Standard Performance Evaluation Corporation (SPEC)

Server Efficiency Rating Tool (SERT) Design Document 1.1.1

7001 Heritage Village Plaza, Suite 225
Gainesville, VA 20155,
USA

SPECpower Committee

Table of Contents

SVN Revision: 2000

SVN Date: 2016/01/13 02:26:29

1. Introduction

1.1. Summary

The Server Efficiency Rating Tool (SERT)™ was created by Standard Performance Evaluation Corporation (SPEC), the world's leading organization for benchmarking expertise, at the request of the US Environmental Protection Agency. It is intended to measure server energy efficiency, initially as part of the second generation of the US Environmental Protection Agency (EPA) ENERGY STAR for Computer Servers program. Designed to be simple to configure and use via a comprehensive graphical user interface, the SERT uses a set of synthetic worklets to test discrete system components such as memory and storage, providing detailed power consumption data at different load levels. Results are provided in both machine- and human-readable forms, enabling automatic submission to government-sponsored certification programs as well as both summary and detail reports for use by potential customers.

This public draft outlines the design of the SERT for public review; please visit http://www.spec.org/sert/docs/SERT-Design_Document.pdf for the latest updates.

1.2. About SPEC

The Standard Performance Evaluation Corporation (SPEC) was formed by the industry in 1988 to establish industry standards for measuring compute performance. SPEC has since become the largest and most influential benchmark consortium world-wide. Its mission is to ensure that the marketplace has a fair and useful set of metrics to analyze the newest generation of IT equipment.

The SPEC community has developed more than 30 industry-standard benchmarks for system performance evaluation in a variety of application areas and has provided thousands of benchmark licenses to companies, resource centers, and educational institutions globally. Organizations using these benchmarks have published more than 20,000 peer-reviewed performance reports on SPEC's website (http://www.spec.org/results.html).

SPEC has a long history of designing, developing, and releasing industry-standard computer system performance benchmarks in a range of industry segments, plus peer-reviewing the results of benchmark runs. Performance benchmarking and the necessary work to develop and release new benchmarks can lead to disagreements among participants. Therefore, SPEC has developed an operating philosophy and range of normative behaviors that encourage cooperation and fairness amongst diverse and competitive organizations.

The increasing demand for energy-efficient IT equipment has resulted in the need for power and performance benchmarks. In response, the SPEC community established SPECpower, an initiative to augment existing industry-standard benchmarks with a power/energy measurement. Leading engineers and scientists in the fields of benchmark development and energy efficiency made a commitment to tackle this task. The development of the first industry-standard benchmark that measures the power and performance characteristics of server-class compute equipment began on January 26 2006. In December of 2007, SPECpower_ssj2008 was released, which exercises the CPUs, caches, memory hierarchy, and the scalability of shared memory processors on multiple load-levels. The benchmark runs on a wide variety of operating systems and hardware architectures. In version 1.10, which was released on April 15 2009, SPEC augmented SPECpower_ssj2008 with multi-node support (e.g., blade-support). Several enhancements and code changes to all benchmark components, documentation updates, and run and reporting rules enhancements were included in version 1.11, released September 13, 2011 and version 1.12, released March 30, 2012.

1.2.1. SPEC's General Development Guidelines

SPEC's philosophy and standards of participation are the basis for the development of the SERT. The tool was been developed cooperatively by a committee representing diverse and competitive companies. The following points guide the committee in the development of a tool that will be useful and widely adopted by the industry:

- Decisions are reached by consensus. Motions require a qualified majority to carry.

- Decisions are based on reality. Experimental results carry more weight than opinions. Data and demonstration overrule assertion.
- Fair benchmarks allow competition among all industry participants in a transparent market.
- Tools and benchmarks should be architecture-neutral and portable.
- All who are willing to contribute may participate. Wide availability of results on the range of available solutions allows the end user to determine the appropriate IT equipment.

Similar guidelines have resulted in the success and wide use of SPEC benchmarks in the performance and power/performance industry and are essential to the success of the SERT.

1.2.2. SPEC Membership

SPEC membership is open to any interested company or entity. The members and associates of SPEC's Open System Group (OSG) are entitled to licensed copies of all released OSG benchmarks and tools as well as unlimited publication of results on SPEC's public website. An initiation fee and annual fees are due for members. Nonprofit organizations and educational institutions have a reduced annual fee structure. Further details on membership information can be found on http://www.spec.org/osg/joining.html or requested at info@spec.org. Also a current list of SPEC members can be found here: http://www.spec.org/spec/membership.html.

2. SERT's Scope and Goals

The current scope of the EPA's Energy Star Program for Servers includes servers with 1-4 processor sockets and blade technologies of similar scope. A design goal of the SERT is to accommodate these and larger technologies.

Among the issues involved with support of larger systems are the overall capacity of the system to complete work, and the ability to design a workload that scales with the inclusion of additional processors, memory, network interface cards, disk drives, etc. Different workload characteristics are required to demonstrate effectiveness for each of these components. Providing a workload that fairly represents their presence while not unfairly representing their absence is a challenge. These issues are more prevalent with larger systems that have more expansion capabilities than smaller servers.

For these areas where it is concluded that the tool does not adequately represent the value of a component compared to its power requirements, the tool will be designed to accommodate the inclusion of "Configuration Power/Performance Modifiers".

2.1. Overview Summary

The following table summarizes some of the design goals that the SERT will and will not provide:

IS	IS NOT
A Rating Tool for overall energy efficiency	A Benchmark nor a Capacity Planning Tool
A Measuring Tool for power, performance, and inlet-temperature	A Measuring Tool for Airflow, Air pressure, outlet-temperature
A general compute-environment measure	A specific application, JVM, or special purpose benchmark measure
Supporting AC-powered servers	Able to support DC-powered servers (See Section 4.7)
Used in single Operating System (OS) instance per server environments	Intended to stress virtualization hypervisor technology [1]
An Energy Efficiency Rating Tool	A Marketing Tool
Planned to be architecture- and OS- neutral	Implemented on architecture and/or OS environments where insufficient resource has been volunteered to accomplish development, testing, and support.

[1] SERT, as a first-order approximation tool, is not designed to evaluate the impacts of virtualized environments

2.2. SERT's Differences from Conventional Benchmarks

Performance and energy efficiency benchmarks tend to focus on capabilities of computer servers in specific business models or application areas. The SERT is focused on providing a first order of approximation of energy efficiency across a broad range of application environments.

- The absolute score is less relevant for the end user, because it will not reflect specific application capabilities.
- A rating tool that provides a pass-fail or a [Level 1/Level 2/Level 3] rating is a better fit for energy-efficiency regulatory programs than a typical benchmark result with multiple digits of precision in the metric.
- Marketing of the absolute scores will be disallowed in order to encourage more participation in the program.

Benchmarks tend to focus on optimal conditions, including tuning options to customize the configuration and software to the application of the benchmark business model. The need to achieve competitive benchmark results often causes significant investment in the benchmark process. The SERT is designed to be more economical and easier to use, requiring minimal equipment and skills through:

- Highly automated processes and leveraging existing SPEC methods
- Focused predetermined settings for the server
- Being free from super-tuning

Where a benchmark represents a fixed reference point, regulatory programs are designed to foster continuous improvement, with thresholds for success rising as the industry progresses. The SERT will be designed to match this paradigm, including:

- Quick adoption of new computing technologies
- Rapid turn-around for tool version updates

2.3. Sockets and Nodes

The SERT 1.0.0 is designed to be scalable and will be tested up to a maximum of 8 sockets and a maximum of 64 nodes (limited to a set of homogenous servers or blade servers). The server under test (SUT) may be a single stand-alone server or a multi-node set of servers. A multi-node SUT will consist of server nodes that cannot run independently of shared infrastructure such as a backplane, power-supplies, fans, or other elements. These shared infrastructure systems are commonly known as "blade servers" or "multi-node servers". Only identical servers are allowed in a multi-node SUT configuration.

2.4. Scaling

Since the server efficiency rating of a given server is the primary objective of the SERT, one of the main design goals for the tool is to be able to scale the performance on the system in proportion to the system configuration. As more components (processors, memory, and disk storage) are added to the server, the workloads should utilize the additional resources so that the resultant performance is higher when compared to the performance on the same server with a lesser configuration. Similarly, for a given server, when the components are upgraded with faster counterparts, the performance should scale accordingly. This is a very important aspect of the tool since adding and upgrading components typically increase the total power consumed by the server, which will affect the overall efficiency result of the server. Creating a tool that scales performance based on the number/speed of CPUs is most readily achievable – for the other components, the complexity of implementing such a tool increases substantially.

