0038]** Figure 23 is a flowchart illustrating a configuration compatibility analysis procedure.
- 8 **[0039]** Figure 24 is a flowchart illustrating a rule set application procedure.
- 9 **[0040]** Figure 25 is a flowchart illustrating a workload data extraction procedure.
- 10 **[0041]** Figure 26 is a flowchart illustrating a workload compatibility analysis procedure.
- 11 **[0042]** Figure 27 is a flowchart illustrating a workload multi-stacking procedure.
- 12 **[0043]** Figure 28 is a flowchart illustrating another workload multi-stacking procedure using
13 a hypothetical target system.
- 14 **[0044]** Figure 29 is a is a hierarchical block diagram illustrating metadata, rules and rule
15 sets.
- 16 **[0045]** Figure 30 is a schematic flow diagram showing the application of a rule set in
17 analyzing a pair of computer systems.
- 18 **[0046]** Figure 31 illustrates a general rule definition.
- 19 **[0047]** Figure 32 shows an example rule set.

DETAILED DESCRIPTION OF THE INVENTION

[0048] Referring therefore to Figure 1a, an analysis program 10 is in communication with a set of computer systems 28 (3 are shown in Figure 1a as an example). The analysis program 10, using a computer station 16, evaluates the computer systems 28 and provides a report showing how the systems differ. The computer systems 28 may be physical systems as well as virtual systems or models. A distinct data set is preferably obtained for each system 28.

[0049] Each data set comprises one or more parameter that relates to characteristics or features of the respective system 28. The parameters can be evaluated by scrutinizing program definitions, properties, objects, instances and any other representation or manifestation of a component, feature or characteristic of the system 28. In general, a parameter is anything related to the system 28 that can be evaluated, quantified, measured, compared etc.

Exemplary Environment

[0050] Referring to Figures 1b and 2, a compatibility analysis program, generally referred to by numeral 10 for clarity is deployed to gather data from the exemplary architecture shown for a computing environment 12 (shown in Figure 1b). The analysis program 10 analyzes the environment 12 to determine whether or not compatibilities exist within the environment 12 for consolidating systems such as servers, desktop computers, routers, storage devices etc. The analysis program 10 is preferably part of a client-server application that is accessible via a web browser client 34 running on, e.g. a computer station 16. The analysis program 10 operates in the environment 12 to collect, analyze and report on audited data for not only consolidation but other functions such as inventory analysis, change and compliance analysis etc. In the following examples, the systems are exemplified as servers.

[0051] As shown in Figure 1b, the example environment 12 generally comprises a master server 14 that controls the operations of a series of slave servers 28 arranged in a distributed system. In the example shown, the master server 14 audits a local network 18 having a series of servers 28 some having local agents and others being agentless. The master server also audits a pair of remote networks 20, 22 having firewalls 24. The remote network 20 includes a proxy for

1 avoiding the need to open a port range. The remote network 22 comprises a collector 30 for
2 concentrating traffic through a single point allowing an audit to be performed through the
3 firewall 24, and also comprises a proxy 32. The proxy 32 is used to convert between
4 Windows™ protocols and UNIX™/Linux™ servers, and can also concentrate traffic. The
5 proxy 32 may be required for auditing agentless Windows™ based server if the master server 14
6 is running another operating system such as UNIX™ or Linux™.

7 **[0052]** The master server 14 is capable of connecting to the slave servers 28 for performing
8 audits of configuration settings, workload etc. and thus can communicate over several applicable
9 protocols, e.g. simple network management protocol (SNMP). As shown, a computer station 16
10 running a web browser and connected to a web server (not shown) on the master server 14, e.g.
11 over HTTP, can be used to operate the analysis program 10 in the environment 12. The analysis
12 program 10 may reside on the master server 14 or may run on a remote server (not shown). The
13 analysis program 10 can gather data as it is available or retrieve a block of data from the master
14 server 14 either via electronic means or other physical means. As such, the analysis program 10
15 can operate in the environment 12 or independently (and remote thereto) so long as it can obtain
16 audited data from the environment 12. The computer station 16 enables the analysis program 10
17 to display reports and gather user input for executing an audit or analysis.

18 **Analysis Program**

19 **[0053]** A example block diagram of the analysis program 10 is shown in Figure 2. The flow
20 of data through the program 10 begins as an audit engine 46 pulls audit data from audited
21 environments 50. The data works its way up to the web client 34 which displays an output on a
22 user interface, e.g. on computer system 16. The program 10 is preferably a client-server
23 application that is accessed via the web client or interface.

24 **[0054]** An audit engine 46 communicates over one or more connections referred to generally
25 by numeral 48 with audited environments 50 which are the actual systems 28, e.g. server
26 machines, that are being analysed. The audit engine 46 typically uses data acquisition (DAQ)
27 adapters to communicate with the end points (e.g. servers 28) or software systems that manage
28 the end points (e.g. management frameworks 52 and/or agent instrumentation 54). The program

21562813.1

1 10 can utilize management framework adapters 52 in the audited environments 50 for
2 communicating with ESM frameworks and agent instrumentation and for communicating with
3 other agents such as a third party or agents belonging to the program 10. The audit engine 46
4 can also communicate directly with candidate and/or target systems 28 using agentless adapters
5 (central arrow in Figure 2) to gather the necessary audit information.

6 **[0055]** An auditing data repository 42 is used to store audit information and previous reports.
7 The audit engine 46, using a set of audit templates 45, controls the acquisition of data that is used
8 by the other software modules to eventually generate a set of reports to display on the interface
9 34. A context engine 40 utilizes metadata 39 stored by the program 10, which indicates the
10 nature of the data, to filter out extraneous information.

11 **[0056]** An analysis engine 41 evaluates data compared in a differential engine 38 based on a
12 set of rules 43. The analysis engine 41 performs the compatibility and, in this example, the
13 consolidation analysis to determine if the environment 12 can operate with fewer systems 28.

14 **[0057]** The program 10 has a report generation tool 36 that utilizes a set of report templates
15 35 for generating custom reports for a particular environment 12. The report generation tool 36
16 utilizes the information generated by the analysis engine 41. Typically, the program 10 includes
17 a web client 34 for communicating with a web interface (e.g. on computer system 16). The web
18 interface allows a user to enter settings, initiate an audit or analysis, display reports etc.

19 **System Compatibility Analysis Visualization**

20 **[0058]** In the following examples, a source system refers to a system from which
21 applications and/or data are to be moved, and a target server or system is a system to which such
22 applications and/or data are to be moved. For example, an underutilized environment having two
23 systems 28 can be consolidated to a target system (one of the systems) by moving applications
24 and/or data from the source system (the other of the systems) to the target system.

1 [0059] As best seen in Figure 3, the systems 28 are audited in stage A to generate
2 configuration reports and workload patterns, which are in turn used to create statistical
3 scorecards in stages B, C and D.

4 [0060] The first stage, stage A, involves data collection, which includes the collection of
5 detailed configuration and workload information from the environment 12. Stage A also
6 includes data extraction to obtain the relevant configuration and workload data from the per-
7 system data gathered in an audit according to compatibility rule sets and workload data types to
8 obtain filtered per-system configuration and workload data sets. In addition, per-system
9 benchmark data and usage limits are considered in Stage A for performing the workload analysis
10 (Stage C).

11 [0061] Stages B and C are then performed using the data that has been collected. Stage B
12 involves performing the system compatibility analysis to generate a system compatibility index
13 (SCI) matrix 60 and Stage C involves performing the workload compatibility analysis to
14 generate a workload compatibility index (WCI) matrix 70. The results from stages B and C are
15 then used in stage D to perform an overall compatibility analysis which involves the generation
16 of a co-habitation index (CHI) matrix 80 and its visual mapping of overall system compatibility.
17 The analysis results may then be used to identify the best server consolidation candidates. It will
18 be appreciated that the principles described herein support many strategies and consolidation is
19 only one example.

20 Objectives

21 [0062] From an analysis perspective, as shown in Figure 4, a cost-savings objective can be
22 evaluated on the basis of certain strategies such as, but not limited to, database consolidation
23 strategies, application stacking strategies, operating system (OS) level stacking strategies and
24 virtualization strategies that can also be visualized graphically using a comprehensive
25 consolidation roadmap 90 incorporating the SCI 60, WCI 70 and CHI 80 matrices as will be
26 explained in greater detail below.

1 **System Configuration Compatibility Visualization**

2 **[0063]** A configuration analysis in Stage B of N systems 18 computes NxN system
3 compatibility scores by individually considering each system 18 as a consolidation source and as
4 a target. Preferably, the scores range from 0 to 100 with higher scores indicating greater system
5 compatibility. The analysis will thus also consider the trivial cases where systems are
6 consolidated with themselves and would be given a maximum score, e.g. 100. For display and
7 reporting purposes, the scores are preferably arranged in an NxN matrix form.

8 **[0064]** An example of an SCI matrix 60 is shown in Figure 6. The SCI matrix 60 provides
9 an organized graphical mapping of system compatibility for each source/target system pair on
10 the basis of configuration data. The SCI matrix 60 shown in Figure 6 is structured having each
11 server 28 in the environment 12 listed both down the leftmost column 64 and along the
12 uppermost row 62. Each row represents a consolidation source system, and each column
13 represents the possible consolidation target. Each cell contains the score corresponding to the
14 case where the row system is consolidated onto the column (target) system.

15 **[0065]** The preferred output shown in Figure 6 arranges the servers 28 in the matrix such that
16 a 100% compatibility exists along the diagonal 63 where each server is naturally 100%
17 compatible with itself. The SCI matrix 60 is preferably displayed such that each cell 66 includes
18 a numerical score and a shade of a certain colour. As noted above, the higher the score (from
19 zero (0) to one hundred (100)), the higher the compatibility. The scores are pre-classified into
20 predefined ranges that indicate the level of compatibility between two systems 18. Each range
21 maps to a corresponding colour or shade for display in the matrix 60. For example, the
22 following ranges and colour codes can be used: score = 100, 100% compatible, dark green; score
23 = 75-99, highly compatible, green; score = 50-74, somewhat compatible, yellow; score = 25-49,
24 low compatibility, orange; and score = 0-24, incompatible, red.

25 **[0066]** The above ranges are only one example. Preferably, the ranges can be adjusted to
26 reflect more conservative and less conservative views on the compatibility results. The ranges
27 can be adjusted using a graphical tool similar to a contrast slider used in graphics programs.

Adjustment of the slider would correspondingly adjust the ranges and in turn the colours. This allows the results to be tailored to a specific situation.

[0067] It is therefore seen that the graphical output of the SCI matrix 60 provides an intuitive mapping between the source/target pairs in the environment 12 to assist in visualizing where compatibilities exist and do not exist. In Figure 6, it can be seen that the server pair identified with an asterisk (*) and by the encircled cell indicates complete compatibility between the two servers for the particular strategy being observed, e.g. based on a chosen rule set. It can also be seen that the server pair identified with an X and the encircled cell at the corresponding row/column crossing comprises a particularly poor score and thus for the strategy being observed, the servers 28 in that pair are not very compatible.

[0068] The scores are calculated based on configuration data that is acquired through a configuration audit performed by the analysis program 10. The data is acquired using tools such as the table 100 shown in Figure 5 that illustrate the various types of configuration settings that are of interest and from which sources they can be obtained. Figure 5 also provided a mapping to where the sample workload data can be obtained. In Figure 5, a number of strategies 104 and sub-strategies 105 map to various configuration and workload sources, collectively referred to by numeral 102. As discussed making reference to Figure 4, the strategies 104 may relate to database consolidation, OS-level stacking, application server stacking and virtualization. Each strategy 104 includes a set of sub-strategies 105, which in turn map to specific rule sets 43. The rule sets, which will be explained in greater detail below, determine whether or not a particular setting or system criterion/criteria have been met and thus how different one server 28 is to the next.

[0069] The table 100 lists the supported consolidation strategies and the relevant data sources that should be audited to perform the corresponding consolidation analysis. In general, collecting more basis data improves the analysis results. The table 100 enables the analysis program 10 to locate the settings and information of interest based on the strategy 104 or sub-strategy 105 (and in turn the rule set) that is to be used to evaluate the systems 28 in the

environment 12. The results can be used to determine source/target candidates for analysing the environment for the purpose of, e.g. consolidation, compliance measures etc.

System Workload Compatibility Visualization

[0070] An example WCI matrix 70 is shown in Figure 7. The WCI matrix 70 is the analog of the SCI matrix 60 for workload analysis. The WCI matrix 70 includes a similar graphical display that indicates a score and a colour or shading for each cell to provide an intuitive mapping between candidate source/target server pairs. The workload data is obtained using tools such as the table 100 shown in Figure 5 and corresponds to a particular workload factor, e.g. CPU utilization, network I/O, disk I/O, etc. A high workload score indicates that the candidate server pair being considered has a high compatibility for accommodating the workload on the target system. The specific algorithms used in determining the score are discussed in greater detail below. The servers are listed in the upper row 72 and leftmost column 74 and each cell 76 represents the compatibility of its corresponding server pair in the matrix. The encircled cell identified by the asterisk (*) in Figure 7 indicates a high workload compatibility for consolidating to the target server, and the one marked by the X indicates an unlikely candidate pair for workload consolidation, compliance etc.

System Co-Habitation Compatibility Visualization

[0071] An example CHI matrix 80 is shown in Figure 8. The CHI matrix 80 comprises a similar arrangement as the SCI and WCI matrices 60, 70, which lists the servers in the uppermost row 82 and leftmost column 84 to provide 100% compatibility along the diagonal. Preferably the same scoring and shading convention is used as shown in Figure 8. The CHI matrix 80 provides a visual display of scoring for candidate system pairs that considers both the configuration compatibility from the SCI matrix 60 and the workload compatibility from the WCI matrix 70.

[0072] The score provided in each cell 86 indicates the co-habitation compatibility for consolidating servers. It should be noted that in some cases two servers 28 can have a high configuration compatibility but a low workload compatibility and thus end up with a reduced or

1 relatively low co-habitation score. It is therefore seen that the CHI 80 provides a comprehensive
2 score that considers not only the compatibility of systems 28 at the setting level but also in its
3 utilization. By displaying the SCI matrix 60, WCI matrix 70 and CHI matrix 80 in the
4 consolidation roadmap 90, a complete picture of the entire system can be ascertained in an
5 organized manner. The matrices 60, 70 and 80 provide a visual representation of the
6 compatibilities and provide an intuitive way to evaluate the likelihood that systems can be
7 consolidated and have associated tools (as explained below) that can be used to analyse
8 compliance and remediation measures to modify systems 28 so that they can become more
9 compatible with other systems 28 in the environment 12. It can therefore be seen that a
10 significant amount of quantitative data can be analysed in a convenient manner using the
11 graphical matrices 60, 70, 80 and associated reports and graphs (described below).

12 **[0073]** For example, a server pair that is not compatible only for the reason that certain
13 critical software upgrades have not been implemented, the information can be uncovered through
14 analysis tools used with the SCI matrix 60, and then investigated, so that upgrades can be
15 implemented, referred to herein as remediation. Remediation can be determined by modeling
16 cost of implementing upgrades, fixes etc that are needed in the rule sets. If remediation is then
17 implemented, a subsequent analysis may then show the same server pair to be highly compatible
18 and thus suitable candidates for consolidation.

19 **Sorting Examples**

20 **[0074]** The matrices 60, 70 and 80 can be sorted in various ways to convey different
21 information. For example, sorting algorithms such as a simple row sort, a simple column sort
22 and a sorting by group can be used.

23 **[0075]** A simple row sort involves computing the total scores for each source system (by
24 row), and subsequently sorting the rows by ascending total scores. In this arrangement, the
25 highest total scores are indicative of source systems that are the best candidates to consolidate
26 onto other systems.

1 [0076] A simple column sort involves computing the total scores for each target system (by
2 column) and subsequently sorting the columns by ascending total score. In this arrangement, the
3 highest total scores are indicative of the best consolidation target systems.

4 [0077] Sorting by group involves computing the difference between each system pair, and
5 arranging the systems to minimize the total difference between each pair of adjacent systems in
6 the matrix. The difference between a system pair can be computed by taking the square root of
7 the sum of the squares of the difference of a pair's individual compatibility score against each
8 other system in the analysis. In general, the smaller the total difference between two systems,
9 the more similar the two systems with respect to their compatibility with the other systems. The
10 group sort promotes the visualization of the logical breakdown of an environment by producing
11 clusters of compatible systems 18 around the matrix diagonal. These clusters are indicative of
12 compatible regions in the environment 12.

13 **Consolidation Roadmap**

14 [0078] The consolidation roadmap 90 shown in Figure 9 illustrates how the matrices 60, 70
15 and 80 can be used to provide a complete visualization. As shown in Figure 9, configuration
16 deviation examples 92 can be generated based on the SCI matrix 60 and displayed in a chart or
17 table to show where compatibilities are lacking. Similarly, workload stacking examples 94 can
18 be generated based on the WCI 70 to show how a candidate server pair would operate when the
19 respective workloads are stacked. Ultimately, a current server utilization versus capacity
20 comparison 96 can also be illustrated to show combined server capacities and other information
21 pulled from the CHI 80.

