Server Efficiency Rating Tool (SERT)

Mike G. Tricker
Microsoft Corporation
mike.tricker@microsoft.com

Klaus-Dieter Lange
Hewlett-Packard Company
klaus.lange@hp.com

Jeremy A. Arnold
IBM Corporation
arnoldje@us.ibm.com

Hansfried Block
Fujitsu Technology Solutions GmbH
hansfried.block@ts.fujitsu.com

Sanjay Sharma
Intel Corporation
sanjay.sharma@intel.com

ABSTRACT
The Server Efficiency Rating Tool (SERT) [1] has been developed by Standard Performance Evaluation Corporation (SPEC) [2] at the request of the US Environmental Protection Agency (EPA) [3], prompted by concerns that US datacenters consumed almost 3% of all energy in 2010. This poster-paper gives an overview of the SERT

Categories and Subject Descriptors
H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness)

General Terms
Design, Experimentation, Measurement, Performance, Reliability, Standardization

Keywords
SPEC, SERT, Rating Tool, Benchmark, Energy Efficiency, Power, Server, Storage, Datacenter, ENERGY STAR, Environmental Protection Agency, EPA

1. AN OVERVIEW OF THE SERT
The SERT is designed to be scalable to a maximum of 64 nodes (limited to a set of homogenous servers or blade servers) and to support multiple power analyzers and temperature sensors. The simplest SERT hardware measurement configuration requires four main hardware components: one **Power Analyzer**, one **Temperature sensor,** a **SUT** and the **Controller**.

The SERT is composed of several elements, starting with the test harness, named **Chauffeur,** which handles the logistical side of measuring and recording the power consumption and inlet temperature of the SUT. It also controls the software installed on both the SUT and Controller, communicating via the TCP/IP transport protocol.

Chauffeur communicates with the **Director**, which instructs the SUT to execute the **suite**, comprising a set of workloads. The **workload** comprises a set of worklets, which exercise the SUT while Chauffeur collects the power and temperature data. The **worklets** are the actual code designed to stress a specific system resource or resources, such as the CPU, memory or storage IO.

The temperature sensor must be placed no more than 50mm in front of (upwind of) the main airflow inlet of the SUT. The SERT will measure the inlet temperature of the SUT and marks the results "valid" only if the temperature measured is 20^{o}C or higher, in order to discourage the "gaming" of the test environment. A stable temperature value is not required during warm-up or measurement phases.

The power analyzer must be located between the AC Line Voltage Source and the SUT. Both are connected to the Controller via their device specific interfaces, as shown in Figure 1. Each analyzer and sensor interacts with its dedicated instance of the **SPEC PTDaemon,** which gathers their readings while the worklets are executed.

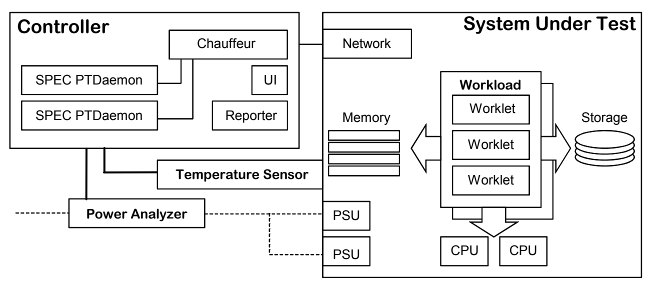

Figure 1. The SERT Overview

The **Reporter**, executed after all measurements phases are completed, compiles all of the environmental, power, and performance data for a complete test run into an easy to read report. The output format will be HTML, plain text, extensible markup language (XML), and comma-separated values (CSV); the HTML report includes a graphical visualization of the results.

2. WORKLET CANDIDATES
SERT worklets were designed under a set of public guidelines [1] to ensure consistent results across a broad spectrum of technologies. For example, each workload must automatically calibrate itself to report the maximum performance available in that specific hardware configuration, and must then be adjustable to target load levels from 100-0% of the maximum performance. Each worklet also needs to scale with the available hardware resources which the execution model deemed "important", e.g., a

ICPE'12, April 22–25, 2012, Boston, Massachusetts, USA.
ACM 978-1-4503-1202-8/12/04.

CPU worklet needs to scale with the number of processors, cores, hardware threads, and the clock frequency.