While the SERT will be designed to scale performance with additional hardware resources of the SUT, the SUT itself may not be able to sustain higher performance if there are performance bottlenecks in system components unrelated to the added hardware. In such cases, the addition of components to the SUT will normally result in

higher power consumption without a commensurate increase in performance. It is also possible that the workload mix that is defined for smaller systems will not scale well when examining larger systems.

2.5. Server Options and Expansion Capabilities

A server may have many optional features that are designed to increase the breadth of applications. These features not only require additional power, but also require more capacity in the power supplies and cooling system. Some of the SERT workload components will be designed to demonstrate the enhanced capabilities that these features provide. However, while the tool needs to credit these capabilities for the expanded workloads that they will accommodate, it cannot penalize efficient servers that are not designed with substantial expansion options. A balance must be struck between providing enhanced ratings for enhanced configurations and avoiding easy qualification of servers by simply adding features that may not be needed in all situations.

The SERT's goal is to avoid unnecessarily penalizing servers that are designed for low expandability, while crediting servers with greater expandability. For example, a configuration with four IO adapters in PCI slots may execute the workload more effectively than a configuration with only one such adapter. On the other hand, it may only run the workload as effectively as a configuration with two network adapters. Because the configuration with four adapters may run some real workloads more effectively than configurations with only two adapters, the regulatory program may elect to allow for some form of "Configuration Power/Performance Modifier" to provide credit for the power infrastructure needed to support the additional PCI slots.

The tool will be designed and tested to ensure that, should "Configuration Power/Performance Modifier" credits be included, the tool will accommodate them.

2.6. Redundancy

Many servers have redundancy built in for power supplies and cooling fans. Some servers include different levels of redundancy for memory, disk, and even processors. A design goal is to include accommodation for redundant components via Configuration Power/Performance Modifier, although no specific tests are planned for energy measurement under fault tolerant conditions when one of a redundant set of components is disabled.

2.7. Run Time

The right balance between high repeatability of the results, high sub-system coverage, and low resource allocation is desirable. The target run time is around five hours.

2.8. Customer Relevant Measurements

To provide results that are representative of a customer environment, the goal is to test systems in an "as-shipped" state. Unless otherwise specified by the Energy Efficiency Regulatory Program that is including the use of the SERT in its evaluation protocol, power management settings for the SUT will be assumed to be the default settings that are shipped with the SUT. Because of the variety of functions that are stressed by the various workloads within the SERT, it is anticipated that some performance tuning will be required to obtain consistent and customer-relevant results. No super-tuning will be allowed. SPEC will host a site with processor vendor-defined JVM tuning and flag options. Per workload (see section 3.3 for the SERT definition of a workload), only one set of JVM tuning flags per OS/JVM/micro-architecture configuration (e.g. Windows Server R2/Oracle HotSpot/AMD x86) may be used. For a valid SERT result, the JVM count is user defined. Other changes will cause the run to be non-compliant. The SERT will launch the JVM within the tool to restrict additional tuning.

2.9. Implementation Languages

The main body of code is in written in Java in order to lower the burden of cross-platform support. The framework is designed to accommodate other language implementations as well.

2.10. Load Levels

The SERT implements our concept of multiple load levels per worklet. The detail load levels can be found in worklet section (see section 8).

2.11. Platforms

The SERT 1.0.0 will be implemented for and is planned to be tested on the following platform/OS/JVM combinations (64 bit only), pending resources. In some cases, SPEC recommend the use of more than one JVM, where more than one JVM is generally available and selecting one may unfairly penalize a specific processor architecture or operating system.

HW Platform	OS	JVM
x86 - AMD	Windows Server 2008 R2	IBM J9 Java Runtime Environment Java HotSpot 64-Bit Server VM
x86 - AMD	Red Hat EL 6.2 SUSE SLES 11 SP2	IBM J9 Java Runtime Environment Java HotSpot 64-Bit Server VM
x86 - Intel	Windows Server 2008 R2	IBM J9 Java Runtime Environment Java HotSpot 64-Bit Server VM
x86 - Intel	Red Hat EL 6.2 SUSE SLES 11 SP2	IBM J9 Java Runtime Environment Java HotSpot 64-Bit Server VM
POWER - IBM	AIX	IBM J9 Java Runtime Environment
POWER - IBM	Red Hat EL 6.2 SUSE SLES 11 SP2	IBM J9 Java Runtime Environment
ARMv7 Cortex-A9 ARMv7 Cortex-A15	Ubuntu 12.04 Ubuntu 13.04	Oracle Java HotSpot VM

Note: OS refers to versions (service pack and patch levels) that are current at the SERT release.

Each row in the above table has a designated Environment Owner/Sponsor within SPEC for the purpose of defining architecture-specific parameters that will be used for valid SERT measurements. Individuals or corporations who are interested in expanding this list are encouraged to join SPEC (see section 1.2) and contribute to SPEC's OSG Power subcommittee as outlined in the next section.

2.12. Platform Addition Requirements

SPEC would welcome the addition of other Platform/OS/JVM combinations. Support for additional Platform/OS/JVM combinations requires active participation from the interested parties (Environment Sponsor). The inclusion of a JVM is dependent on an agreement from the JVM provider for unrestricted use of their JVM for the SERT. Companies dedicating additional resources to the SPECpower committee for development of the SERT would relax the schedule constraints.

Required Environment Sponsor commitment:

- Provide substantial initial testing and share test results with the SPECpower Committee, to ensure that the SERT is functioning as expected for the target environment. At the Environment Sponsor's discretion, the test results may be presented as measured or in normalized form.
- Develop a list of recommended tuning parameters that are comparable to parameters designated for other environments and that will be used for all official SERT measurements.

- Recommend documentation changes associated with the environment, including, but not limited to instructions for Storage Worklet cache use, setting huge pages, and descriptive wording, as appropriate
- Provide ongoing testing for any times when enhancements are made to the SERT or when there is a question as to the functionality of the SERT in the target environment.
- Provide technical support in the event that a SERT licensee has a problem running in the environment which cannot be resolved by the SPEC support staff or the current OSG Power Support liaison.
- Participate in SPECpower Committee follow-on work as they examine options for enhancing the SERT product and for generating metrics using the SERT.
- Ensure that the SERT can discover the necessary logical processor and memory information needed to run the tool. Highly desired, but not required: Test and assist with enhancement of discovery routines to automate discovery of much of the configuration for the new Environment.

Note that, in order to be effective in satisfying these commitments, the Environment Sponsor must be a member of SPEC and actively participate in the SPEC OSG Power subcommittee that manages the SERT. As discussed in Section 1.2, SPEC is a consortium with open membership (within the legal limits SPEC's not-for-profit incorporation).

Acceptance of some environments may require only a subset of the items listed in this section. For example, the new environment may be a new version of an operating system or a new distribution of an existing operating system or a hardware implementation that is related to one that is already accepted. These may only require a set of 5 tests for consistency, proof that discovery is working as intended, and one or two other spot checks. Environment owners must propose a limited acceptance process to the SPECpower Committee and be prepared to add tests to the plan if discussion warrants this.

2.12.1. Test

Demonstrate that the SERT functions as expected and delivers realistic results for a specific configuration. The target configuration used for other environments has been a two-socket server with multiple disk drives and a moderate (32-128 GB) amount of memory. In case a SERT worklet does not function as expected, the environment sponsor works with the SERT development team to resolve the issue.

Demonstrate that the SERT scales as expected by running:

- On 1-socket, 2-socket, 4-socket, and 8-socket configurations (as supported by the environment).
- On at least 4GB, 8GB, 32GB, 128GB, and 1024GB of memory (as supported by the environment).
- With a single disk, 2 disks, 4 disks, and 8 disks (as supported by the environment).
- With processor cores of different frequencies.
- With memory of different speeds.
- With storage of different speeds.
- With five measurements of the full SERT suite on a single configuration to ensure that at least 4 of them complete without invalid messages for coefficient of variation (CV) or throughput targets, and that run-to-run variation also meets CV requirements.
- If the environment is expected to participate in multi-node measurements, it is highly recommended that a set of measurements be made in a multi-node environment, to ensure that measurements complete correctly and that cross-node CVs are satisfactory.

It is not required to measure every possible combination in the above list, but rather to cover the list in some combination. The following table is an example of a possible test matrix.

Environment owner/sponsors should be prepared to share the results of these tests on a SPEC-Confidential basis with the members of the SPEC OSG Power subcommittee. Results may be shared using actual performance values, or may be normalized to relative values that are consistent across the scope of measurements submitted for the acceptance proposal of the environment. However, other data may need to

be shared as a part of the owner/sponsor's ongoing commitment to participate in future development and testing.