22 **System Compatibility Analysis Overview**

23 [0079] The following illustrates an example for generating the matrices 60, 70 and 80
24 discussed above. The analysis program 10 generally executes four primary stages as shown in
25 Figures 3 and 21.

1 **System Configuration Compatibility**

2 **[0080]** In stage A, a configuration data extraction step is performed, which is shown in
3 greater detail in Figure 22. The per-system configuration data comprises a data set for each
4 system 28 that is obtained during the auditing process. The compatibility rule set defines which
5 settings are important for determining compatibility. The compatibility rule set is typically a
6 predefined set of rules which can be revised as necessary based on the specific environment 12.
7 The rule set is thus preferably compiled according to the target systems being analysed and prior
8 knowledge of what makes a system compatible with another system for a particular purpose.

9 *Differential Rule Sets*

10 **[0081]** Further detail regarding the rules and rule sets is now described making reference to
11 Figures 29-32, as described in co-pending U.S. Patent Application No. * filed on September 26,
12 2006, and entitled "Method for Evaluating Computer Systems", the contents of which are
13 incorporated herein by reference.

14 **[0082]** With respect to the following description of the differential rule sets 74 and the
15 general application of the rule sets 74 for evaluating differences between systems, the following
16 alternative nomenclature may be used. A target system refers to a system being evaluated, and a
17 baseline system is a system to which the target system is being compared. The baseline and
18 target systems may be the same system at different instances in time (baseline = prior, target =
19 now) or may be different systems being compared to each other. As such, a single system can be
20 evaluated against itself to indicate changes with respect to a datum as well as how it compares to
21 its peers. It will be appreciated that the terms "source system" and "baseline system" are herein
22 generally synonymous, whereby a source system is a type of baseline system.

23 **[0083]** Figure 29 illustrates a visual representation of the relationships between data used by
24 the analysis program 10. Audited data 70 is obtained from the source and target computer
25 systems 28 and is used to evaluate the differences between the systems 28. In this example, the
26 differences are evaluated to determine configuration compatibilities for a consolidation strategy.

1 A distinct data set is preferably obtained for each system 28 (or instance in time for the same
2 system 28 as required).

3 **[0084]** Metadata 39 describes the meaning of the audited data 70 as it pertains to the
4 analysis. Preferably, comprehensive metadata 39 is included in the analysis program 10 and
5 should be capable of being modified based on the specific application and the nature of the
6 computer systems 28.

7 **[0085]** Differential rules 43 are conceptually a form of metadata 39 that represent the
8 importance of differences in certain parameters for the baseline and target systems 28, the
9 dependencies between different aspects of the audited data 70, and the costs associated with the
10 remediation of differences between the system parameters.

11 **[0086]** Differential rule sets 76 are groupings of rules that represent higher-level
12 considerations such as business objectives or administrative concerns that are taken into account
13 when reporting on or analysing the systems 28. In this example, four differential rules 43, A, B,
14 C and D, are grouped into two differential rule sets 76, Rule Set 1 and Rule Set 2. It will be
15 appreciated that there may be any number of rules 43 in any number of differential rule sets 76
16 and those shown in Figure 2 are for illustrative purposes only.

17 **[0087]** The differential rules 43 evaluate the differences in parameters in the audited data 70
18 according to rule definitions. The rule definitions include weights that are indicative of the
19 importance of the differences in particular parameters as they relates to the operation of the
20 systems 28. The weights are applied during an evaluation of the baseline and target systems 28
21 if the difference exists. The evaluation may include the computation of a score or generation of
22 other information indicative of nature of the difference(s) between the baseline and target
23 systems 28.

24 **[0088]** The flow of data for applying an exemplary rule set 76 is shown in Figure 30. In the
25 example shown in Figure 30, the audit engine 46 gathers audited data at step 300 from a pair of
26 laptop computer systems 28. At step 302, the context engine 40 filters the audited data 70 using
27 metadata 39 to determine which parameters of the laptop computer systems 28 are applicable to

1 the analysis. The differential engine 38 and the analysis engine 41 look at the differences in
2 parameters for the systems 28 and apply a differential rule set at step 304 which in turn evaluates
3 the differential rule definitions for exemplary rules A, B, C and D.

4 **[0089]** As noted above, the rules 43 evaluate the differences in the baseline and target
5 systems 28 and apply weights that indicate the importance of such differences in the parameters
6 that have been analysed as well as the dependencies between different aspects of the data. The
7 rule sets 76, e.g. Rule Set 1 and Rule Set 2, determine which parameters in the audited data 70
8 are to be evaluated and the differential rules 43 in the differential rule sets 76 are applied to the
9 differences between the parameters in the baseline and target systems 28 based on the presence
10 of a difference. The difference may simply be whether or not the parameter is different but
11 nature of the difference may also be considered and have weights that vary based on how
12 different the parameter is. As such, the differential rules 43 and corresponding weights may vary
13 accordingly. For example a version 4 operating system versus a version 3 operating system may
14 be considered less costly to remedy and thus less detrimental than a version 5 operating system
15 compared to a version 1 operating system. As can be seen, even though the operating systems
16 are different in both cases, the nature of the difference can also be considered and different
17 weights and/or remedies applied accordingly.

18 **[0090]** The report generator 36 uses the results of the application of the differential rules 43
19 to generate a report at step 306, which is then in turn displayed on the computing station 16 for
20 subsequent analysis, use and/or storage.

21 **[0091]** A general definition for a differential rule 43 is shown in Figure 31. Each rule
22 definition comprises a number rule fields and the corresponding values. A rule definition can be
23 extended to include any number of rules 43 to form a rule set 76 as shown by way of example
24 only in Figure 32. The rule definitions are computer readable and storable so that they may be
25 accessed by the program 10 and modified if necessary, for use in evaluating the computer
26 systems 28.

27 **[0092]** The rule type specifies whether the rule 43 applies to audited data directly
28 (UrlQuery) or normalized values (AliasQuery). The rule specifier specifies the URL of the data

21562813.1

1 object or property that is being evaluated. The optional URL fragment (i.e. the portion after the
2 “#” symbol) specifies the specific object instance (table row) that is being evaluated, with “*”
3 denoting a wildcard that matches all instances. For AliasQuery rules, this field specifies the alias
4 name.

5 **[0093]** If specified, the source field represents the literal value that would need to match the
6 value of the object/property on the source system in order for the rule 43 to match. For objects
7 and object instances, the keywords “absent” and “present” are preferably used to match cases
8 where that object is absent or present respectively. Similar to the source field, the target field
9 allows a literal match against the value of the object/property on the target system. The target
10 field also supports the absent/present specifiers. For numeric properties, relational operators (>
11 <, =, !=) can be used to cause the rule 43 to trigger if the target value has the specified
12 relationship with the source value.

13 **[0094]** The weight field specifies the relative importance of that property and combination
14 of source/target values (if specified) in regard to the overall context of the comparison. Higher
15 values indicate that the condition detected by the rule 43 has a high impact on the target
16 environment.

17 **[0095]** The mutex flag field can be used to avoid multiple penalties that would otherwise
18 skew the scores. A “Y” in the mutex flag field specifies that multiple matches of the same rule
19 43 will incur only a single penalty on the overall score (as specified in the weight field), as
20 opposed to multiple accumulating penalties (which is the default behaviour).

21 **[0096]** The match flag field enables an optional symbolic flag to be “set” when a rule 43
22 matches, and which can subsequently be used to suppress other rules 43 (through the “Suppress
23 Flags” field). This effectively allows rule dependencies to be modeled in the rule set 76. The
24 suppress flag field allows symbolic flags (as specified in the “Match Flag” field) to be used to
25 suppress the processing of rules. This allows specific checks to be skipped if certain higher-level
26 conditions exist. For example, if the operating systems are different, there is no need to check
27 the patches.

1 [0097] The remediation cost field is preferably optional. The remediation field represents
2 the cost of “fixing” the system(s) (i.e. eliminating the condition or discrepancy detected by the
3 rule 43). When analyzing differences between (or changes to) IT systems this is used to
4 represent hardware/software upgrade costs, administrative costs and other costs associated with
5 making the required changes to the target systems. The calculations behind this field vary based
6 on the nature of the system and the parameter that would need to be added, upgraded etc.

7 [0098] The description field is a lay description of the condition or discrepancy detected by
8 the rule 43. These descriptions are used to provide management-level summaries when
9 processing rule sets 76. The description field can provide as much or as little information as
10 required by the application.

11 [0099] Figure 32 provides an example rule set 76, which includes a number of rules 43.
12 The following refers to the number indicated in the leftmost column of Figure 32.

13 [00100] Rule 1 scrutinizes the normalized (AliasQuery) representation of the operating
14 systems (e.g. Windows™, Solaris™, AIX™, Linux™, etc.) on both the source and target
15 systems and heavily penalizes cases where these are different as evident from the high weight
16 factor (70%). Rule 2 penalizes systems that have different operating system versions (e.g.
17 Windows™ NT vs Windows™ 2000), and is suppressed (i.e. not processed) in cases where the
18 systems have different overall operating systems (as detected in the previous rule 43). Rule 3
19 detects if systems are in different time zones. Rule 4 penalizes combinations of systems where
20 the target has less memory than the source (this is what is referred to as a directional rule 43,
21 which can give differing results if sources and targets are reversed, e.g. asymmetric results).
22 Rule 5 operates directly against audit data and detects cases where the operating system patch
23 level differs. This rule is not processed if either the operating system or the operating system
24 version are different (since this renders the comparison of patches meaningless).

25 [00101] Rule 6 scrutinizes the lists of all patches applied to the source and target systems and
26 penalizes cases where they differ. The mutex flag is set, indicating that the penalty is applied
27 only once, no matter how many patch differences exist. This rule is ignored in cases where
28 either the operating system or operating system version are different. Rule 7 penalizes system

21562813.1

1 combinations of servers that are running the same OS but are configured to run a different
2 number of kernel bits (e.g. 64-bit vs 32-bit). Rule 8 penalizes combinations where there are
3 kernel parameters defined on the source that are not defined on the target. This rule is not
4 applied if the operating systems are different.

5 **[00102]** Rule 9 scrutinizes a specific kernel setting (SHMMAX, the setting that specifies how
6 much shared memory a system can have) and penalizes combinations where it is set to a lower
7 value on the target than it is on the source system. Rule 10 penalizes combinations of systems
8 that are running different versions of Oracle™. It should be noted that the remediation cost is
9 relatively high, owing to the fact that it will take a software upgrade to eliminate this
10 discrepancy. Rule 11 penalizes combinations of systems that are running different database
11 version, e.g. Oracle™ 9 vs. Oracle™ 8. In some cases the remediation cost can be low where the
12 upgrade is less expensive. Rule 12 penalizes combinations of systems that are running different
13 versions of Apache. It should be noted that the remediation cost is relatively low, as apache is an
14 open source product and the cost of upgrade is based on the hourly cost of a system administrator
15 and how long it will take to perform the upgrade.

16 **[00103]** Rule 13 scrutinizes a windows-specific area of the audit data to determine if the
17 source and target systems are running different service pack levels. It should be noted that this
18 rule closely mirrors rule 5, which uses a rule specifier that scrutinizes the UNIX™/Linux™ area
19 of the audit data. Rule 14 scrutinizes the lists of all hotfixes applied to the source and target
20 systems and penalizes cases where they differ. This rule closely mirrors rule 6, which scrutinizes
21 patches on UNIX™ and Linux™. Rule 15 detects differing startup commands between systems.
22 Rule 16 is a rule 43 to detect differing Paths between systems, and rule 17 detects differing
23 System Paths between systems.

24 **[00104]** Rule 18 penalizes system combinations where there are services installed on the
25 source that are not installed on the target. This rule has the mutex flag set, and will therefore
26 only penalize a system combination once, no matter how many services are missing. Rule 19
27 penalizes system combinations where there are services started on the source that are not started
28 on the target. It should be noted that both the weight and the remediation cost are lower than the

1 previous rule 43, owing to the fact that it is generally easier and less expensive to start a service
2 than install it. Finally, rule 20 penalizes combinations where the target system is missing the
3 virus scanner software.

4 **[00105]** It will be appreciated that the above described rules 43 and rule set 76 are shown for
5 illustrative purposes only and that any combination of rules 43 can be used to achieve specific
6 goals. For example, rules that are applicable to the OS can be grouped together to evaluate how
7 a system 28 compares to its peers. Similarly, rules pertaining to database, Java applications etc.
8 can also be grouped.

9 *System Configuration Compatibility continued...*

10 **[00106]** Turning back to Figure 22, configuration data extraction analyses the per-system
11 configuration data and the compatibility rule sets. Each system is analysed for each rule in the
12 rule set. For each rule, the configuration data referenced by the rule (according to its definition)
13 is extracted and saved by the analysis engine 41. The extraction process results in a filtered data
14 set for each system 28 that corresponds to the actual configuration data that can be used to
15 determine compatibilities between the systems 28. The filtered data is used in Stage B for the
16 compatibility analysis.

17 **[00107]** An exemplary flow chart for a configuration compatibility analysis is shown in
18 greater detail in Figures 23-24. When analyzing system compatibility, the list of target and
19 source systems 28 are the same. The compatibility is evaluated in two directions, e.g. for a
20 Server A and a Server B, migrating A to B is considered as well as migrating B to A.

21 **[00108]** Turning first to Figure 23, for each target system T ($T = 1$ to N where N is the number
22 of systems), the differential engine 38 first looks at each source system S ($S = 1$ to N). If the
23 source=target then the SCI score for that source is set to 100, no further analysis is required and
24 the next pair is analyzed. If the source and target are different, the rules are evaluated against the
25 source/target pair to compute the SCI score, remediation cost and to compile the associated rule
26 details. Estimated remediation costs are optionally specified with each rule item. As part of the
27 rule evaluation and subsequent SCI score calculation, if a rule is true (a difference is identified),

21562813.1

1 the corresponding cost to address the deficiency is added to the remediation cost for the pair of
2 systems 28 being analysed.

3 **[00109]** The evaluation of the rules is shown in Figure 24. The evaluation of the rules
4 considers the filtered configuration data for both the source system and the target system, as well
5 as the compatibility rule set that is being applied. For each rule in the set, the data referenced by
6 the rule is obtained from both the target data and source data. The rule is evaluated by having
7 the differential engine 38 compare the data. If the rule is not true (i.e. if the systems are the same
8 according to the rule definition) then the data is not considered in the SCI score and the next rule
9 is evaluated. If the rule is true, the rule details are added to an intermediate result. The
10 intermediate result includes all true rules.

11 **[00110]** Preferably, a suppression tag is included with each rule. As discussed above, the
12 suppression tag indicates other rules that are not relevant if that rule is true. The suppression flag
13 allows the program 10 to avoid unnecessary computations. A mutex flag is also preferably used
14 to avoid unfairly reducing the score for each true rule when the rules are closely affected by each
15 other.

16 **[00111]** Once each rule has been evaluated, a list of applicable rules is created by removing
17 inapplicable rule entries from the intermediate results based on rule dependencies, which are
18 defined by rule matching and suppression settings (e.g. mutex flags and suppression tags). The
19 applicable rules are then processed to calculate the SCI score for that particular source/target pair
20 based on the rule weights. Remediation costs are also calculated based on the cost of
21 updating/upgrading etc. and the mutex settings.

22 **[00112]** Turning back to Figure 23, the current target is then evaluated against all remaining
23 sources and then the next target is evaluated. As a result, a N x N matrix can be created that
24 shows a compatibility score for each system against each other system. The matrix can be sorted
25 by grouping the most compatible systems. The sorted SCI matrix 60 is comprised of every
26 source/target combination and thus provides an organized view of the compatibilities of the
27 systems 28.

1 [00113] Preferably, an SCI report is then generated comprising the SCI matrix 60 (e.g. Figure
2 6) and for each source-target pair details available pertaining to the SCI scoring weights,
3 remediation costs and applicable rules. The details can preferably be pulled for each
4 source/target pair by selecting the appropriate cell.

5 **System Workload Compatibility**

6 [00114] Turning back to Figure 21, in stage A, a workload data extraction step is shown. The
7 workload data extraction is shown in greater detail in Figure 25. The data extraction considers
8 the per-system workload histogram data obtained during the auditing process and workload data
9 category, e.g. CPU utilization). Preferably, the histogram data includes data obtained during one
10 or more days. For each system in the workload data set, each day of the workload data for that
11 system is analyzed. The analysis engine 41 evaluates the workload data at each hour according
12 to the workload data category. Preferably, the analysis engine 41 also converts the data if
13 required to accommodate inverted percentages. For example, if the data indicates that the CPU
14 is idle for a certain percentage of the day, it is converted to the percentage of the day in which it
15 is busy. This is done since in this example, the percentage that the system is busy is more
16 relevant to how the CPU is being utilized rather than how much it is not being utilized.