The SERT Design Document [1] offers a detailed breakdown of what each worklet does and how it works. Currently a total of 16 worklets is under evaluation, which can be summarized as:

CPU: Data compression, encryption/decryption, complex number arithmetic, matrix factorization, floating point array manipulation, sorting algorithm, string manipulation, and XML document validation;

Memory: XML document manipulation and validation using pre-computed and cached data lookup, and array manipulation with read/write operations across four major classes of data transformation;

Storage IO: Four individual transaction pairs combining sequential/random read/write and a mixed transaction, which combines all four;

Combined: The concept of CSSJ is derived from ssj2008, which simulated an on-line Transaction Processing workload in which customers order and pay for goods from warehouses that handle delivery and stock replenishment;

Active Idle: A steady state in which the server is ready to execute any worklet but is not actually doing so, leading to a measure of efficiency for a fully functional but otherwise idle state.

There are no worklets related to **Network IO**, which will be handled by a "configuration modifier" that simulates the steady state efficiency of a network device. After testing a variety of network interface cards (NICs) across a range of workloads it was observed that the power consumption of the actual devices approximated very closely to a constant (including in the case of NICs that perform offloading from the host processor), with CPU and memory power consumption being the biggest factors influencing overall system efficiency. Combined with the extensive set of external hardware required to effectively test network bandwidth and performance, it was agreed with the EPA that a modifier would be applied to simulate the network IO contribution to overall server efficiency.

3. SERT UI

Users may configure the SERT by manually editing the various configuration files or utilizing the newly designed SERT User Interface (SERT UI) in order to manage the behavior of each component.

During **Host Discovery,** the detailed hardware and software configuration of the SUT are gathered automatically by a remote task that uses the industry standard Common Information Model (CIM) definitions that are widely supported across hardware and OS platforms.

The SERT UI provides a graphical interface for gathering all the SUT hardware and software configuration data, configuring and running the SERT, as well as archiving the measured results and log files. It also supports the ability to save and re-import complete configurations to simplify repeated testing.

The default mode executes the entire SERT suite (all worklets) in sequence, with each worklet in a new instance of the local Java Virtual Machine (JVM), in order to create an EPA compliant test record. The SERT UI also offers an advanced research mode allowing the selective execution of a subset of workloads and worklets.

At the **Launch Test** (Figure 2) the progress of the entire suite can be observed, as well as the status of the currently executing worklet.

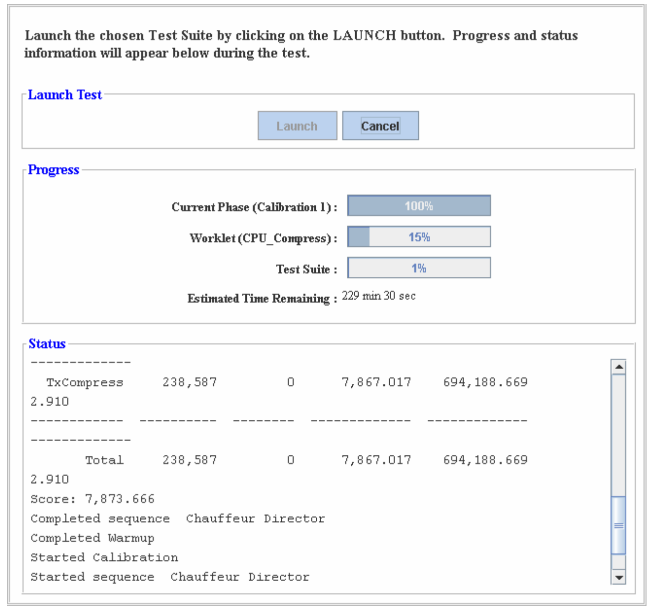

Figure 2. SERT UI: Suite Launch Menu

4. CONCLUSIONS

At the time of writing, the first Beta of the SERT has been delivered, with a second to follow soon, and a Release Candidate is targeted for the first half of 2012. The expectation is that the SERT will be released, together with Version 2 of the ENERGY STAR Computer Server program, in the first half of 2012.

5. ACKNOWLEDGMENTS

The authors want to acknowledge the additional members of the SPECpower Committee who have contributed to the design, development, testing, and overall success of the SERT: Paul Muehr, Van Smith, Greg Darnell, John Beckett, Karl Huppler, Nathan Totura and David Ott.

6. REFERENCES

[1] Server Efficiency Rating Tool public Design Document (latest version): http://www.spec.org/sert/docs/SERT-Design_Doc.pdf

[2] Standard Performance Evaluation Corporation home page: http://www.spec.org

[3] US EPA ENERGY STAR Enterprise Servers home page: http://www.energystar.gov/index.cfm?c=archives.enterprise_servers