	System A – 1 socket 2.5 GHz	System B – 2 sockets 2.5 GHz	System C – 4 sockets 3.5 GHz	System D – 8 sockets 2.2 GHz
Full SERT Suite	Measure on 4 GB and 8 GB memory with one storage device	Measure five times on 128 GB memory with 4 storage devices	Measure on 32 GB and 1024 GB memory with one storage device	Measure on 128 GB memory with one storage device
Memory tests		Repeat full suite with 128 GB memory at different speed than prior runs		
Storage tests		Measure storage workload on one configuration with 1, 2, 4, 8 similar disks	Measure storage workload with 2 HDDs and 2 SSDs	
Processor Speeds			Repeat Full Suite with processor down-clocked to 2.5 GHz	
Second JVM	Repeat suite with 8 GB of memory and one storage device	Measure five times on 128 GB memory with 4 storage devices	Repeat Full Suite at one of the prior processor speeds	

Measurements in this matrix should be done using the tuning parameters identified for the environment. In particular, the 5 repeated measures require the use of final tuning parameters. Note that the above matrix is an example and is not the only allowable arrangement of tests.

2.12.2. Tuning Parameters

- Ensure that affinity tuning within SERT is operating properly for the target environment.
- Identify Java tuning parameters needed to obtain consistent measurement information that is comparable with other environments supported by the SERT.
- Identify requirements for setting huge pages for the environment, if appropriate
- Identify requirements for deactivating write cache for storage devices

2.12.3. Documentation

- Review existing documents, particularly this Design Document, the Users Guide and the Run and Reporting Rules
- Propose changes needed for each to accommodate the new environment

2.12.4. Ongoing Testing

- As minor changes are made to the SERT, primarily as a result of problem fixes, but perhaps with additional function, conduct a subset of the initial testing to ensure that the target environment has not been impacted.
- For significant changes to the SERT, such as the addition of new worklets, metric changes and alterations to the key functions of existing worklets, more extensive testing is required. This is particularly true if the changes make the operation of the SERT more restrictive in some way than prior versions.

2.12.5. Technical Support

- Participate in the overall SERT technical support process. There will be questions that require the technical expertise of the SERT development team and the Environment Sponsor.

2.12.6. Future Enhancements

- Work with the development team to enhance the SERT to include better discovery of the configuration tested in the target environment.
- Participate in the decision-making process regarding other possible enhancements, such as new worklets or modifications to existing worklets.
- Participate in the process for development and recommendation of future metrics associated with SERT-based measurements
- Test prospective changes to the SERT in the target environment to ensure that they function and perform as expected.

2.12.7. Discovery

- Except for those components necessary to run the tool (logical processors, memory capacity), configuration discovery is not required to be complete, but can greatly enhance the ability to properly document a test result using the SERT.
- Work with the SERT development team to test and enhance discovery routines for the new environment.

2.13. Incremental Changes to Accepted Environments

In general, update enhancements to environments that already supported by an environment sponsor and accepted by the SPEC OSG Power committee are accepted with the existing parameters as specified by the environment sponsor. For example, an update to a processor offering that increases frequency or the number of cores per socket, but remains in the same named family as previously tested offerings is assumed to be supported using the same JVM and other tuning parameters as were previously specified. Similarly, update revisions to a JVM, operating system, or firmware are assumed to be accepted when the run parameters of the overall environment are not changed.

If update enhancements to any of the environment components cause the environment sponsor to alter the JVM and other tuning parameters from those previously specified, the updated environment becomes a new environment that requires at least some testing and related support commitments, as defined in Clause 2.12.

2.13.1. Acceptance of Unannounced Products

Testing for unannounced processors, JVMs, or versions of software may be presented on a SPEC-Confidential basis to the OSG Power committee to facilitate rapid deployment of results from the SERT when the product is announced. All requirements documented in Clause 2.12 still hold. An official acceptance statement will not be made until the product announcement becomes public.

2.13.2. Acceptance for Different Revisions of the SERT

Acceptance of an environment and associated environmental parameters for one revision of the SERT does not guarantee acceptance for either future or prior revisions. Additional testing for each revision may be required, as discussed in Clause 2.12.4. The level of testing required will be determined by the SPEC OSG Power committee, depending on the nature of the changes in the revision.

3. The SERT Architecture

3.1. Environment Overview

The SERT is composed of multiple software components and shares design philosophies and elements from SPECpower_ssj2008 in its overall architecture.

For the most basic SERT measurement setup the following is required:

- System under Test (HW) – the actual system for which the measurements are being taken. The controller and SUT are connected to each other via a TCP/IP connection.
- Controller (HW, e.g. server, PC, laptop) – the system to which the power and temperature sensor are connected. Multi-Controller environments are not supported.
- Power analyzer (HW) – connected to the Controller and used to measure the power consumption of the SUT. Multiple Power Analyzer environments are supported.
- Temperature sensor (HW) – connected to the Controller and used to measure the ambient temperature where the SUT is located.
- The SPEC PTDaemon (SW) – connects to the power analyzer and temperature sensor and gathers their readings while the suite of workloads executes. All instances of the PTDaemon (each must be the same version) must run on the Controller, each listening on a different port.
- The Reporter (SW) – summarizes the environmental, power, and performance data after a run is complete and compiles it into an easy to read format.
- The GUI (SW) – eases configuring and executing the SERT suite.

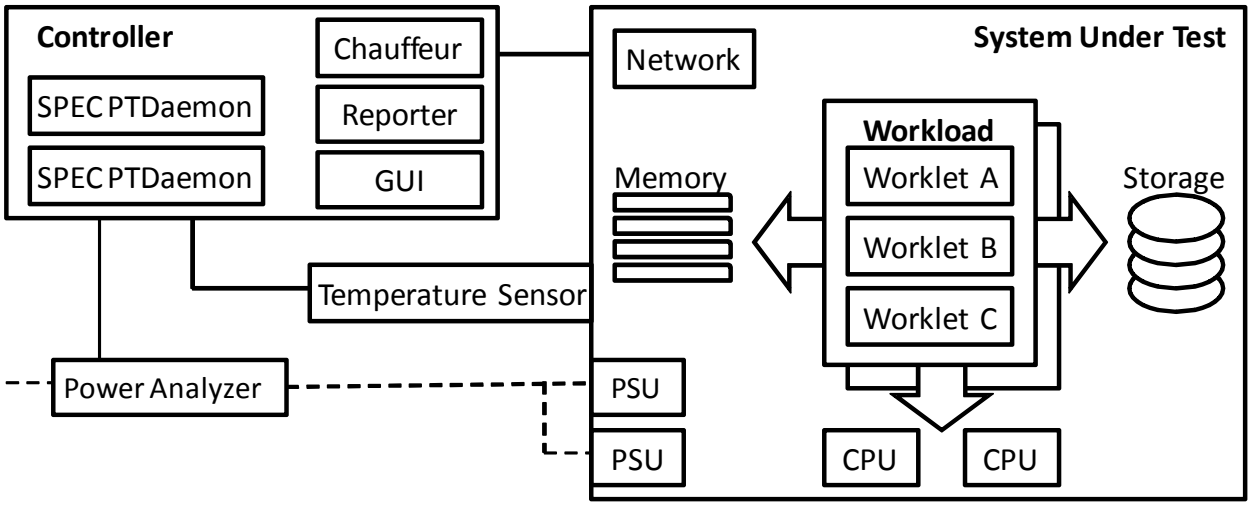


Figure 1: SERT System Diagram

Chauffeur (SERT test harness) handles the logistical side of measuring and recording power data along with controlling the software installed on the SUT and Controller. It is responsible for the initialization of the following JVMs:

- The Client JVM – executes the workload.
- The Host JVM – starts all Client JVMs on one SUT.
- The Director – instructs the Host JVM to start the workload.

The basic system overview in Figure 2: SERT Components shows these components in relationship to each other.