17 [00115] In the current day being evaluated, the analysis engine 41 determines if that sample of
18 data is the most representative data, e.g. if it is the worst day. If the sample is the most
19 representative, then it is saved as the current workload data sample for the system. If not and/or
20 once the sample is saved, the analysis engine 41 then determines if there are any more days to
21 evaluate. If not, the current representative day is deemed to be the most representative for that
22 system 28 and the remaining systems 28 are evaluated if necessary. In the result, a filtered per-
23 system workload data set is produced for each system, i.e. the representative sample for each
24 system.

25 [00116] The filtered workload data is then used in Stage C for conducting the workload
26 compatibility analysis. When analyzing workload compatibility, the list of target and source
27 systems 28 are the same. The compatibility is evaluated in two directions, e.g. for a Server A
28 and a Server B, migrating A to B is considered as well as migrating B to A.

21562813.1

1 [00117] An exemplary workload compatibility analysis procedure is shown in Figure 26 that
2 uses the workload data obtained in stage A as discussed above, and also considers the workload
3 limit usage, e.g. 75% of the CPU being busy, and the per-system benchmarks, e.g. relative CPU
4 power. Each system 28 in the workload data set is considered as a target ($T = 1$ to N) and
5 compared to each other system 28 in the data set as the source ($S = 1$ to N). The analysis engine
6 first determines if the source and target are the same. If yes, then the WCI score is set to 100
7 and no additional analysis is required for that pair. If the source and target are different, the
8 system benchmarks are then used to normalize the workloads (if required). The normalized
9 source workload histogram is then stacked on the normalized target system.

10 [00118] System benchmarks can normalize workloads as follows. For systems A and B, with
11 CPU benchmarks of 200 and 400 respectively (i.e. B is 2x more powerful than A), if systems A
12 and B have CPU utilization of 10% and 15% respectively, the workloads can be normalized
13 through the benchmarks as follows. To normalize A's workload to B, multiply A's workload by
14 the benchmark ratio A/B , i.e. $10\% \times 200/400 = 5\%$. Stacking A onto B would then yield a total
15 workload of $5\% + 15\% = 20\%$. Conversely, stacking B onto A would yield the following total
16 workload: $10\% + 15\% \times 400/200 = 40\%$.

17 [00119] Using the stacked histogram, the maximum peak (4th quartile) and sustained (3rd
18 quartile) workloads at like times are determined. The worst case maximum peak (4th quartile)
19 and sustained (3rd quartile) stacked workloads are then determined for the stacked workloads of
20 the source/target pair. Typically, in contrast to the maximum workloads at like times, the worst
21 case maximum workloads may be based on different times. The WCI score is then computed by
22 applying the usage limit to the following stacked workload values: maximum sustained (3rd
23 quartile) workload at like times, maximum peak (4th quartile) workload at like times, maximum
24 sustained (3rd quartile) workload (worst case), and maximum peak (4th quartile) workload (worst
25 case). The aggregate WCI score is then computed for the source/target pair based on the
26 component WCI scores. To calculate the aggregate score, a weighting factor is applied to each
27 component based on its relative importance to workload compatibility.

1 [00120] Each source is evaluated against the target, and each target is evaluated to produce an
2 N x N matrix of scores, which can be sorted to group compatible systems. Preferably, a WCI
3 report is generated that includes the WCI matrix 70 and WCI scoring details and normalized
4 stacked workload histograms that can be viewed by selecting the appropriate cell. The workload
5 compatibility results are then used for the CHI analysis.

6 **System Co-Habitation Compatibility**

7 [00121] Stage D uses the results of both the configuration and workload analyses to perform
8 an overall co-habitation analysis. The co-habitation analysis is performed for each server pair
9 found in the WCI 70. The corresponding configuration score is first located from the SCI 60 and
10 a co-habitation score for the server pair is computed by applying a formula specified by an
11 overlay algorithm which performs a mathematical operation such as multiply or average, and the
12 score is recorded. If more servers are to be examined, the above process is repeated. When CHI
13 scores for all server pairs have been computed, the CHI matrix 80 is displayed graphically (see
14 Figure 8) and each cell 86 is linked to a scorecard that provides further information. The further
15 information can be viewed by selecting the cell. A sorting algorithm is then preferably executed
16 to configure the matrix as shown in Figure 8.

17 [00122] Once the above computations are complete, the complete roadmap 90 can be
18 generated by displaying not only the CHI matrix 80, but also the SCI matrix 60 and the WCI
19 matrix 70 and selected information from the associated scorecards as shown in Figure 9.

20 **System Workload Multi-Stacking**

21 [00123] A procedure for multi-stacking source systems on a target system is shown in Figure
22 27. The multi-stacking procedure considers the workload usage limit that is specified using the
23 program 150, the per-system benchmarks (e.g. CPU power), and the filtered workload data for
24 the source and target systems. The multi-stack may include any number of sources stacked on a
25 target. For each target workload data set, each source data set is evaluated. If the source and
26 target are not the same, the system benchmarks are used to normalize the workloads as discussed
27 above, and the source workload histogram is stacked on the target system. Each source in the set

1 is then stacked on the target system. From the multi-stacked workload histogram (i.e. multiple
2 sources on a target), the WCI component scores are computed similar to that discussed above,
3 and an aggregate score for the multi-stack is computed to determine if such a multi-stack of
4 systems can be supported by the target. The above is repeated for each target to determine which
5 (if any) multi-stack arrangements are feasible. A multi-stack report may then be generated,
6 which gives a WCI scorecard for the target systems and for each target, provides WCI scoring
7 details and normalized multi-stacked workload histograms.

8 **[00124]** Figure 28 shows a similar procedure as that shown in Figure 27 but for multi-stacking
9 multiple sources on a hypothetical target. As can be seen from Figure 28, a hypothetical target
10 system benchmark is considered as well as per-system benchmarks for the sources. It can also
11 be seen that the procedure shown in Figure 28 is used to show how multiple sources can be
12 stacked on a hypothetical system. The hypothetical system is a “clean” system that can be used
13 to simulate the case where a new target system is acquired to consolidate existing systems
14 thereon.

15 **System Consolidation Analysis Example**

16 **[00125]** An example server consolidation analysis is provided below for an arbitrary
17 environment 12 having four servers, namely server A, server B, server C and server D.

18 **[00126]** The audit engine 46 collects detailed configuration and workload information from
19 the candidate servers 28, which may include, e.g. UNIX, Linux, Windows, AS/400 etc. The
20 process of collecting the detailed information is herein referred to as an audit. The audit engine
21 46 collects the data from instrumented candidate systems 12 through various protocols such as
22 simple network management protocol (SNMP), Windows management instrumentation (WMI),
23 SSH etc. as shown in Figure 1. Depending on the protocol used and the data to be collected,
24 instrumentation 54 on the environment 12 may need to be augmented with additional software
25 such as an agent associated with the analysis program 10.

1 [00127] The web client 34 and web interface allows the user to define the servers 28 to be
2 audited, the actual data to be collected and the audit communication protocol. An example
3 screen shot of an audit request program 110 is shown in Figure 10.

4 [00128] The program 110 enables a user to name and describe the audit so that it can be later
5 identified by entering the requisite information into the entry boxes 112. The audit request
6 information is included in sub-window 114 and the target server information is included in sub-
7 window 122. The type of request, e.g. SNMP is selected using the drop-down box 116. Based
8 on this selection, a list of request parameters are listed in sub-window 118. As shown in the
9 example, the SNMP version and the particular port are listed as well the type of community
10 string. It will be appreciated that any number of request parameters may be listed. Below the
11 request parameters, sub-window 120 provides a visual list of request templates which defines the
12 data that is to be collected and from where it can be obtained. This list can preferably be edited,
13 e.g. by selecting the "Edit" button as shown.

14 [00129] Examples of the request templates that are not necessarily included in Figure 10 are
15 MIB-II, which includes detailed network configuration and statistical data; host resources, which
16 includes system hardware configuration, devices, installed software, installed patches etc.; a
17 configuration monitor, which obtains OS configuration details; a software auditor, which obtains
18 data pertaining to installed patches; a base system workload via a system activity reporting
19 (SAR), which obtains hourly and daily performance statistics of basis properties such as CPU,
20 memory, swap etc.; and extended system workload via SAR, which obtains hourly and daily
21 performance data for additional properties including file I/O, swap I/O, page faults etc. It will be
22 appreciated that any number and form of request templates can be used based on the particular
23 strategies being used for a particular environment 12.

24 [00130] The portion 122 of the audit request program 110 displays a list of the audit targets,
25 which are each server 28 that is to be audited. In the example shown, servers A-D are listed and
26 this list can be edited should the user wish to add or remove certain servers 28. Once the user
27 has selected the desired settings and targets etc., they may choose an "Audit" button to begin the
28 audit, or they may cancel the audit and/or save the audit before it runs.

1 **[00131]** The audit results are acquired by requesting information specified in the selected
2 audit request templates from the servers 28 in the audited environments 50, and the audited data
3 is stored. The data is then accessed by the analysis program 10 and processed to generate reports
4 and display such reports.

5 **[00132]** A detailed configuration report 130 is shown in Figure 11. The detailed configuration
6 report organizes the audit data by system and category in the table 132. As shown in Figure 11,
7 both configuration and workload audit data can be viewed by selecting "View". The report 130
8 also displays audit information such as the name for the targeted audit, the type of audit and a
9 timestamp indicating when the audit was performed.

10 **[00133]** Portions of the detailed information that has been collapsed within the configuration
11 report can be viewed by selecting the appropriate "View" link in the report 130. An example of
12 a detailed configuration information table 136 is also shown in Figure 11. In the example shown,
13 server information 138 and kernel parameters 136 are displayed with detailed data pertaining to
14 the associated settings, kernels etc. An example of a detailed workload data table 145 is shown
15 in Figure 12. In the example shown in Figure 12, daily and hourly medium frequency workload
16 data 144 is listed in the table. Each property is listed such as "disk free" and "disk used" along
17 with the corresponding attributes.

18 **[00134]** The system compatibility analysis, which ultimately generates the SCI matrix 60,
19 compares the detailed configuration settings for the candidate servers 28 with other candidate
20 systems (not shown) based on a pre-defined rule set 43. The resulting analysis yields a system
21 compatibility index score for each candidate server combination. As noted above, these scores
22 are preferably arranged in the graphical SCI 60 format shown in Figure 6 and sorted using a
23 preferred sorting algorithm as discussed above.

24 **[00135]** As previously discussed, the analysis program 10 supports multiple rule sets to
25 address a variety of server consolidation strategies and scenarios. Example consolidation rule
26 sets include those for UNIX systems, Windows systems virtualization, SQL servers and Oracle
27 databases. As noted above, each rule specifies the important configuration property that is to be
28 compared between the candidate servers, the criteria for comparison, a relative weight to

21562813.1

1 compute the SCI score, a rule description, and any rule dependencies. Other fields may also be
2 considered such as the source and/or target instance, conditional flags, mutex etc. An example
3 rule set arranged in a table 146 for an arbitrary system is shown in Figure 13. Each rule and its
4 associated information is listed as a separate entry 148. Various OS settings (not all shown in
5 Figure 13) are assessed including the OS name, OS version, OS kernel bits, memory size, patch
6 level, name service settings, kernel parameters, locale settings, timezone settings etc.

7 **[00136]** In this example, total and shared memory criteria tests whether or not the target
8 system has less memory than hypothetical systems that it could be migrated to. If the rule
9 criterion is not explicitly specified, the rule simply checks whether or not the property values of
10 the two systems are different. In general, critical rules are assigned larger weights than less
11 critical rules. As can be seen from Figure 13, a different operating system is deemed to be more
12 important than a different time zone and thus is given a larger weight. The criticality of a rule
13 may vary from system to system and thus the rule sets should be flexible and configurable.

14 **[00137]** The SCI score is computed for each candidate system combination by evaluating the
15 rules based on the configuration data obtained from the audit of the environment 12. For each
16 rule that applies to the candidate server pair, the pair's SCI score is reduced by iteratively
17 applying the corresponding rule weight at each iteration (i.e. for each rule), from an initial value
18 to a final value, the final value being the score. A suitable formula for calculating the SCI score
19 is:

20 **[00138]** $SCI_{(n+1)} = SCI_n (1 - Weight) ;$

21 **[00139]** where SCI_n is the current SCI score (initially 100) before the next rule is evaluated,
22 SCI_{n+1} is the new SCI score, and Weight is the rule weight.

23 **[00140]** For example, for an arbitrary pair of systems, if the following rules: different
24 operating system (weight=0.5); not running the same kernel bits (weight=0.1); and target host
25 has less memory (weight=0.05); were found to apply to a pair of servers 28, the corresponding
26 SCI score would be computed as follows:

1 **[00141]** $SCI_1 = 100(1 - 0.5) = 50$

2 **[00142]** $SCI_2 = 50(1 - 0.1) = 45$

3 **[00143]** $SCI_3 = 45(1 - 0.05) = 42.75$

4 **[00144]** Final SCI score = 42.75.

5 **[00145]** As can be seen from the example above, where two servers 28 have different
6 operating systems, different kernel bits and the target has less memory than the baseline data, the
7 configuration compatibility for that server pair is quite low. It is therefore unlikely that these
8 servers could be consolidated since they are not compatible at the configuration level.

9 **[00146]** The compatibility program 10 provides a user interface for performing the overall
10 CHI analysis as shown in Figure 14. Through such an interface, users can specify the input
11 parameters required to generate the underlying SCI and WCI results to produce the CHI report.

12 **[00147]** In the example shown in Figure 14, the program 150 is used to generate a system
13 compatibility report by having the user select an Overall Compatibility Report type to perform
14 the CHI analysis on the selected target systems. An audit that has been saved is loaded into the
15 program 150, which lists identifying information such as the timestamp, number of target
16 servers, and the number of failed evaluations in the table 152. The user can select the report
17 category from the drop-down list 154, e.g. optimization, inventory, change, compliance,
18 administration etc; and can choose the report type from the drop-down list 156. The report
19 parameters 158 may also be selected which are specific to the type of report being generated and
20 define how the report is structured. The target information 162 is also shown and the target can
21 be removed. Once the user has made the appropriate selections, choosing the “Generate” button
22 creates the SCI matrix 60a shown in Figure 15.

23 **[00148]** The SCI matrix 60a is presented as an NxN matrix, where the top of each column
24 indicates the server on which each server listed in the first column is consolidated, and the row
25 name indicates the server being migrated. The SCI 60 shown in Figure 15 provides an example

matrix of scores for the example environment 12 including server A to server D. The SCI scores apply to each possible server consolidation pair. The higher scores indicate more compatibility.

[00149] As noted above, a group sort can be applied to the SCI matrix 60, which includes a difference calculation. The difference calculation between a pair of systems can be illustrated making reference to the SCI matrix 60a shown in Figure 15. From this example, the difference between server A and server B may be computed as follows:

[00150] $\text{Diff} = \sqrt{(\text{square}(S(a,a) - S(a,b)) + \text{square}(S(b,a) - S(b,b)) +$

[00151] $\text{square}(S(c,a) - S(c,b)) + \text{square}(S(d,a) - S(d,b)) +$

[00152] $\text{square}(S(a,a) - S(b,a)) + \text{square}(S(a,b) - S(b,b)) +$

[00153] $\text{square}(S(a,c) - S(b,c)) + \text{square}(S(a,d) - S(b,d))$

[00154] $= \sqrt{(\text{square}(100 - 90) + \text{square}(90 - 100) +$

[00155] $\text{square}(83 - 79) + \text{square}(52 - 52) +$

[00156] $\text{square}(100 - 90) + \text{square}(90 - 100) +$

[00157] $\text{square}(83 - 79) + \text{square}(57 - 57))$

[00158] where sqrt represents a square root operation.

[00159] The detailed differences shown in Figures 16 and 17 can be viewed by clicking on the relevant cell 66. As can be seen in the example, migrating server C to server D yields a score of 57. Selecting this cell accesses the detailed differences table 178 shown in Figure 17, which shows the important differences between the two systems, the rules and weights that were applied and preferably a remediation cost for making the servers more compatible, the data being collectively referred to by numeral 176. As shown in Figure 16, a summary differences table 170 may also be presented when selecting a particular cell, which lists the description of the differences 174 and the weight applied for each difference, to give a high level overview of where the differences arise.

21562813.1

1 **[00160]** The program 150 can also be used to generate a workload compatibility report, which
2 provides a WCI matrix 70a as shown in Figure 18. While the system compatibility analysis
3 considers the compatibility between systems from a configuration standpoint, the workload
4 analysis assesses the compatibility of two servers from a workload perspective.

5 **[00161]** The workload analysis computes the WCI scores for the environment 12, in each
6 server combination, similar to the SCI scores. The WCI scores identify the combinations of
7 systems with the most compatibility in consolidating their workloads onto a single machine.
8 This is based on the expected and historical workload patterns for the server and the capacity to
9 take on more work at different times in the pattern.