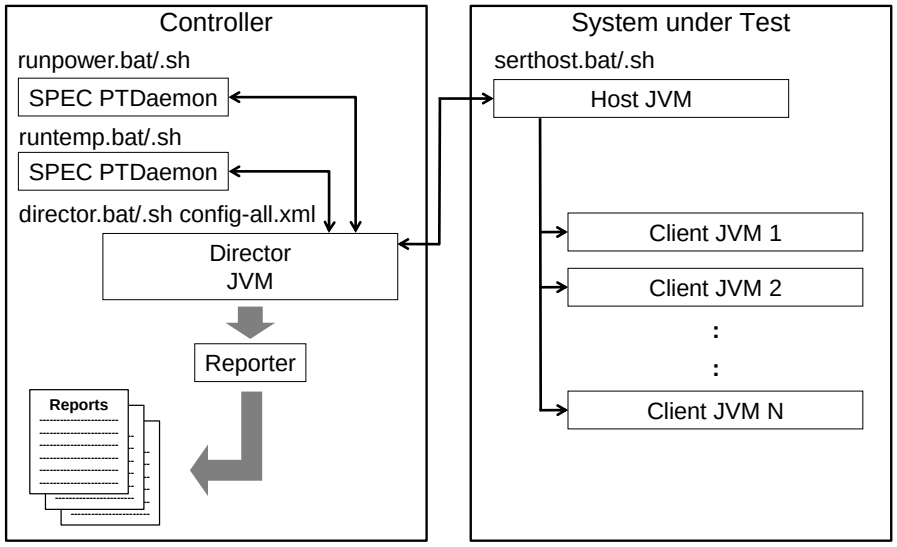


Figure 2: SERT Components

3.2. SERT Suite

The SERT Suite is composed of several logical elements including:

- User
  - A User is a representation of an external agent that can initiate work (e.g. a human being).
  - Each User may maintain identifying information (e.g. each User represents a Warehouse).
  - Each User may maintain temporary state information that persists from one transaction to another.
- Transaction
  - A Transaction receives a User and transaction-specific Input as parameters.
  - It produces some result.
  - Some transactions may be able to verify their results — this could be used for a small portion of transactions for auditing purposes.
- Worklet (Compress, CryptoAES, LU, SHA256, SOR, SORT, XMLValidate, Flood, Capacity, SSJ, Idle)
  - A worklet defines a set of transactions that can be executed by a particular type of User.
- Scenario
  - A Scenario is a group of one or more Transactions which are executed in sequence by a particular User.
  - When a worklet is running a load level, each scenario is scheduled to being execution at a specific time. If that time is in the future, the User will execute no transactions until the scheduled time arrives.
  - Each transaction in a scenario is submitted to a JVM-wide thread pool for execution.
- Interval
  - Each Interval includes pre-measurement, recording, and post-measurement periods.
  - Transactions are executed throughout the entire interval, but results are only recorded during the recording period.
  - Power consumption is recorded only during the recording period.
- Sequence
  - A Sequence of related intervals.
  - The intervals in a sequence may be executed identically (e.g. during calibration).
  - The intervals in a sequence may execute at different load levels (e.g. 100%, 80%, 60%, 40%, and 20% of the calibrated throughput).

    - The intervals in a sequence may run with different configuration parameters (e.g. Capacity2 using a data store size of 4GB, 8GB, etc.).
- Phase
    - A phase of execution: warm-up, calibration, or measurement
    - Each phase consists of a sequence of intervals.
    - Chauffeur supports multiple sequences in the measurement phase, but SERT always runs a single sequence in each phase.
- Workload (CPU, Memory, Storage, Hybrid, Idle)
    - A workload is a group of worklets designed to stress some aspect of the SUT.
    - The worklets in each workload run one at a time in a defined order.

3.3. Workload

The design goal for the SERT suite is to include all major aspects of server architecture, thus avoiding any preference for specific architectural features which might make a server look good under one workload and show disadvantages with another workload. The SERT workload will take advantage of different server capabilities by using various load patterns, which are intended to stress all major components of a server uniformly.

The existing SPEC benchmarks are mainly based on tailored versions of real world applications representing a typical workload for one application area or a synthetic workload derived from the analysis of existing server implementations. These benchmarks are suitable to evaluate different sub-areas of the overall server performance or efficiency if power measurements are included. They are not designed to give a representative assessment of the overall server performance or efficiency.

It is highly unlikely that a single workload can be designed which achieves the goals outlined above, especially given the time constraints of the schedule targeted for the anticipated regulatory program. Therefore, the SERT workload will consist of several different worklets, each stressing specific capabilities of a server. This approach furthermore supports generating individual efficiency scores for the server components besides the overall system score.

3.4. Worklet Execution Phases

The SERT test suite consists of several workloads which are designed to stress the various components of the system under test (SUT): CPU, memory, and storage. Each workload includes one or more worklets which execute specific code sequences to focus on one of these system components. The overall design structure of the SERT is shown in Figure 3: SERT Suite Components.

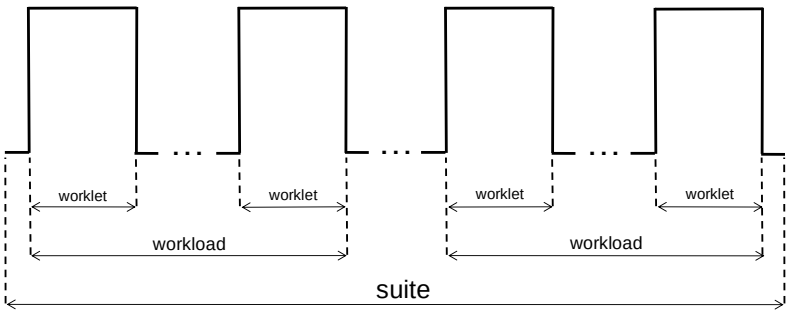
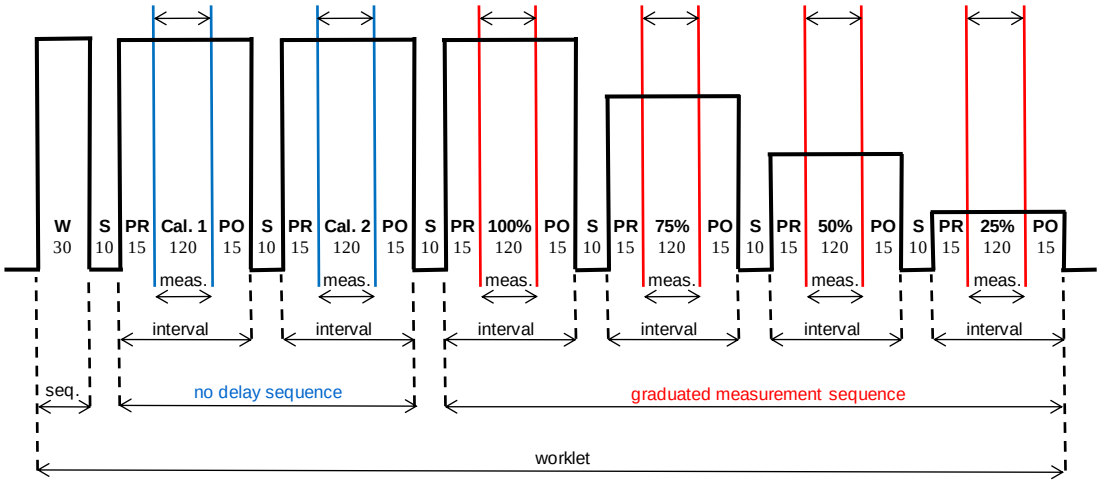


Figure 3: SERT Suite Components

Worklets are run one at a time. Most worklets consist of a warm-up phase, a calibration phase, and a measurement phase; some worklets do not include warm-up or calibration. Each of these phases consists of a sequence of one or more intervals. Each interval includes a pre-measurement, recording, and post-measurement period. Score calculations are based on the number of transactions and the power consumed during the recording period only. All of the worklets other than Flood2 use fixed time periods for the pre-measurement, recording, and

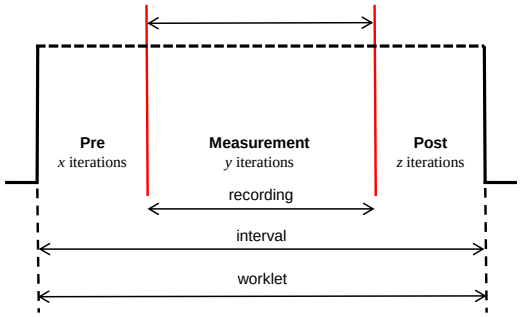
post-measurement periods. The Flood2 worklet runs a fixed number of iterations for each period. The pre-measurement period allows the worklet to reach steady state before recording begins. The pre- and post-measurement periods also ensure that in multi-client and multi-host runs, all clients are executing transactions concurrently during the recording period, even if there are slight discrepancies in the time that the recording period begins and ends. Intervals are separated by short delays to summarize results and allow new power analyzer ranges to take effect.

Most worklets (with the exception of the memory and idle worklets) run as shown in Figure 4: Graduated Measurement Execution. Each of these worklets begins with a warm-up phase which runs transactions for a short period of time to allow the worklet execution to stabilize. Throughput during the warm-up interval may vary due to factors such as JIT compilation, memory allocation, and garbage collection, and system caches adjusting to the new load. After warm-up, a sequence of calibration intervals is run to identify the maximum rate that transactions can be executed for this worklet. Two calibration intervals are used, and the calibrated throughput is the average for these two intervals. After the calibrated throughput is found, SERT runs a series of intervals where the worklet is throttled back to run at some percentage of the maximum throughput. The series of load levels varies by worklet, as defined in Section 8.



W = Warm-up (30 sec)
S = Sleep (10 sec)
PR = Pre measurement (15 sec)
PO = Post measurement (15 sec)
Cal. N = Calibration Interval N (120 sec)
nnn% = Measurement Interval (120 Sec)

Figure 4: Graduated Measurement Execution



Post = Post measurement (z iterations) (optional)
Pre = Pre measurement (x iterations) (optional)
Measurement = Measurement (y iterations)
Worklet execution time = not fixed, depending on system capacity

Figure 5: Fixed Iteration Execution

4. Power and Temperature Measurements

The SERT provides the ability to automatically gather measurement data from accepted power analyzers and temperature sensors and integrate that data into the SERT result. It will be required that the analyzers and sensors must be supported by the measurement framework, and must be compliant with the specifications in this section.

4.1. Environmental Conditions

Power measurements need to be taken in an environment representative of the majority of usage environments. The intent is to discourage extreme environments that may artificially impact power consumption or performance of the server, before and during the SERT run.

The following environmental conditions need to be met:

- Ambient temperature lower limit: 20°C
- Ambient temperature upper limit: within documented operating specification of the SUT
- Elevation and Humidity: within documented operating specification of the SUT
- Overtly directing air flow in the vicinity of the measured equipment in a way that would be inconsistent with normal data center practices is not allowed.

4.2. Temperature Sensor Specifications

Temperature must be measured no more than 50mm in front of (upwind of) the main airflow inlet of the SUT. To ensure comparability and repeatability of temperature measurements, SPEC requires the following attributes for the temperature measurement device used during the SERT run:

- Logging - The sensor must have an interface that allows its measurements to be read by the SERT harness. The reading rate supported by the sensor must be at least four samples per minute.
- Accuracy - Measurements must be reported by the sensor with an overall accuracy of +/- 0.5 degrees Celsius or better for the ranges measured during the SERT run.

4.3. Power Analyzer Requirements

To ensure comparability and repeatability of power measurements, the following attributes for the power measurement device are required for the SERT. Please note that a power analyzer may meet these requirements when used in some power ranges, but not in others, due to the dynamic nature of power analyzer Accuracy and Crest Factor. The usage of power analyzer's auto-ranging function is not permitted.

- Measurements - The analyzer must report true RMS power (watts) and at least two of the following measurement units: voltage, amperes, and power factor.
- Accuracy - Measurements must be reported by the analyzer with an overall uncertainty of 1% or better for the ranges measured during the benchmark run. Overall uncertainty means the sum of all specified analyzer uncertainties for the measurements made during the benchmark run.
- Calibration - The analyzer must be able to be calibrated by a standard traceable to NIST (U.S.A.) (http://nist.gov) or a counterpart national metrology institute in other countries. The analyzer must have been calibrated within the past year.
- Crest Factor - The analyzer must provide a current crest factor of a minimum value of 3. For analyzers which do not specify the crest factor, the analyzer must be capable of measuring an amperage spike of at least three times the maximum amperage measured during any one-second sample of the benchmark run.
- Logging - The analyzer must have an interface that allows its measurements to be read by the SPEC PTDaemon. The reading rate supported by the analyzer must be at least one set of measurements per second, where set is defined as watts and at least two of the following readings: voltage, amperes, and power factor. The data averaging interval of the analyzer must be either one (preferred) or two times the reading interval.

"Data averaging interval" is defined as the time period over which all samples captured by the high-speed sampling electronics of the analyzer are averaged to provide the measurement set.

Examples:

An analyzer with a vendor-specified accuracy of +/- 0.5% of reading +/- 4 digits, used in a test with a maximum power value of 200W, would have "overall" accuracy of (((0.5%*200W)+0.4W)=1.4W/200W) or 0.7% at 200W.

An analyzer with a wattage range 20-400W, with a vendor-specified accuracy of +/- 0.25% of range +/- 4 digits, used in a test with a maximum power value of 200W, would have "overall" accuracy of (((0.25%*400W)+0.4W)=1.4W/200W) or 0.7% at 200W.

4.4. SPEC PTDaemon

SPEC PTDaemon (also known as power/temperature daemon, PTD, or ptd) is used by the SERT to offload the work of controlling a power analyzer or temperature sensor during measurement intervals to a system other than the SUT. It hides the details of different power analyzer interface protocols and behaviors from the SERT software, presenting a common TCP-IP-based interface that can be readily integrated into different benchmark harnesses.

The SERT harness connects to PTDaemon by opening a TCP port and using the simple commands. For larger configurations, multiple IP/port combinations can be used to control multiple devices.

PTDaemon can connect to multiple analyzer and sensor types, via protocols and interfaces specific to each device type. The device type is specified by a parameter passed locally on the command line on initial invocation of the daemon.

The communication protocol between the SUT and PTDaemon does not change regardless of device type. This allows the SERT to be developed independently of the device types to be supported.

4.5. Supported and Compliant Devices

The SERT will utilize SPEC's accepted measurement devices list and SPEC PTDaemon update process. See Device List (http://www.spec.org/power_ssj2008/docs/device-list.html) for a list of currently supported (by the SPEC PTDaemon) and compliant (in specifications) power analyzers and temperature sensors.

The process to add software support for a power analyzer or temperature sensor to the infrastructure can be found on the Power Analyzer Acceptance Process page (http://www.spec.org/power/docs/SPEC-Power_Analyzer_Acceptance_Process.pdf).

4.6. Power Analyzer Setup

The power analyzer must be located between the AC Line Voltage Source and the SUT. No other active components are allowed between the AC Line Voltage Source and the SUT. Power analyzer configuration settings that are set by the SPEC PTDaemon must not be manually overridden.

4.6.1. 3-Phase Measurements

A 3-phase analyzer must be used for 3-phase measurements.

4.7. DC Line Voltage

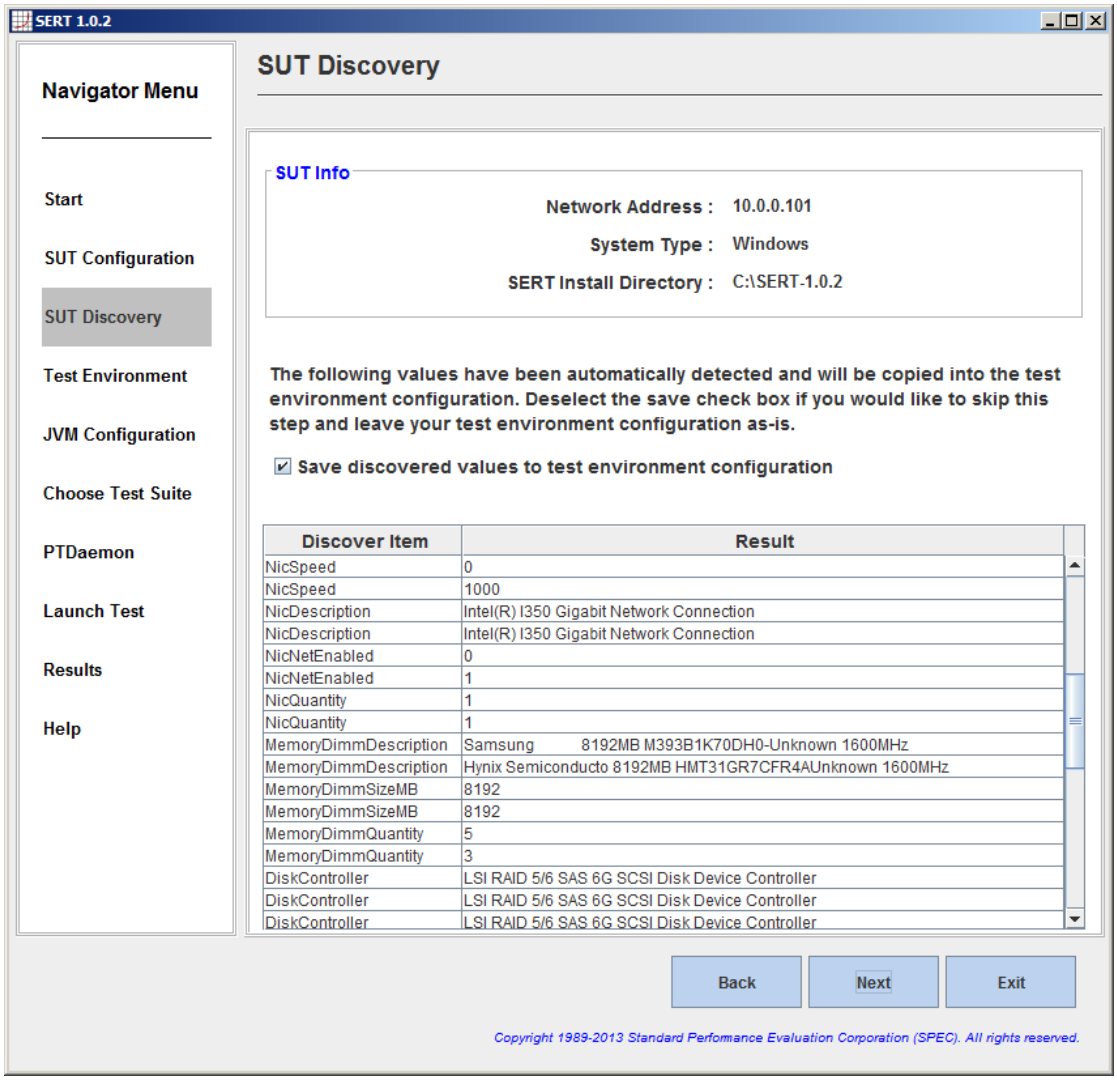
The SPECpower Committee is in favor of including DC support if new resources from companies whose focus is DC computing become available to the SPECpower Committee to address the development and support opportunity. Additionally, directly comparing servers powered by AC against servers powered by DC is not appropriate, since the AC-DC conversion losses are not included in DC-powered server. Therefore, the SPECpower Committee recommends creating a separate category for DC-powered servers.