10 **[00162]** The workload analysis is performed by making the appropriate selections in the
11 program 150. The program 150, when instructed, uses the workload data collected during the
12 audit. The program 150 allows the user to specify the workload property to analyze (e.g. CPU
13 utilization) and the threshold (or usage limit) 160 to be applied when computing the WCI scores.
14 In the example shown in Figures 18 and 19, the WCI scores are penalized when the stacked CPU
15 utilization exceeds 75% of the target server's CPU capability.

16 **[00163]** The workload compatibility report generates the WCI matrix 70a as an NxN matrix
17 comparing the candidate servers 28. The report provides the WCI score and workload details for
18 each server combination. The WCI matrix columns represent the systems on which the
19 workloads are combined and the rows list the servers being migrated.

20 **[00164]** It can be seen from the example provided in Figure 18 that server A can best
21 accommodate the workload of the other servers. In contrast, server C and server D are
22 incompatible from a workload perspective. It should be noted that as seen above, server C and
23 server D were more compatible from a configuration standpoint than a workload. Even with a
24 100% compatibility in configuration settings, two servers may still score poorly in workload
25 compatibility because they may be sufficiently utilized.

26 **[00165]** The WCI scores are computed for each server combination, e.g. as follows. The
27 stacked peak workload values for the candidate servers are first computed for the maximum peak

(4th quartile) workloads at like times, the maximum peak (4th quartile) workloads at worst times, the maximum sustained (3rd quartile) workloads at like times, and the maximum sustained (3rd quartile) workloads at worst times. When stacking the server workloads comprising data such as CPU utilization, system benchmarks are preferably used to account for relative differences in the candidate servers' workload capabilities (e.g. SPECInt for CPU).

[00166] The user-specified workload threshold is then applied to the stacked workload values to derive the workload component scores. The component scores can then be used to calculate the aggregate WCI scores:

[00167] Aggregate score =

[00168] (0.2 * Peak 4th Quartile Like Score) *Peak Like Times Score*

[00169] + (0.1 * Peak 4th Quartile Worst Score) *Peak Worst Times Score*

[00170] + (0.4 * Peak 3rd Quartile Like Score) *Sustained Like Times Score*

[00171] + (0.3 * Peak 3rd Quartile Worst Score) *Sustained Worst Times Score*

[00172] The above shown weighting factors are for illustrative purposes only and are preferably flexible. The weighting factors represent the relative importance of peak versus sustained activity in a system 28 and as-measured versus time-shifted patterns.

[00173] An example workload compatibility report 180 showing the migration of server B to server C is shown in Figure 19. The report 180 indicates the inputs such as source server (the one to be stacked), target server (the one being stacked on), the threshold usage limit and normalization parameter(s) in a table 182. A normalization parameter is a number that is indicative of how powerful a system is 18 compared to other systems, and thus normalizes the scores based on the parameter. A WCI computation scorecard 184 is also shown in the report 180, which indicates the value for each quartile, its value and the corresponding score. The aggregate WCI is also provided, which in this example indicates a score of 70. The score (70) is also shown in the WCI matrix 70a shown in Figure 18. Preferably, by selecting a cell in the

WCI matrix 70, access is provided to the underlying calculations and data used to derive the WCI score for the corresponding server pair.

[00174] The overall compatibility of the environment 12 for this example is determined by combining the results of the system compatibility and workload compatibility analyses to derive the CHI matrix 80a shown in Figure 20. The CHI matrix 80a is a report that combines the SCI and WCI scores mathematically. For example, a multiply overlay algorithm can be used which multiplies the cell values together, as well as an “average” of the two scores. It will be appreciated that other algorithms may also be used. In the example shown, the CHI scores are the products of the corresponding SCI and WCI scores. The matrices 60, 70 and 80 can then be arranged in a visual display such as the roadmap 90 or in a single report that creates each matrix and hyperlinks them together.

[00175] It can be seen from the matrix shown in Figure 20 that the most promising consolidation pairs from the above example are server C to server A, server D to server A, server A to server C, and server A to server D. It should be noted when reviewing Figures 15, 18 and 20 together that even though server C and server D are very compatible from a configuration standpoint (SCI=90), they are a poor consolidation pair (CHI=35) due to incompatible workloads (WCI=39).

Commentary

[00176] Accordingly, a consolidation analysis can be performed and visually represented by performing an audit, performing system compatibility and workload compatibility analyses using the audit data and combining the system and workload analyses to create an overall compatibility co-habitation analyses. It will be appreciated that although the system and workload analyses are performed in this example to contribute to the overall co-habitation analyses, each analysis is suitable to be performed on its own and can be conducted separately for finer analyses. The finer analysis may be performed to focus on the remediation of only configuration settings at one time and spreading workload at another time. As such, each analysis and associated matrix may be generated on an individual basis without the need to perform the other analyses.

1 [00177] It will be appreciated that each analysis and associated matrix may instead be used for
2 purposes other than consolidation such as capacity planning, regulatory compliance, change,
3 inventory, optimization, administration etc. and any other purpose where compatibility of
4 systems is useful for analyzing systems 28. It will also be appreciated that the program 10 may
5 also be configured to allow user-entered attributes (e.g. location) that are not available via the
6 auditing process and can factor such attributes into the rules and subsequent analysis.

7 [00178] It will further be appreciated that although the examples provided above are in the
8 context of a distributed system of servers, the principles and algorithms discusses are applicable
9 to any system having a plurality of sub-systems where the sub-systems perform similar tasks and
10 thus are capable theoretically of being consolidation. For example, a local network having a
11 number of personal computers (PCs) could also benefit from a consolidation analysis.

12 [00179] Although the invention has been described with reference to certain specific
13 embodiments, various modifications thereof will be apparent to those skilled in the art without
14 departing from the spirit and scope of the invention as outlined in the claims appended hereto.

What is claimed is:

1. A method for determining compatibilities for a plurality of computer systems comprising:
 - generating a configuration compatibility score for each pair of said plurality of systems based on configuration data obtained for each of said plurality of systems;
 - generating a workload compatibility score for each pair of said plurality of systems based on workload data obtained for each of said plurality of systems; and
 - generating a co-habitation score for each pair of said plurality of systems using the respective configuration compatibility score and workload compatibility score, said co-habitation score indicating an overall compatibility for each system with respect to the others of said plurality of systems.
2. A method according to claim 1 further comprising generating a graphical interface for displaying said compatibility scores, said graphical interface comprising a matrix of cells, each said cell corresponding to a pair of said plurality of computer systems, each row of said matrix indicating one of said plurality of computer systems and each column of said matrix indicating one of said plurality of computer systems, each cell displaying a compatibility score indicating the compatibility of the respective pair of said plurality of systems indicated in the corresponding row and column, and computed according to predefined criteria.
3. A method according to claim 2 wherein upon selecting said cells, information related to the respective score is provided.
4. A method according to claim 1 further comprising generating a report indicative of said compatibilities according to said compatibility scores.
5. A method according to claim 1 wherein said configuration compatibility score is computed by applying at least one rule set comprising one or more parameters in said configuration data.

6. A method according to claim 5 further comprising assigning a weight to each said parameter indicative of the importance of said parameter to the compatibility of said plurality of systems, said configuration compatibility score being computed for each pair of said plurality of computer systems according to said weights.
7. A method according to claim 5 wherein said configuration compatibility scores are computed by iteratively decrementing an initial value to produce a final value and applying one of said weights at each iteration.
8. A method according to claim 1 wherein said workload compatibility score is computed by computing a stacked workload value for each pair of said plurality of systems at one or more time instance according to said workload data.
9. A method according to claim 8 wherein said stacked workload value is computed according to maximum sustained workloads and maximum peak workloads for each said pair.
10. A method according to claim 9 wherein said stacked workload value considers said maximum sustained workloads at like times and worst times, and considers said maximum peak workloads at like times and worst times.
11. A method according to claim 1 further comprising performing an audit of each of said plurality of systems to obtain said configuration data and said workload data.
12. A method according to claim 1 further comprising stacking said workload data for a plurality of source systems on a target system to virtualize a multi-stacked system, and generating a workload compatibility score for said multi-stacked system based on said stacked workload data.
13. A computer program for determining compatibilities for a plurality of computer systems comprising an audit engine for obtaining information pertaining to the compatibility of said

plurality of computer systems; an analysis engine for generating a compatibility score for each pair of said plurality of systems based on said information that is specific to respective pairs; and a client for displaying said compatibility score on an interface.

14. A computer program according to claim 13 further comprising a report generator for generating a report comprising said compatibility scores according to a report template.
15. A computer program according to claim 14, said client comprising a report generation program for indicating the type of analysis to be performed by said analysis engine and compiled by said report generator.
16. A computer program according to claim 13 wherein said information comprises configuration data.
17. A computer program according to claim 13 wherein said information comprises workload data.
18. A computer program according to claim 13 wherein said information comprises configuration and workload data and said compatibility score is computed by combining a configuration compatibility score based on said configuration data and a workload compatibility score based on said workload data.
19. A method for determining configuration compatibilities for a plurality of computer systems comprising:
 - obtaining configuration data for each of said plurality of computer systems;
 - assigning a weight to one or more parameter in said configuration data indicating the importance of said parameter to the compatibility of said plurality of systems;
 - generating a rule set comprising one or more of said parameters; and
 - computing a configuration score for each pair of said plurality of systems according to the weights in said rule set.

20. A method according to claim 19 wherein said configuration scores are computed by iteratively decrementing an initial value to produce a final value and applying one of said weights at each iteration.
21. A method for determining workload compatibilities for a plurality of computer systems comprising:
- obtaining workload data for each of said plurality of systems;
 - computing a stacked workload value for each pair of said plurality of systems at one or more time instance according to said workload data; and
 - computing a workload score for each pair of said plurality of systems using said stacked workload values.
22. A method according to claim 21 wherein said stacked workload value is computed according to maximum sustained workloads and maximum peak workloads for each said pair.
23. A method according to claim 22 wherein said stacked workload value considers said maximum sustained workloads at like times and worst times, and considers said maximum peak workloads at like times and worst times.
24. A method according to claim 21 further comprising stacking said workload data for a plurality of source systems on a target system to virtualize a multi-stacked system, and generating said workload score for said multi-stacked system based on said stacked workload data.
25. A graphical interface for displaying compatibility scores for a plurality of computer systems comprising a matrix of cells, each said cell corresponding to a pair of said plurality of computer systems, each row of said matrix indicating one of said plurality of computer systems and each column of said matrix indicating one of said plurality of computer systems, each cell displaying a compatibility score indicating the compatibility of the respective pair

of said plurality of systems indicated in the corresponding row and column, and computed according to predefined criteria.

26. A graphical interface according to claim 25 wherein upon selecting said cells, information related to the respective score is provided.
27. A graphical interface according to claim 25 wherein each said cell comprises a background pattern chosen from a plurality of background patterns based on said compatibility score.
28. A graphical interface according to claim 27 wherein said background patterns are different colours, said colours each representing a range of score values.
29. A graphical interface according to claim 28 further comprising a tool for selectively changing said range.
30. A graphical interface according to claim 25 wherein said compatibility score is based on one of configuration data, workload data and a combination of configuration and workload data.

1 ABSTRACT

2
3 Systems and method for performing a consolidation analysis and visually representing the
4 analysis are provided. The analysis is conducted by performing an audit, performing system
5 compatibility and workload compatibility analyses using the audit data and rule sets. The results
6 of the individual analyses are combined to create an overall compatibility co-habitation analysis
7 and visual consolidation roadmap including scorecard information. The system and workload
8 analyses may be performed to contribute to the overall co-habitation analyses, but also
9 performed on their own.