5. Graphical User Interface

A graphical user interface (GUI) is provided to facilitate configuration and setup of test runs, to allow real-time monitoring of test runs, and to review the results. The SERT GUI leads the user through the steps of configuring the SUT and initiating a test sequence. These steps include detecting and customizing the hardware and software configurations, selecting the test type, and displaying the results.

The SERT GUI includes several features to enable SERT testing with minimal training, thereby enhancing the accuracy of results. Some of the features include:

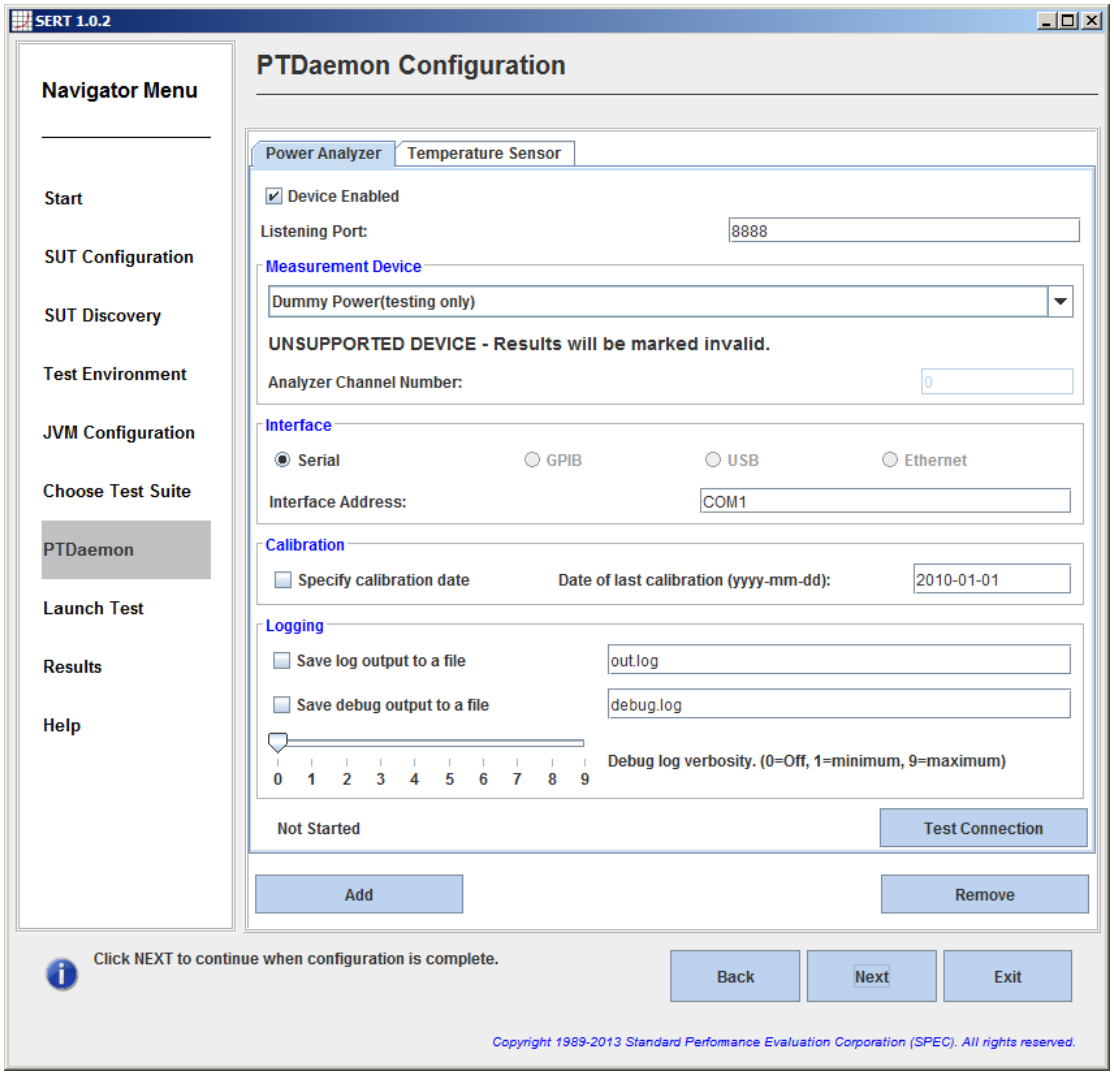
- Easy Navigation with Tabbed Screens via “Navigation Menu” and/or “Back/Next” buttons
- In-line usage guidance and help
- SUT Configuration Discovery (Detect function) automatically detects configuration information and populates many fields with SUT and Controller hardware and software details.



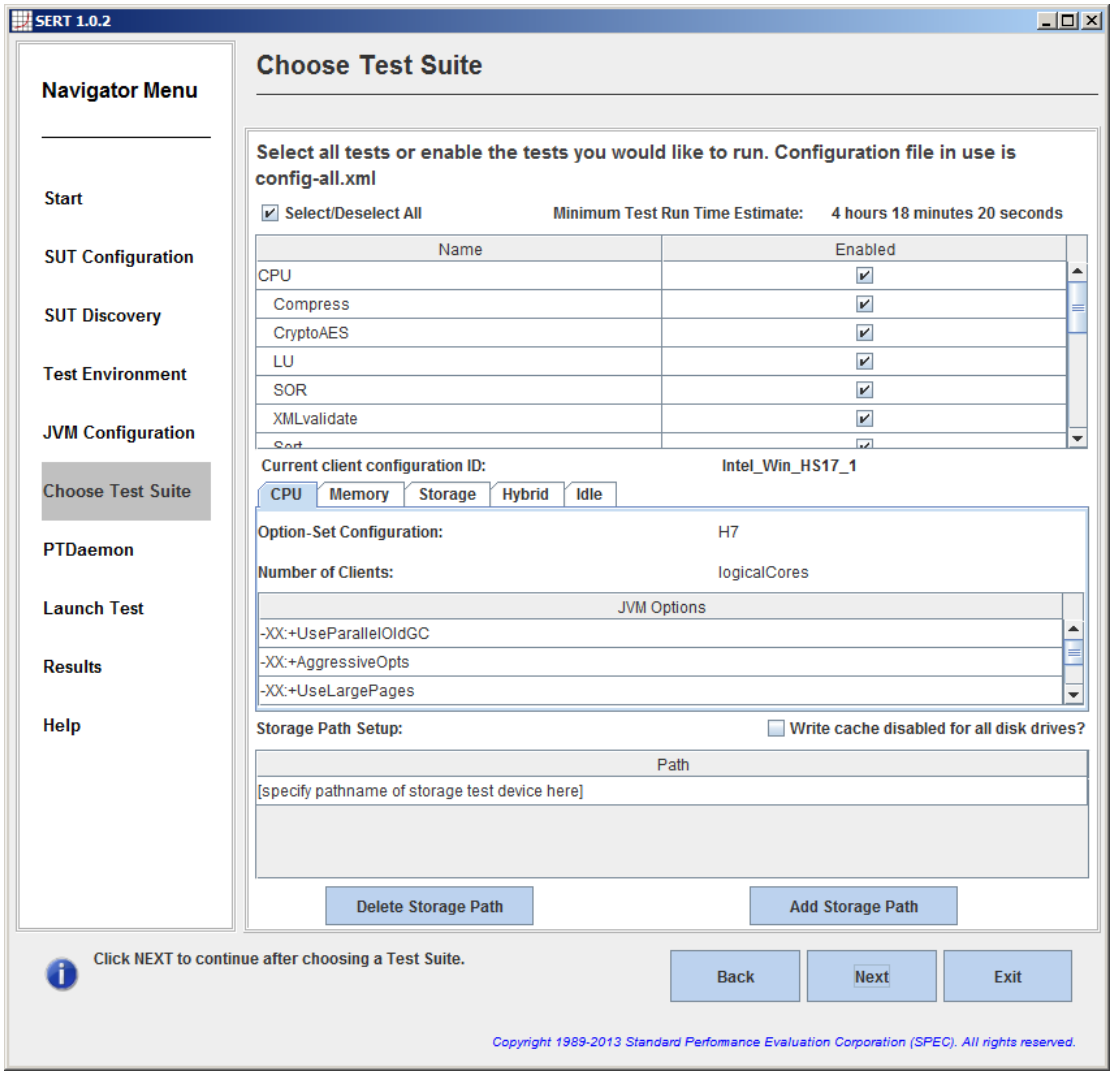
- The GUI displays test environment information, allowing the modification and saving of updated fields.
  - For use in reports: e.g. Vendor Information, Platform Configuration, Run-Time parameters, etc.

- Test Run information files can be stored for subsequent reuse, thus saving time when testing multiple identical platforms. For running on multiple platforms that are similar to each other, the saved information for one platform can be loaded and quickly modified for the other platform.

- The GUI allows setting up the PTDaemon Configuration, and testing of the connection to a Power Analyzer or Temperature Sensor (as shown on the tabs in the screenshot below).
  - Support for all PTDaemon supported Power Analyzers and Temperature Sensors
  - Access to all PTDaemon debugging and configuration options
  - Establish and test a connection with PTDaemon supported devices
  - Progress, warnings, and errors are displayed in Launch Test panel output



- The GUI provides the capability to set up the Test Suite and Configuration.
  - Selects entire test suite for an official test run
  - Selective execution of a subset of workloads and worklets for a quick validation of test setup or experimentation
  - Configures JVM options and number of clients
  - Sets up one or multiple storage path(s) pointing to the physical storage device(s)



- Displays the Test Execution and Progress
  - Validates storage path parameters and storage device space remaining
  - Start measurements
  - Displays progress, warnings, and errors
  - PTDaemon messages displayed in separate tabs for clarity
- Displays Results and provides access to the final report
  - Save, View, and Recall archived reports

6. Metric/Score, Reporting, Logging

6.1. Metric/Score

While the SERT is not intended to be a benchmark, nevertheless as a rating tool, it must produce a metric or score(s) indicative of the efficiency of the SUT.

Since different architectures perform differently on different workloads, the SERT is composed of several discreet worklets to further ensure architecture neutrality. Each worklet will produce a measure representing the performance and power consumption achieved by the SUT.

In order to get performance values in the same order of magnitude from all worklets the individual worklet performance scores will be normalized relative to an arbitrarily selected baseline in the 1.0.0 release of the SERT. No summation of metrics across multiple worklets is planned. Please see section 9.1.1 (Scoring and Metric).

Here is the description of the reference platform used to define the baseline for result normalization:

- 1-socket tower server
- 1 x Intel Celeron CPU G1101 @ 2.26GHz
- 4 x 2GB PC3L-10600E DIMMs
- 1 x SATA 80GB 7200rpm 3.5” HDD connected to an onboard SATA controller

The baseline results have been derived from 6 SERT runs on this reference platform:

- 3 runs under RedHat Enterprise Linux 6.2 (2.6.32-220.el6.x86_64) plus
  Oracle Java(TM) SE Runtime Environment (build 1.7.0_07-b10)
- 3 runs under Microsoft Windows Server 2008 R2 Enterprise (6.1.7600 (64-bit)) plus
  Oracle Java(TM) SE Runtime Environment (build 1.7.0_07-b10)

The reference score for each worklet is defined as the average worklet performance score over all 6 SERT runs on the configurations described above.

The result that is produced by the SERT is separate from the rating of the different energy efficiency regulatory programs. The SUT might be placed in different categories of the energy efficiency regulatory programs.

6.2. Reporting and Output Files

The SERT produces multiple reports, and includes code that will ensure the authenticity of the reports.

6.2.1. Report Formats

In order to reduce the effort of displaying and/or storing the desired information, the primary report is generated in the XML format. This report contains all the information regarding the SERT run including all hardware and software configuration of the Controller, the SUT, and the SERT workloads. It includes all pertinent information about the worklets like JVM affinity, options, and other launch settings, along with the resulting performance, power and efficiency results.

In addition, multiple human readable versions of the report are also generated, two HTML and two plain text. These reports are designed to be viewed on a screen with minimum width of 1280 pixels and contain a subset of data that is derived from the primary XML report. The names and brief descriptions of the reports are given below:

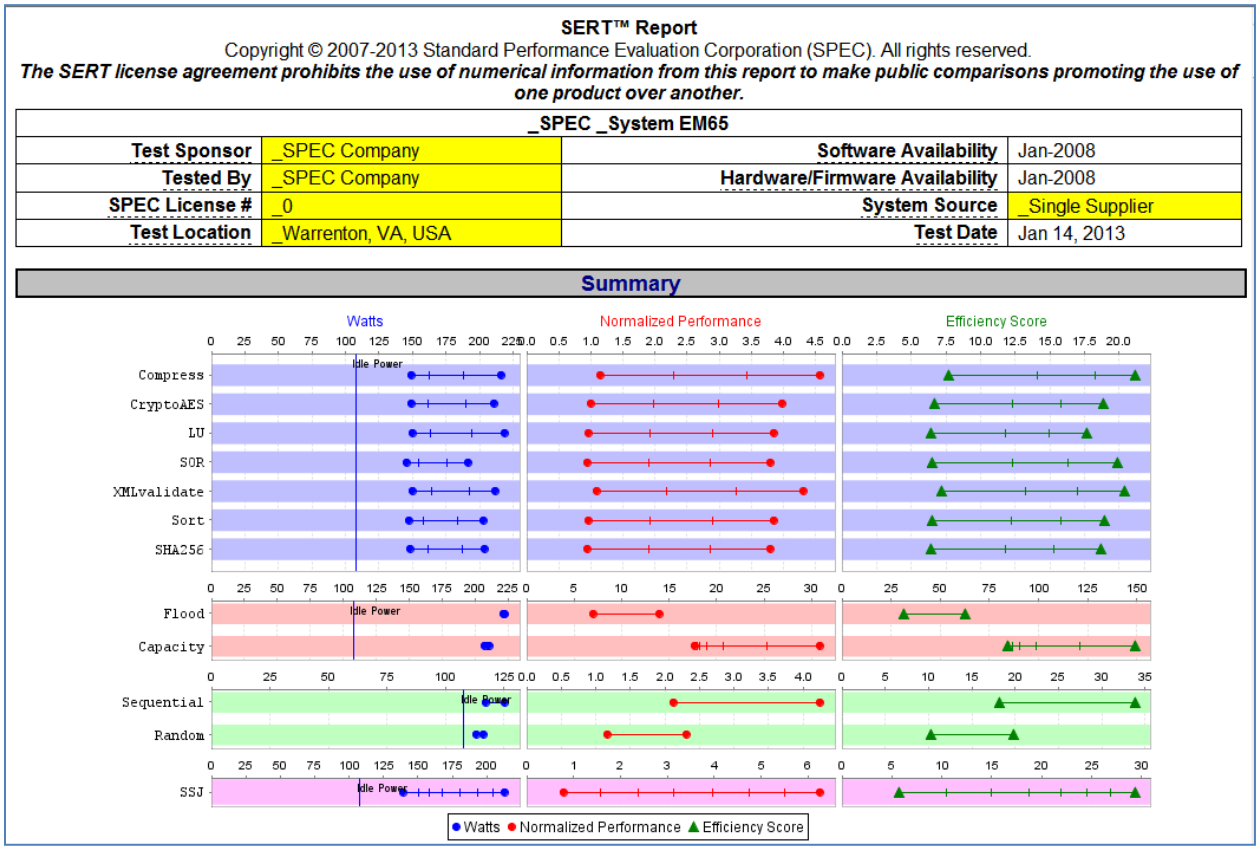
1. The first HTML report, named “**results.html**”, contains all pertinent hardware and software information about the test environment, including the SUT, and the resulting aggregate performance, power consumed, and efficiency score achieved by the different worklets.
2. The text report, named “**results.txt**” contains the same information as the results.html file above, only it is in a plain text format.

3. The third report is in HTML format as well, and is named "**results-details.html**" and contains all the information present in the results.html file above, along with detailed breakdown of the performance and power at each load level for all the worklets. In addition, for each worklet, the run-time parameters like number of CPUs, threads, and command line options are also captured.
4. The fourth report is in TXT format as well, and is named "**results-details.txt**" and contains all the information present in the results.txt file above, along with detailed breakdown of the performance and power at each load level for all the worklets. In addition, for each worklet, the run-time parameters like number of CPUs, threads, and command line options are also captured.