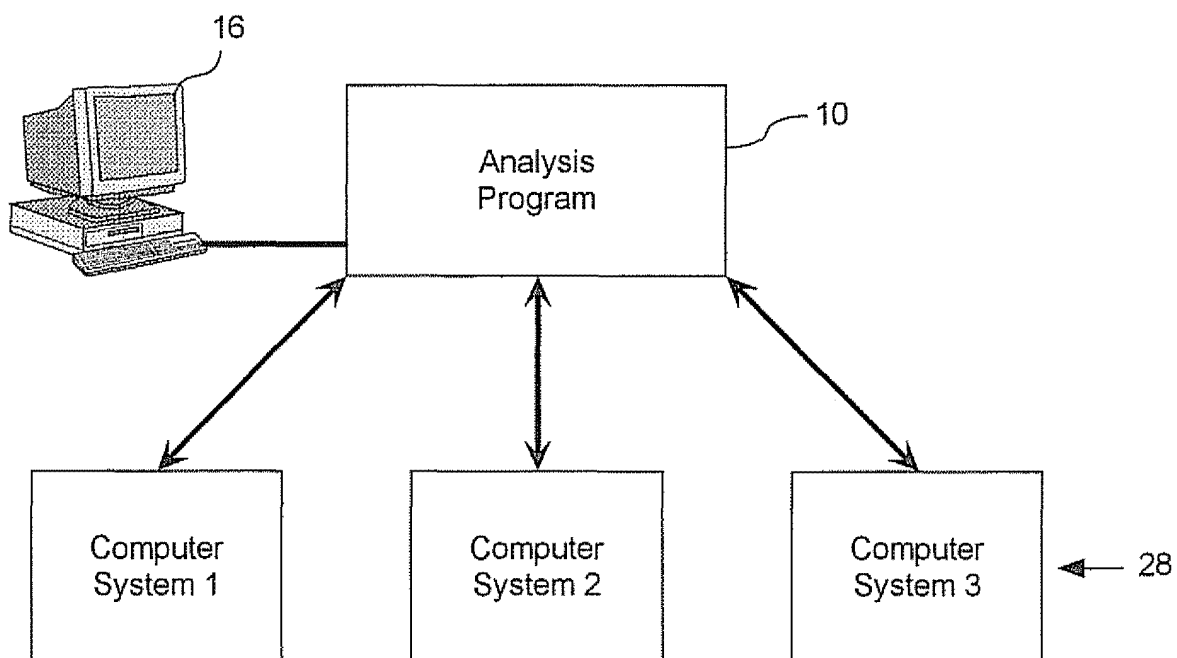


Figure 1a

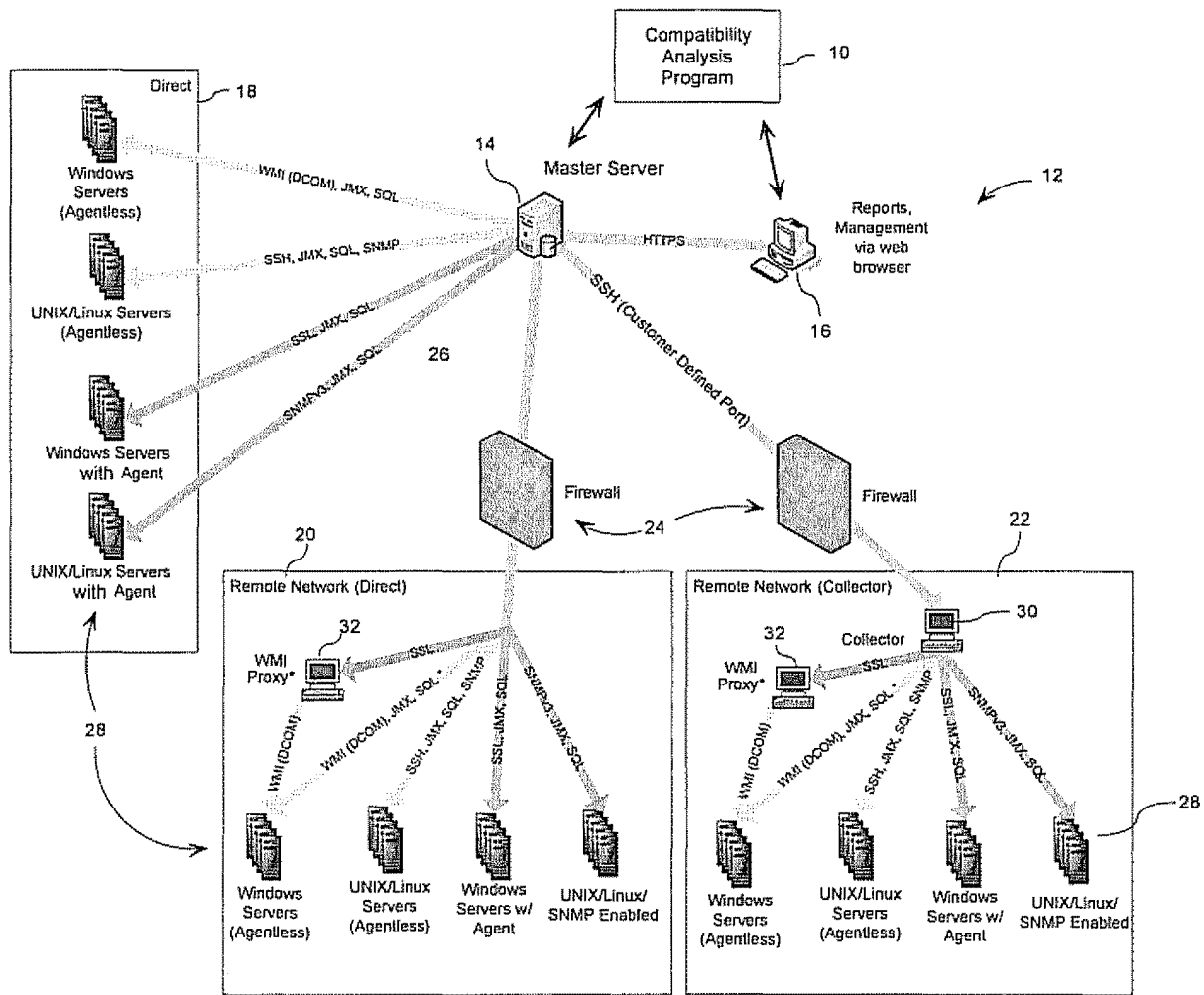


Figure 1b

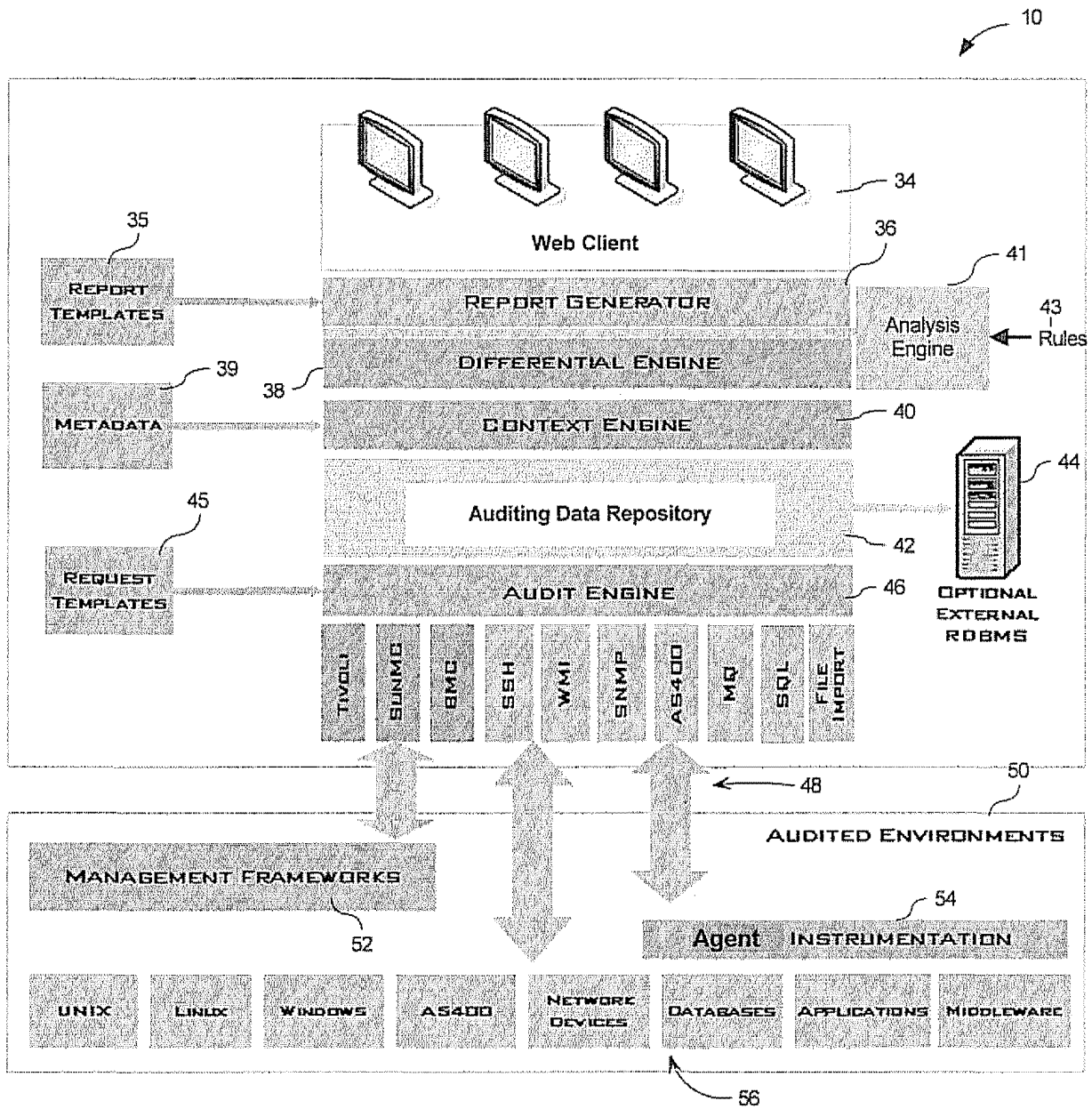


Figure 2

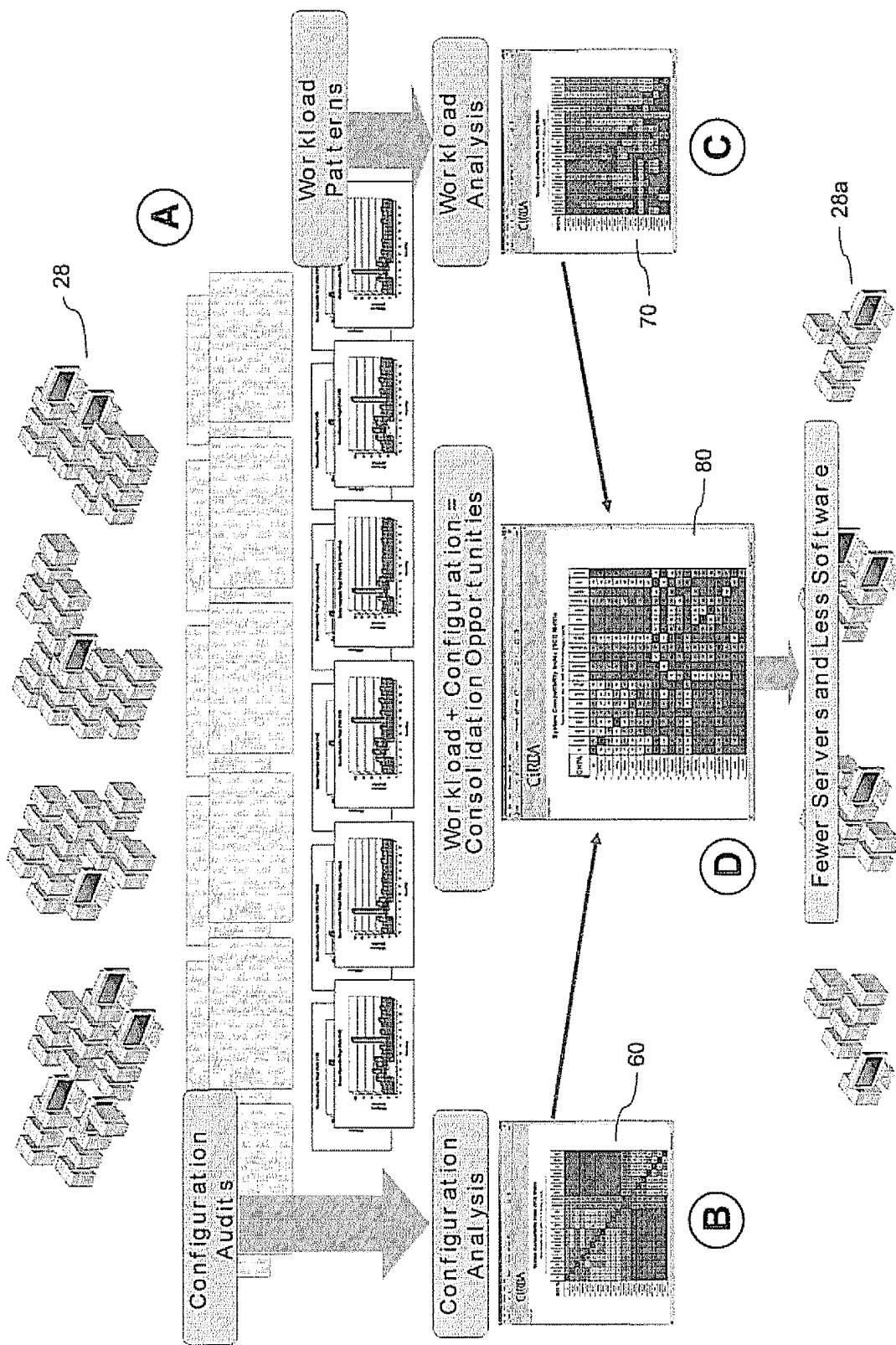


Figure 3

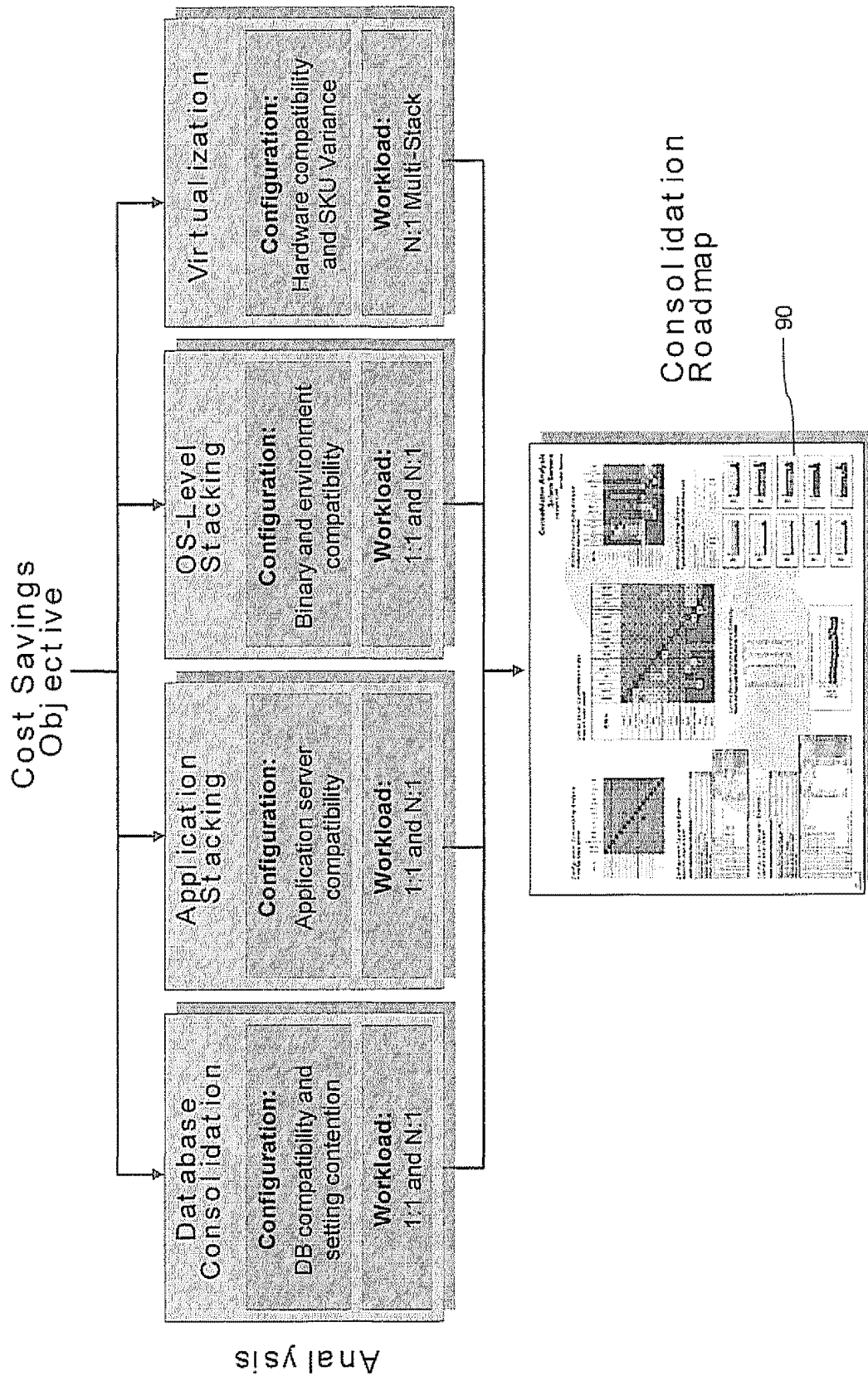


Figure 4

SCI %	00000000	00000001	00000010	00000011	00000100	00000101	00000110	00000111	00001000	00001001	00001010	00001011	00001100	00001101	00001110	00001111	00010000	00010001	00010010	00010011	00010100	00010101	00010110	00010111	00011000	00011001	00011010	00011011	00011100	00011101	00011110	00011111	00100000	00100001	00100010	00100011	00100100	00100101	00100110	00100111	00101000	00101001	00101010	00101011	00101100	00101101	00101110	00101111	00110000	00110001	00110010	00110011	00110100	00110101	00110110	00110111	00111000	00111001	00111010	00111011	00111100	00111101	00111110	00111111	01000000	01000001	01000010	01000011	01000100	01000101	01000110	01000111	01001000	01001001	01001010	01001011	01001100	01001101	01001110	01001111	01010000	01010001	01010010	01010011	01010100	01010101	01010110	01010111	01011000	01011001	01011010	01011011	01011100	01011101	01011110	01011111	01100000	01100001	01100010	01100011	01100100	01100101	01100110	01100111	01101000	01101001	01101010	01101011	01101100	01101101	01101110	01101111	01110000	01110001	01110010	01110011	01110100	01110101	01110110	01110111	01111000	01111001	01111010	01111011	01111100	01111101	01111110	01111111	10000000	10000001	10000010	10000011	10000100	10000101	10000110	10000111	10001000	10001001	10001010	10001011	10001100	10001101	10001110	10001111	10010000	10010001	10010010	10010011	10010100	10010101	10010110	10010111	10011000	10011001	10011010	10011011	10011100	10011101	10011110	10011111	10100000	10100001	10100010	10100011	10100100	10100101	10100110	10100111	10101000	10101001	10101010	10101011	10101100	10101101	10101110	10101111	10110000	10110001	10110010	10110011	10110100	10110101	10110110	10110111	10111000	10111001	10111010	10111011	10111100	10111101	10111110	10111111	11000000	11000001	11000010	11000011	11000100	11000101	11000110	11000111	11001000	11001001	11001010	11001011	11001100	11001101	11001110	11001111	11010000	11010001	11010010	11010011	11010100	11010101	11010110	11010111	11011000	11011001	11011010	11011011	11011100	11011101	11011110	11011111	11100000	11100001	11100010	11100011	11100100	11100101	11100110	11100111	11101000	11101001	11101010	11101011	11101100	11101101	11101110	11101111	11110000	11110001	11110010	11110011	11110100	11110101	11110110	11110111	11111000	11111001	11111010	11111011	11111100	11111101	11111110	11111111
00000000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF										
00000001	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF											
00000010	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF												
00000011	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF													
00000100	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF														
00000101	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75																																																																																																																																															

[illegible]

60 Configuration Compatibility Analysis:
UNIX Systems

System	Model	OS	Kernel	Architecture	Processor	Memory	Storage	Network	Other
System 1	Model 1	OS 1	Kernel 1	Architecture 1	Processor 1	Memory 1	Storage 1	Network 1	Other 1
System 2	Model 2	OS 2	Kernel 2	Architecture 2	Processor 2	Memory 2	Storage 2	Network 2	Other 2
System 3	Model 3	OS 3	Kernel 3	Architecture 3	Processor 3	Memory 3	Storage 3	Network 3	Other 3
System 4	Model 4	OS 4	Kernel 4	Architecture 4	Processor 4	Memory 4	Storage 4	Network 4	Other 4
System 5	Model 5	OS 5	Kernel 5	Architecture 5	Processor 5	Memory 5	Storage 5	Network 5	Other 5
System 6	Model 6	OS 6	Kernel 6	Architecture 6	Processor 6	Memory 6	Storage 6	Network 6	Other 6
System 7	Model 7	OS 7	Kernel 7	Architecture 7	Processor 7	Memory 7	Storage 7	Network 7	Other 7
System 8	Model 8	OS 8	Kernel 8	Architecture 8	Processor 8	Memory 8	Storage 8	Network 8	Other 8
System 9	Model 9	OS 9	Kernel 9	Architecture 9	Processor 9	Memory 9	Storage 9	Network 9	Other 9
System 10	Model 10	OS 10	Kernel 10	Architecture 10	Processor 10	Memory 10	Storage 10	Network 10	Other 10

Configuration Deviation Example:
sunq01 versus sunq02

Property	sunq01	sunq02
Model	Model 1	Model 2
OS	OS 1	OS 2
Kernel	Kernel 1	Kernel 2
Architecture	Architecture 1	Architecture 2
Processor	Processor 1	Processor 2
Memory	Memory 1	Memory 2
Storage	Storage 1	Storage 2
Network	Network 1	Network 2
Other	Other 1	Other 2

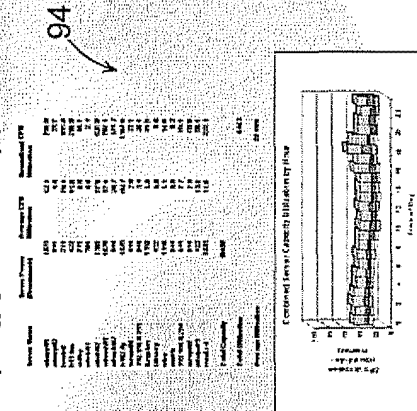
Configuration Deviation Example:
Kilberry versus wiso

Property	Kilberry	wiso
Model	Model 1	Model 2
OS	OS 1	OS 2
Kernel	Kernel 1	Kernel 2
Architecture	Architecture 1	Architecture 2
Processor	Processor 1	Processor 2
Memory	Memory 1	Memory 2
Storage	Storage 1	Storage 2
Network	Network 1	Network 2
Other	Other 1	Other 2

80 Overall Server Co-Habitation Index:
UNIX Systems

System	Model	OS	Kernel	Architecture	Processor	Memory	Storage	Network	Other
System 1	Model 1	OS 1	Kernel 1	Architecture 1	Processor 1	Memory 1	Storage 1	Network 1	Other 1
System 2	Model 2	OS 2	Kernel 2	Architecture 2	Processor 2	Memory 2	Storage 2	Network 2	Other 2
System 3	Model 3	OS 3	Kernel 3	Architecture 3	Processor 3	Memory 3	Storage 3	Network 3	Other 3
System 4	Model 4	OS 4	Kernel 4	Architecture 4	Processor 4	Memory 4	Storage 4	Network 4	Other 4
System 5	Model 5	OS 5	Kernel 5	Architecture 5	Processor 5	Memory 5	Storage 5	Network 5	Other 5
System 6	Model 6	OS 6	Kernel 6	Architecture 6	Processor 6	Memory 6	Storage 6	Network 6	Other 6
System 7	Model 7	OS 7	Kernel 7	Architecture 7	Processor 7	Memory 7	Storage 7	Network 7	Other 7
System 8	Model 8	OS 8	Kernel 8	Architecture 8	Processor 8	Memory 8	Storage 8	Network 8	Other 8
System 9	Model 9	OS 9	Kernel 9	Architecture 9	Processor 9	Memory 9	Storage 9	Network 9	Other 9
System 10	Model 10	OS 10	Kernel 10	Architecture 10	Processor 10	Memory 10	Storage 10	Network 10	Other 10

Current Server Utilization versus Capacity:
Hourly and Aggregate Server Utilization by System



70 Workload Compatibility Analysis:
UNIX Systems

System	Model	OS	Kernel	Architecture	Processor	Memory	Storage	Network	Other
System 1	Model 1	OS 1	Kernel 1	Architecture 1	Processor 1	Memory 1	Storage 1	Network 1	Other 1
System 2	Model 2	OS 2	Kernel 2	Architecture 2	Processor 2	Memory 2	Storage 2	Network 2	Other 2
System 3	Model 3	OS 3	Kernel 3	Architecture 3	Processor 3	Memory 3	Storage 3	Network 3	Other 3
System 4	Model 4	OS 4	Kernel 4	Architecture 4	Processor 4	Memory 4	Storage 4	Network 4	Other 4
System 5	Model 5	OS 5	Kernel 5	Architecture 5	Processor 5	Memory 5	Storage 5	Network 5	Other 5
System 6	Model 6	OS 6	Kernel 6	Architecture 6	Processor 6	Memory 6	Storage 6	Network 6	Other 6
System 7	Model 7	OS 7	Kernel 7	Architecture 7	Processor 7	Memory 7	Storage 7	Network 7	Other 7
System 8	Model 8	OS 8	Kernel 8	Architecture 8	Processor 8	Memory 8	Storage 8	Network 8	Other 8
System 9	Model 9	OS 9	Kernel 9	Architecture 9	Processor 9	Memory 9	Storage 9	Network 9	Other 9
System 10	Model 10	OS 10	Kernel 10	Architecture 10	Processor 10	Memory 10	Storage 10	Network 10	Other 10

96 Workload Stacking Examples:
Good combination (left) and bad combination (right)

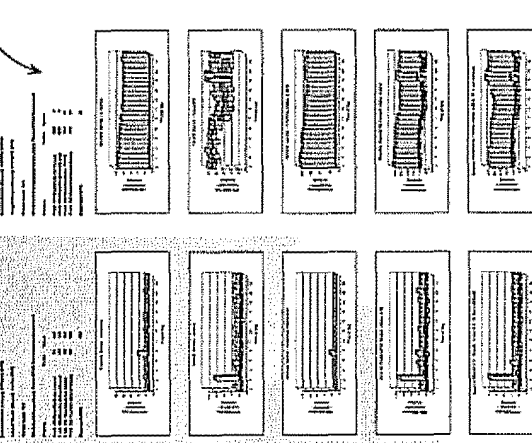


Figure 9

AUDIT NAME:
COMPANY X

DESCRIPTION:
Server Consolidation

REQUEST TYPE:
SNMP ☒
← 116

REQUEST PARAMETERS:

NAME	Value
SNMP Version	1
Port	3161
Comm. String	public

REQUEST TEMPLATES:

/Database_Settings/IETFNetwork_Services
 /Installed_Patches/Patch_Auditor
 /System_Information/Security_Details
 .
 .
 .
 .
 .
 .
 .
 .

EDIT

AUDIT TARGETS:

Server A
 Server B
 Server C
 server D

EDIT

SAVE
AUDIT
CANCEL

FIGURE 10

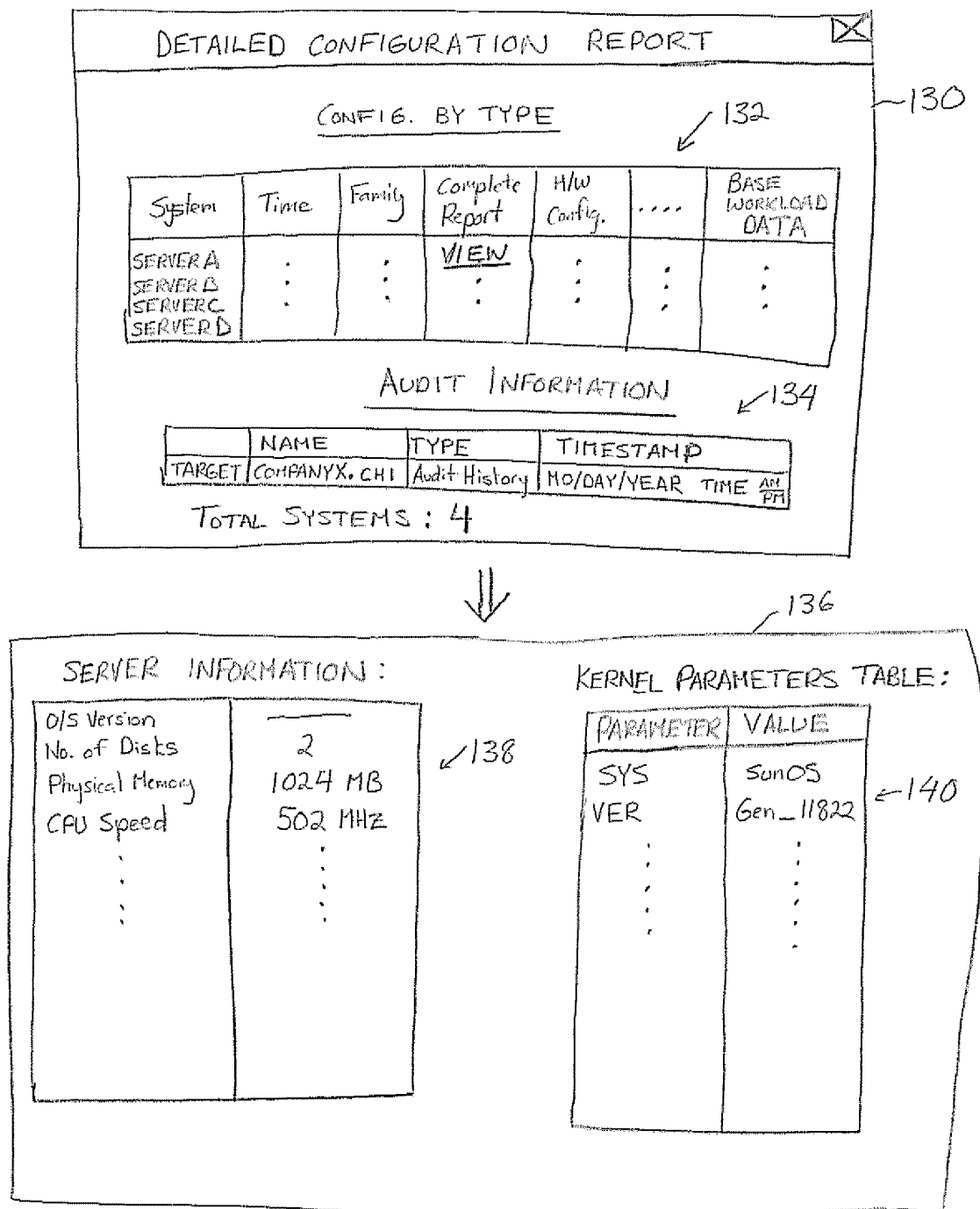


FIGURE 11

145

Property Name	Property Instance	DAILY MIN	Daily 1st Quartile	Daily Median	...	Daily Max.	Daily Average	Hourly MIN	...
disk free	/data	46971097	—	—		—	—		
disk used	/data	9158733	—	—		—	—		
.		
.		
.		
.		
.		

144 ↗

FIGURE 12

146

DATA ELEMENT	CONDITIONAL FLAGS	SOURCE, TARGET, INSTANCE	WEIGHT	DESCRIPTION	Remediation Cost	MUTEX
SYS_NAME			0.5	Diff. O/S	—	
O/S_Kernel_Bits			0.1	Not same kernel bits	—	
TIME_ZONE			0.02	Different time zones	—	
.			.	.		
.			.	.		
.			.	.		
.			.	.		
.			.	.		
.			.	.		
.			.	.		
.			.	.		

148 —

FIGURE 13

AUDIT NAME : COMPANYX.CHI

TIMESTAMP	TARGETS	FAILED
MO/DAY/YEAR TIME	4	0
<input type="button" value="DELETE"/>	152	

REPORT CATEGORY: 154

REPORT TYPE: 156

REPORT PARAMETERS:

NAME	Value
ABC123 158	<input type="button" value="UNIX"/> 160 CPU Utilization

TARGETS: 160

| CompanyX.CHI | AUDIT History | MO/DAY/YEAR TIME

164 162

FIGURE 14

SCI SCORE CARD:

SORT ROW	SORT COLUMN			
SCI %	SERVER A	SERVER B	SERVER C	SERVER D
SERVER A	100	90	83	57
SERVER B	90	100	79	57
SERVER C	83	79	100	90
SERVER D	52	52	52	100

60a

FIGURE 15

DIFFERENCES: (MIGRATING A to D)

DESCRIPTION	WEIGHT
DIFFERENT O/S Versions	30%
DIFFERENT TIME ZONES	2%
⋮	⋮
⋮	⋮
⋮	⋮

174 → (points to the table)

172 → (points to 30%)

170 → (points to the vertical ellipsis in the weight column)

FIGURE 16

DIFFERENCE DETAILS:

Module	Object	Property	Instance	Baseline	Target	Weight	Remed. Cost
Config. Monitor	Name Service Locale settings	Value Time Zone	ipnodes	files US/ Eastern	files.dns Canada/ Eastern	2%	—
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

178 → (points to the title)

176 → (points to the Remed. Cost column)

FIGURE 17

WCI SCORECARD:
 SORT Row | Sort Column

WCI %	SERVER A	SERVER B	SERVER C	SERVER D
SERVER A	100	76	73	66
SERVER B	84	100	70	60
SERVER C	82	70	100	39
SERVER D	80	60	39	100

FIGURE 18

180

WORKLOAD COMPATIBILITY REPORT

X

INPUTS:

NORMALIZATION	
TARGET SERVER	SERVER C
SOURCE SERVER	SERVER B
USAGE LIMIT	75%

182

WCI COMPUTATION SCORECARD:

CRITERIA	VALUE	SCORE
Peak 4th Quartile (like)	197	0
Peak 4th Quartile (worst)	198	0
Peak 3rd Quartile (like)	50	100
Peak 3rd Quartile (worst)	59	100
Aggregate WCI		70

184

FIGURE 19

CHI SCORE CARD :

CHI %	SERVER A	SERVER B	SERVER C	SERVER D
SERVER A	100	43	61	52
SERVER B	44	100	36	31
SERVER C	65	40	100	35
SERVER D	63	34	35	100

- 80a

FIGURE 20

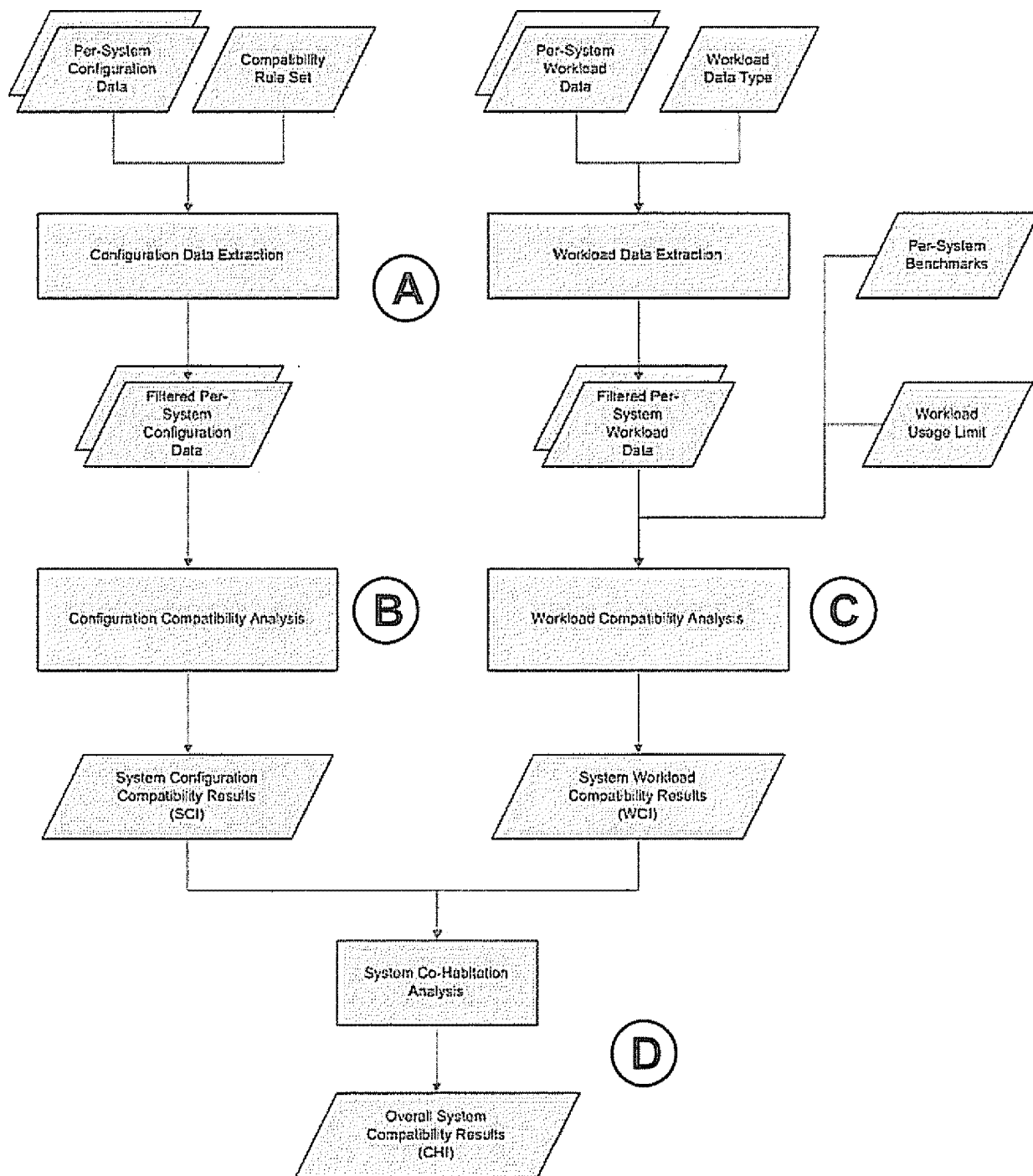


Figure 21

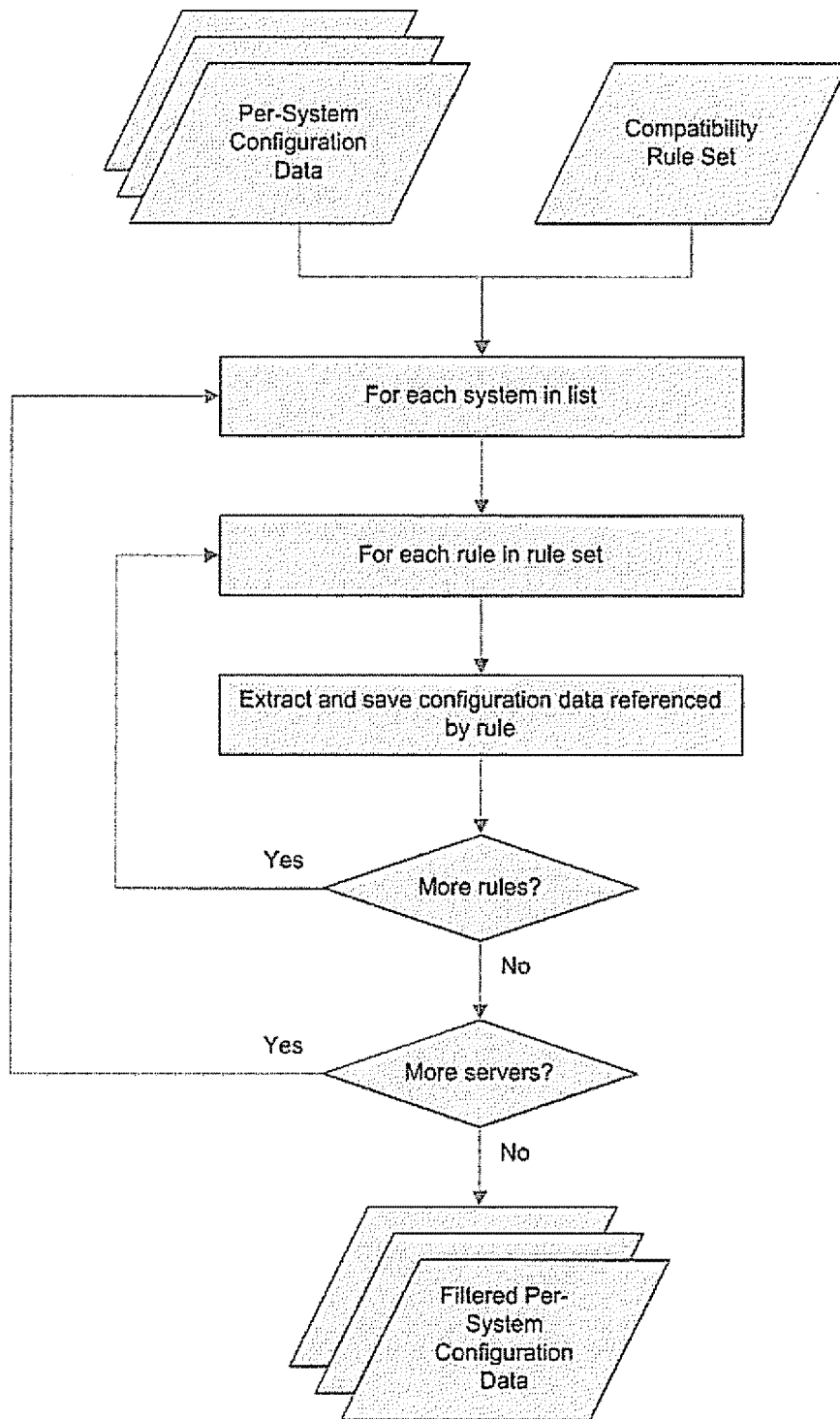


Figure 22

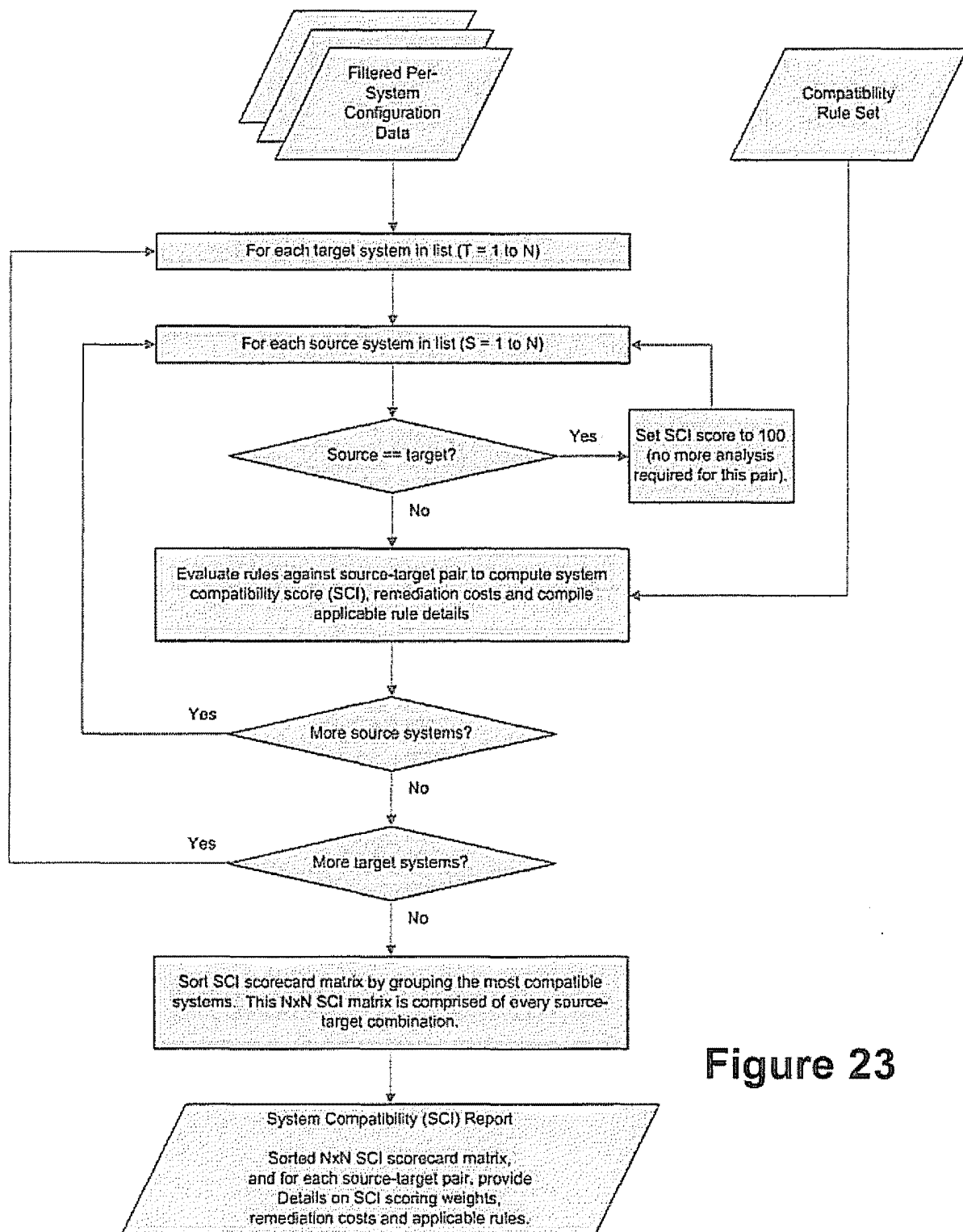


Figure 23

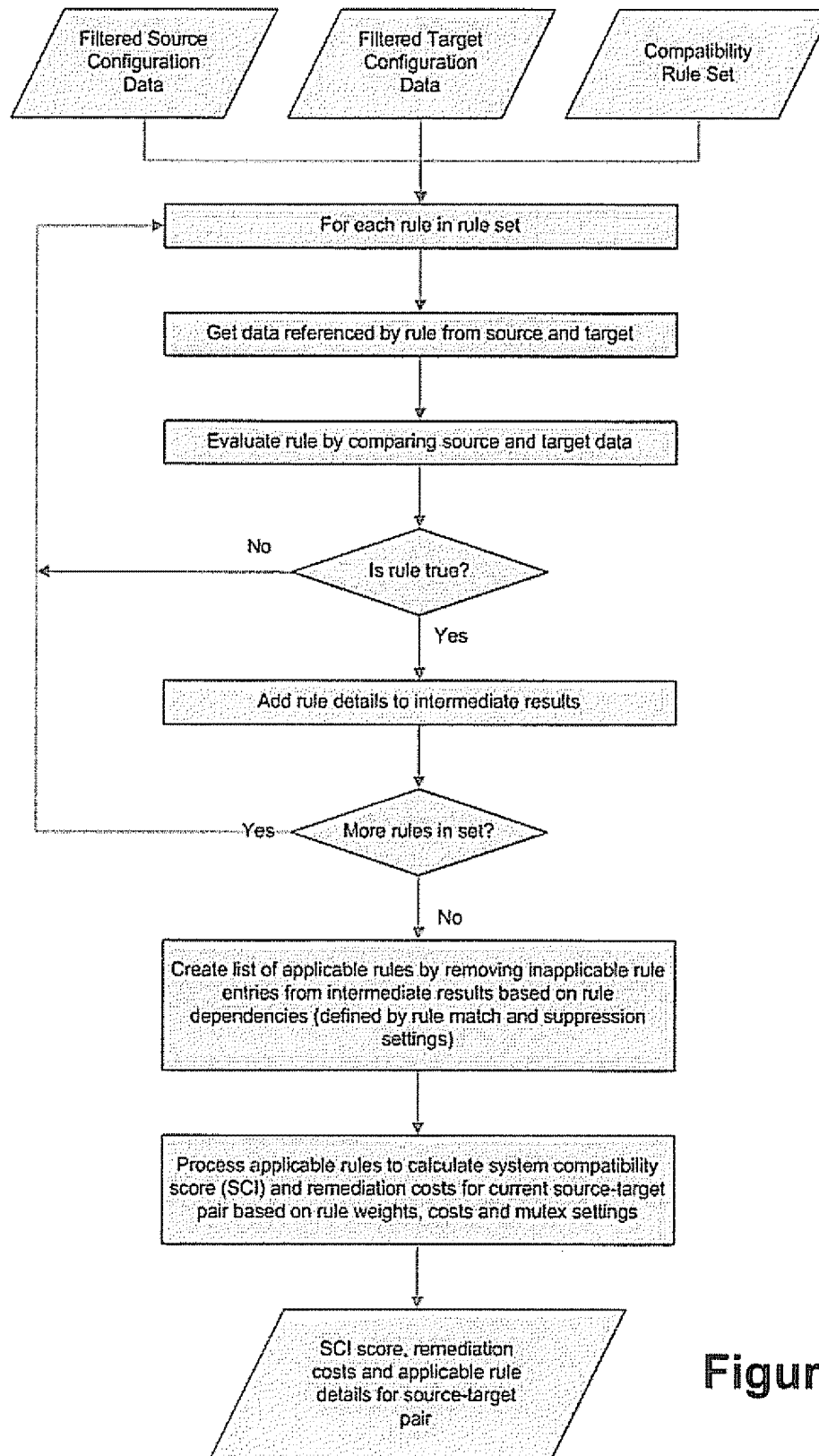


Figure 24

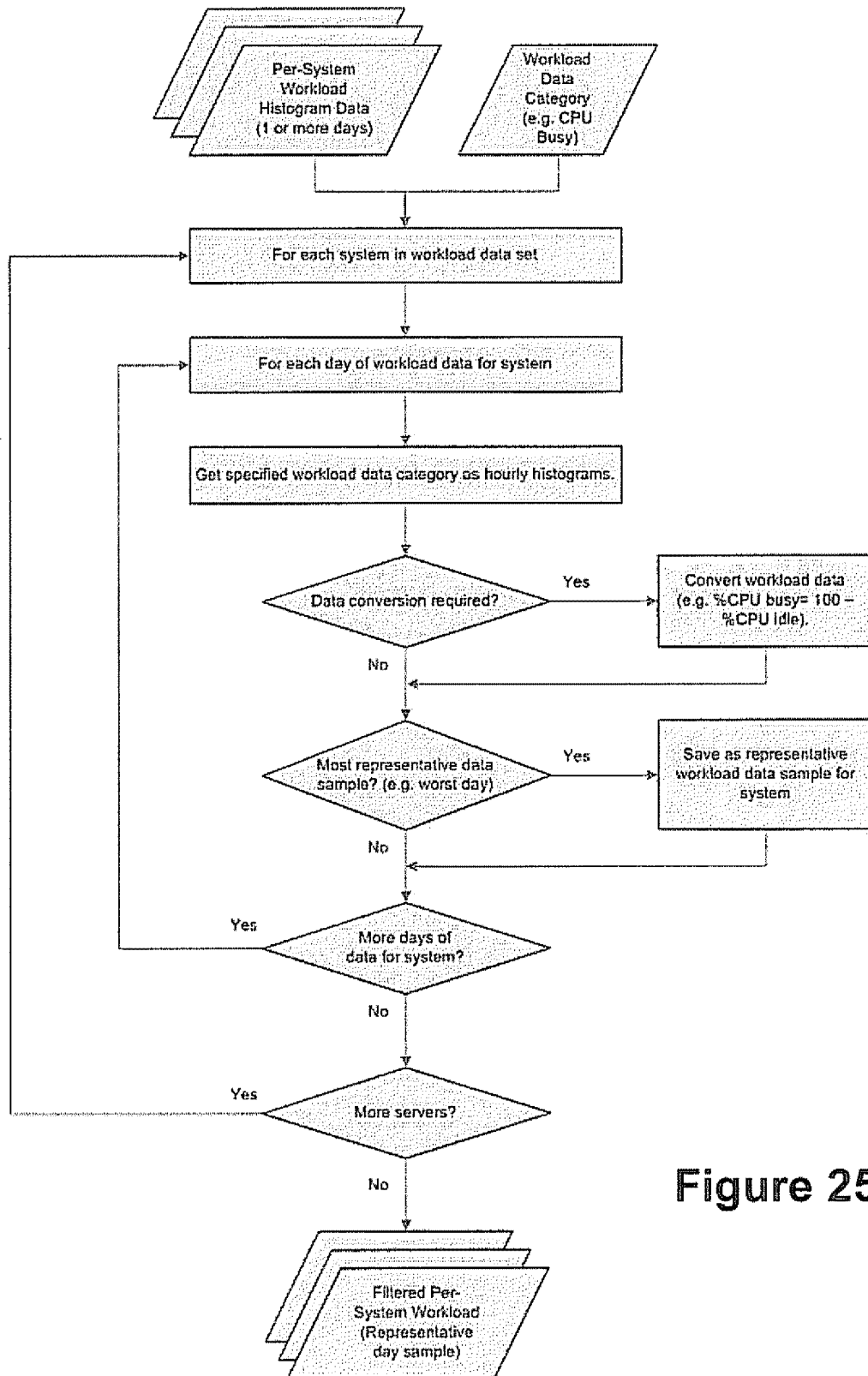


Figure 25

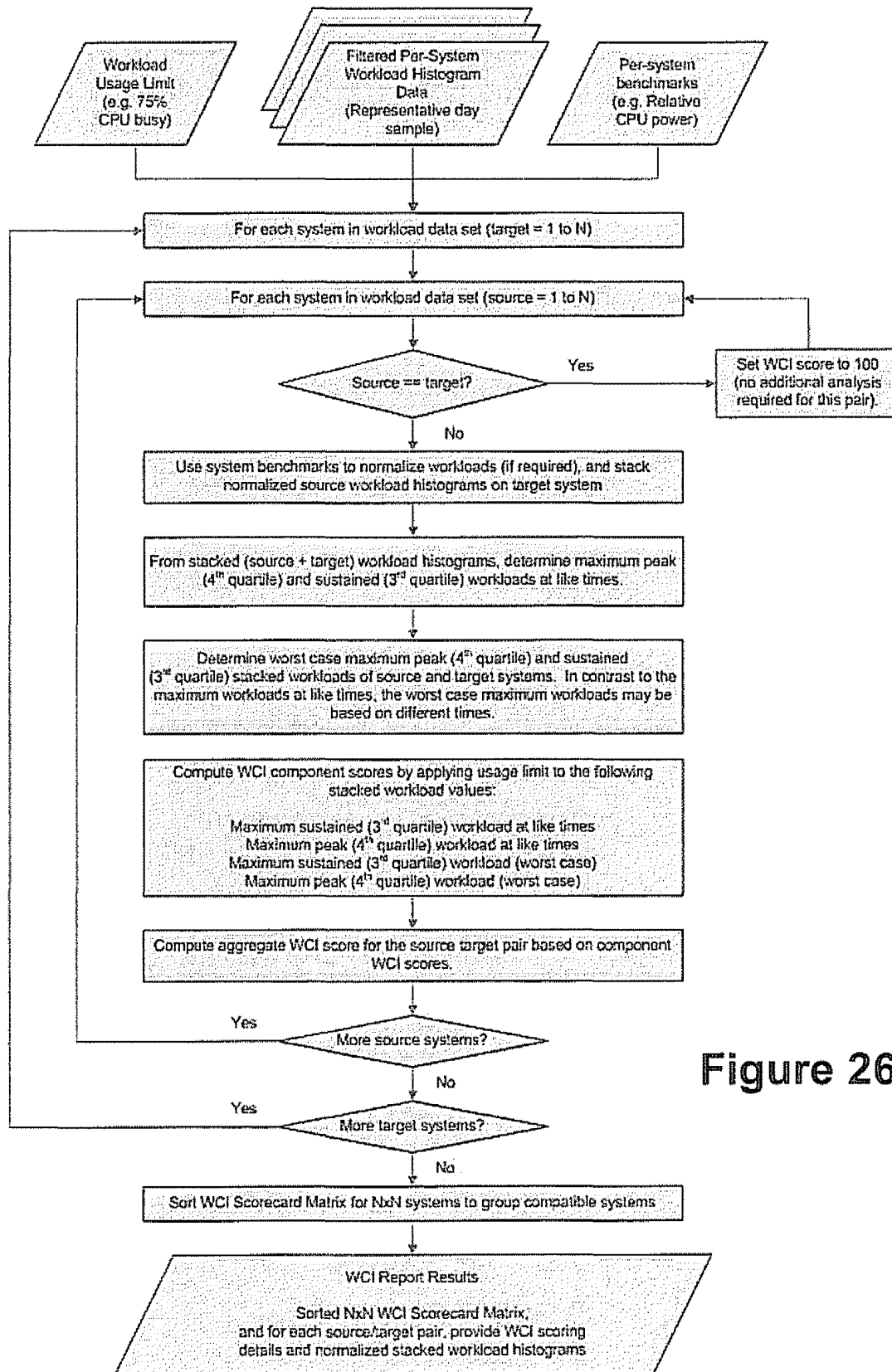


Figure 26

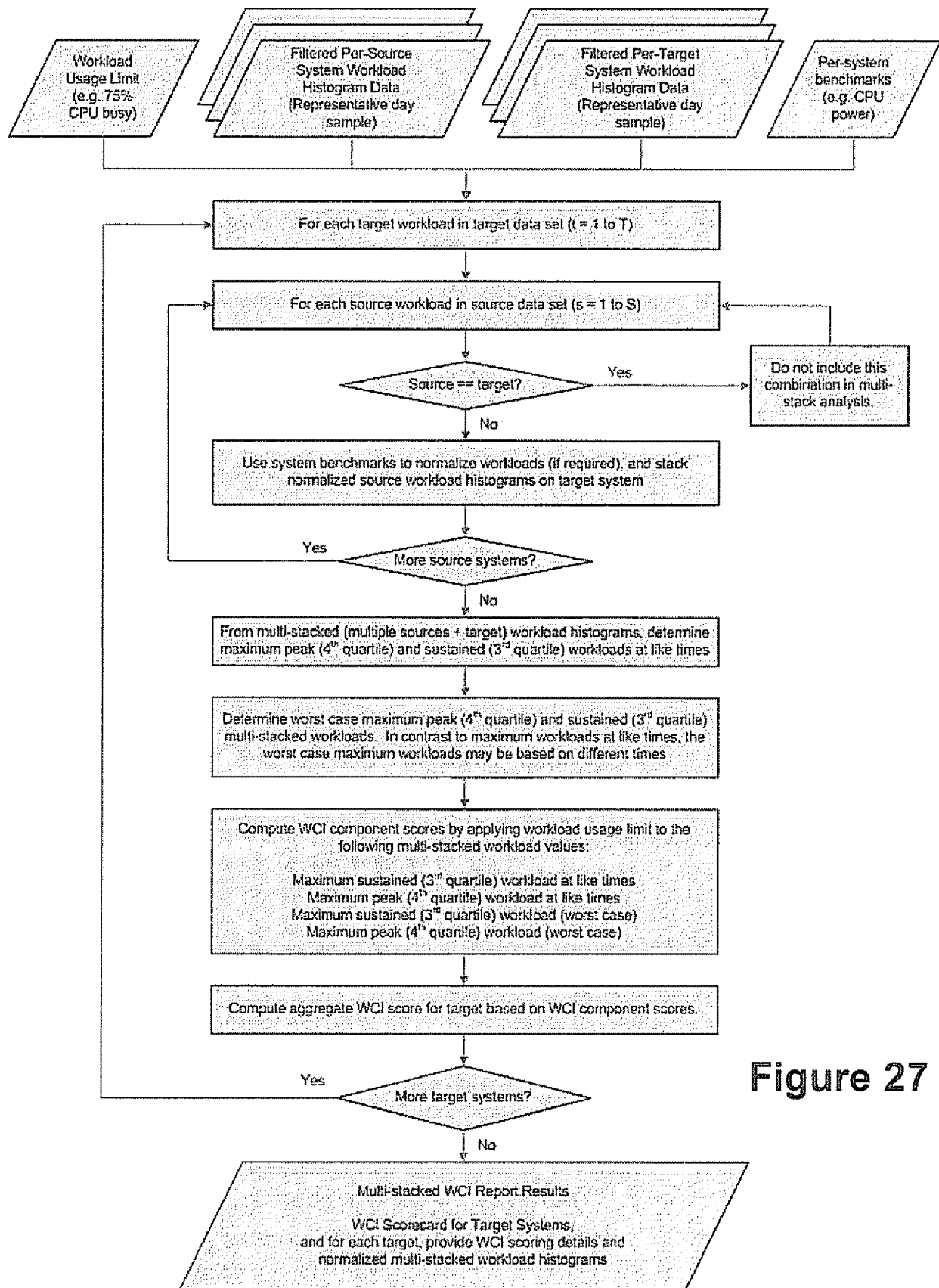


Figure 27

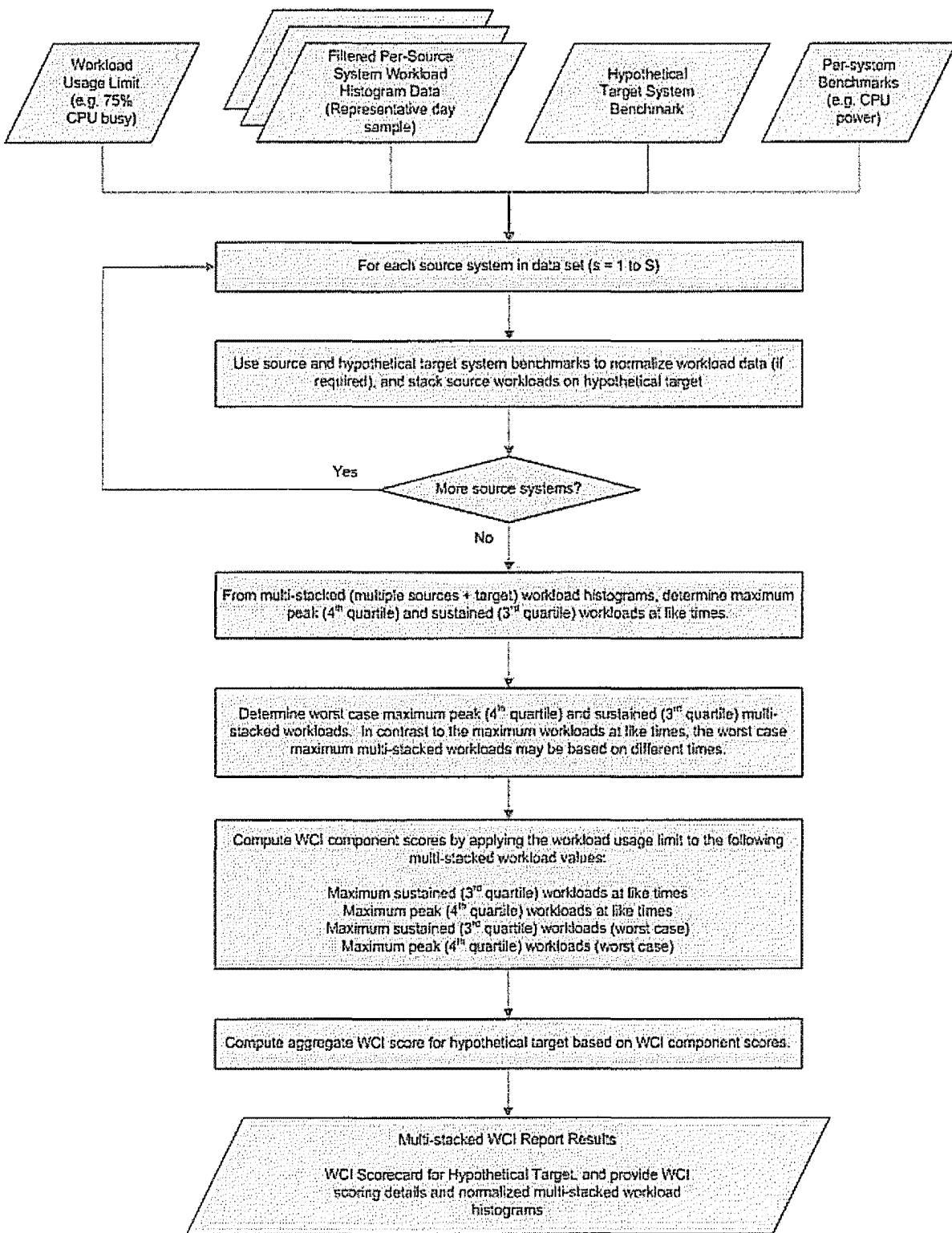


Figure 28

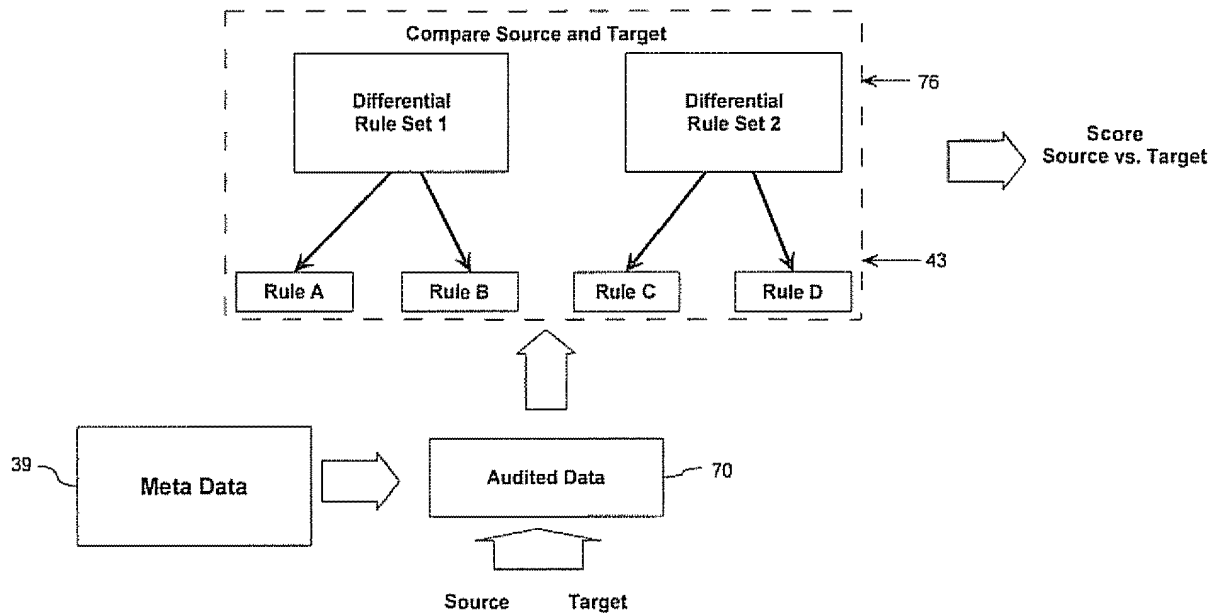


Figure 29

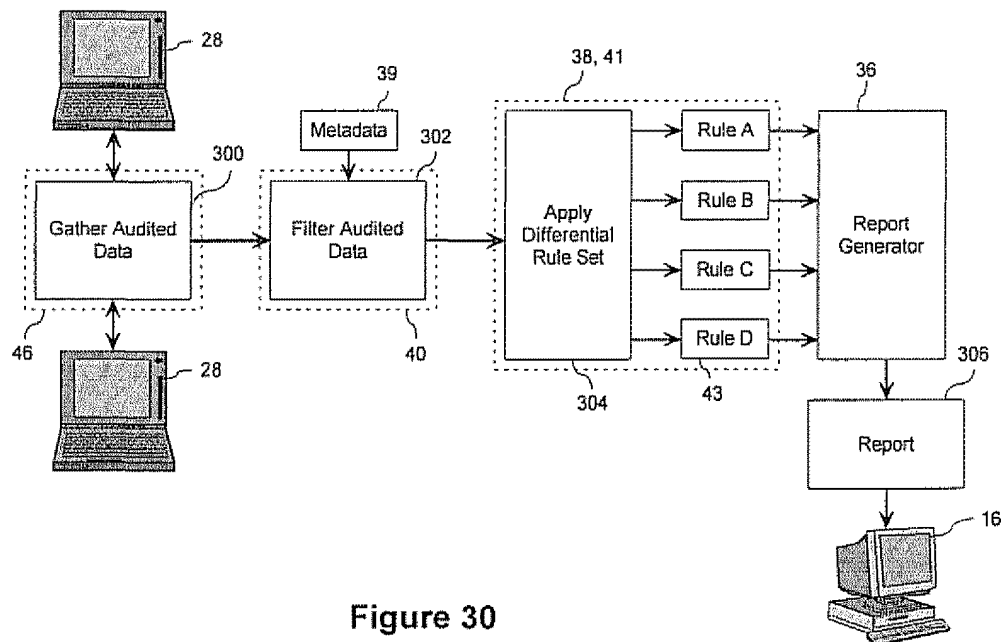


Figure 30

Differential Rule Definition

Rule Type	Rule Specifier	Source	Target	Weight	Mutex Flag	Match Flag	Supress Flag	Remediation Cost	Description
a	b	c	d	%	Y/N	e	f	\$	g

Figure 31

	Rule Type	Rule Specifier	Source	Target	Weight	Mutex Flag	Match Flag	Suppress Flags	Remediation Cost	Description
1	AliasQuery	OS			70%		OS		\$1,200	Different Operating Systems
2	AliasQuery	OS Version			30%		VER	OS	\$320	Different Operating System Versions
3	AliasQuery	Time Zone			20%				\$320	Systems are in Different Time Zones
4	AliasQuery	Total memory		<	5%				\$350	Target Has Less Memory
5	UrlQuery	cirba-cm/ServerInformation/OSPatchLevel			5%		PATC H	OS VER	\$160	Different Patch Levels
6	UrlQuery	cirba-pa/patchesTable#*			5% Y			OS VER	\$80	Patch Differences Between Systems
7	UrlQuery	cirba-cm/ServerInformation/OSKernelBits			10%			OS	\$80	Not Running Same Kernel Bits
8	UrlQuery	cirba-cm/KernelParametersTable#*	present	absent	5% Y			OS	\$80	Some Kernel Parameters are Absent
9	UrlQuery	cirba-cm/KernelParametersTable/Setting#SHMMAX		<	5%				\$80	Maximum Shared Memory Settings Differ
10	UrlQuery	cirba-si/appInvTableTable.appInvTable/version#oracle			5%				\$3,000	Different Version of Oracle
11	UrlQuery	cirba-si/appInvTableTable.appInvTable/version#oracle	9	8	4%				\$1,500	Different Version of Oracle
12	UrlQuery	cirba-si/appInvTableTable.appInvTable/version#apache			2%				\$80	Different Version of Apache
13	UrlQuery	win32-os/Win32_OperatingSystem/ServicePackMajorVersion			5%		PATC H	OS VER	\$160	Different Service Pack Levels
14	UrlQuery	win32-quickfix/win-hotfix#*	present	absent	5% Y			OS VER	\$80	Patch Differences Between Systems
15	UrlQuery	win32-os/Win32_StartupCommand#*			2% Y			OS	\$80	Startup Commands are Different
16	UrlQuery	win32-misc/Win32_Environment/VariableValue#Path			2%			OS	\$80	Global Path is Different
17	UrlQuery	win32-misc/Win32_Environment/VariableValue#Path <SYSTEM>			2%			OS	\$80	System Path is Different
18	UrlQuery	win32-os/Win32_Service#*	present	absent	5% Y			OS	\$160	Some Services not On Target
19	UrlQuery	win32-os/Win32_Service/Started			1% Y			OS	\$80	Target
20	UrlQuery	win32-application/win-products/McAfee VirusScan Enterprise	present	absent	5%			OS	\$200	McAfee Not On Target

Figure 32