6.2.2. Sample HTML Report Output

A partial image of a sample "results.html" report is shown below. This snapshot is from the report generated using version 1.0.0 of the SERT (Data shown in this screen shot is from an arbitrary system and meant as an example only).

The default parameters in the test-environmentl.xml file are defined with values that are intentionally incorrect. To highlight this, default text parameters are defined with a leading underscore. The reporter recognizes this and highlights these fields with a yellow background, except for the system name in the headline of the general information table. This highlighting is not possible for numeric parameters.



The summary results chart graphically displays the power, performance and efficiency scores for the different worklets. Each worklet is presented on a separate row beginning with the worklet name on the left. The lines are printed with distinct background colors for the different workloads.

The summary result chart is divided in three sections.

- **Watts**
  The leftmost section displays the range of measured power consumption for all load levels. The Watts values are given on vertical scales at the top of each workload section.
  The power range is represented by a blue line with a dot at the left for minimum power corresponding to the lowest load level and a dot at the right for maximum power corresponding to the highest load level. The intermediate load levels, if defined, are represented by small vertical bars. The vertical blue line across all worklets within each workload, named "Idle Power" represents the power consumption of the idle worklet. It does not correspond to any load.

- **Normalized Performance**
  The middle section displays the range of normalized performance values for all load levels. The normalized performance values are given on vertical scales at the top of each workload section.
  These values are calculated separately for each load level by taking the interval performance score and dividing it by the corresponding reference score for that worklet. The reference score for each worklet was determined as described above (see section 6.1 Metric/Score).
  The performance range is represented by a red line with a dot at the left for minimum performance corresponding to the lowest load level and a dot at the right for maximum performance corresponding to the highest load level. The intermediate load levels, if defined, are represented by small vertical bars.

- **Efficiency Score**
  The rightmost section displays the range of efficiency scores for all load levels. The efficiency scores are given on vertical scales at the top of each workload section.
  The efficiency scores are calculated separately for each load level by dividing the normalized-performance for that interval by the average-watts for that interval. Note that the total efficiency score for a worklet is **NOT** the average of the efficiency scores for each interval.
  The efficiency score range is represented by a green line with a triangle at the left for minimum efficiency typically corresponding to the lowest load level and a triangle at the right for maximum efficiency typically corresponding to the highest load level. The intermediate load levels, if defined, are represented by small vertical bars.

6.2.3. Sample TXT Report Output

Partial data from a sample “results.txt” report is shown below. This snapshot is from the report generated using version 1.0.0 of the SERT (Data shown in this screen shot is from an arbitrary system and meant as an example only).

SERT (TM) Report

_SPEC _SYSTEM EM65

Test Sponsor	_SPEC Company	Software Availability	Jan-2008
Tested By	_SPEC Company	Hardware/Firmware Availability	Jan-2008
SPEC License #	_0	System Source	_Single Supplier
Test Location	_Warrenton, VA, USA	Test Date	Jan 14, 2013

Summary

Workload	Worklet	Normalized Peak Performance	Watts at Lowest Load Level	Watts at Highest Load Level	sum of Normalized Performance	sum of Power (Watts)	Efficiency Score
CPU	Compress	4.571	149.6	216.0	11.443	716.5	15.971
	CryptoAES	3.974	149.4	211.2	9.939	711.6	13.967
	LU	3.861	150.2	218.9	9.648	726.5	13.279
	SOR	3.807	146.0	191.5	9.520	667.4	14.265
	XMLvalidate	4.320	150.1	211.6	10.842	718.2	15.097
	Sort	3.852	147.8	203.4	9.645	693.0	13.916
	SHA256	3.798	148.4	203.6	9.510	701.3	13.561
Memory	Flood	13.948	222.0	222.5	20.969	444.5	47.173
	Capacity	30.897	207.1	210.6	211.591	1,882.2	112.418
Storage	Sequential	4.246	117.2	125.3	6.376	242.5	26.287
	Random	2.309	113.0	116.1	3.479	229.1	15.186
Hybrid	SSJ	6.261	139.5	213.2	28.350	1,407.7	20.139
Idle	Idle	n/a	107.8	107.8	n/a	107.8	n/a

The summary results table shows the power, performance and efficiency scores for the different worklets. The results for each worklet are presented on a separate row grouped together per workload.

This table does include totalized performance and power values used to calculate the worklet efficiency score, shown in the last column.

6.2.4. Score Calculation

The total performance score (sum of Normalized Performance) is calculated by summing up the normalized performance scores of all measurement intervals of a worklet. The “sum of Power (Watts)” value is derived by adding the average-watts readings of all measurement intervals of this worklet.

The efficiency score for each worklet is defined as:

1000 * *"sum of Normalized Performance"* / *"sum of Power (Watts)"*

Efficiency for the Idle worklet is marked as not applicable (n/a) because the performance part is missing by definition. Please note that Idle power is **NOT** included in the per worklet efficiency score calculation.

Raw performance scores (not normalized) for each worklet are shown in the SERT “results-details.txt/html” report files. The details reports include tables with the scores and additional performance and power information for each interval of all worklets. For more information about worklet scores see section 8 Worklet Details.

6.3. 32-Bit Results

A 32-bit result is clearly marked as such and cannot be compared to a 64-bit result.

6.4. Validation / Verification

The SERT software components implement software checks wherever possible to increase information accuracy, verify user input, monitor run-time data collection, and validate results. The intent is to improve accuracy, remedying user errors and to prevent invalid data being reported.

When conditions or results do not meet specific criteria, warnings are displayed and error messages will appear in the SERT reports.

Examples of compliance checking are:

- Java class files in the SERT jar files have not been modified in any way.
- Host and Client JVMs must be 64-bit.
- There are no failures in the transactions that are run as part of the SERT execution.
- Minimum temperature of the SUT must be >= 20 °C.
- <2% error readings recorded by the temperature sensor for entire run, at least one sample per interval
- Non-editable sections of results.xml file must not have been edited.
- Configuration file (default name: “config-all.xml”) is compliant.
- Results must have been obtained using a released version of SERT to be compliant.
- The power analyzer used must be an accepted device per the PTDaemon. The test date must be within 1 year of calibration date. The analyzers used for the test must be the same as described in the test-environment.xml file.
- <=1% error readings for Power, <=2% for Volt, Ampere, and PF, measured only during recording period
- For each measurement interval on each analyzer, no more than 5% of all samples can have uncertainty >1%, no more than 1% can have unknown uncertainty, average uncertainty of all samples <= 1%.
- The temperature sensor must be an accepted device per the PTDaemon. The sensors used for the test must be the same as described in the test-environment.xml file.
- The results must meet the target throughput (within +2%/-4% for 90-100%, within +/- 2% for others).
- Throughput variation between clients on a host (less or equal 5%)
- Throughput variation between hosts (less or equal 5%)

All the SERT software components will perform validation checks within the domain of their functions, e.g. warnings of connection problems, log measurement errors and out-of-range conditions, warning the user of missing or incomplete information and to check the validity of some entered data.

6.5. Logging

If there are any errors or warnings generated while SERT is running, they are saved in the log files. Depending on where the error/warning occurs, the log files are created on the Controller system or the SUT.

SERT package includes two scripts, one for Windows based systems and another for Linux based systems. The two script files are below:

`-collectlogfiles.bat` is intended to be executed on a Microsoft Windows Operating System.

`-collectlogfiles.sh` is intended to be executed on a Linux or AIX system.

These scripts collect all the log files on the respective system and create a ZIP or tar.gz file respectively in the SERTlog subdirectory under the main SERT installation directory. The file is name is formatted as:

`<hostname>_YYYY-MM-DD_HHMM.<zip/tar.gz>` (where YYYY is the year, MM is the month, DD is the day, and HHMM represents the hours and minutes count.)

The ZIP/tar.gz file can be provided to SPEC for analysis and root cause analysis of the issue.

7. Worklet Design Guidelines

In order to achieve consistent results from all worklets and a broad coverage of technologies the following guidelines have been observed:

- Each worklet is adjustable to different performance levels, in particular to some predefined levels between 100% (maximum load) and 0% (idle).
- Each worklet automatically calibrates load levels based on the maximum performance measured by the tool, independent of user action.
- Precompiled binaries of the test programs are used.
- The worklets scale with the available hardware resources. More resources should result in an appropriate increase in the performance score; e.g. more processor/memory/disk capacity or additional processor/memory/disk modules yield a better result in the performance component of the efficiency rating.
- Each worklet is portable code that follows all SPEC rules for licensing, reuse, and adaptation.
- The worklets are both architecture- and OS-agnostic or accommodate different architectures and/or OSes by using mildly different code paths.
- The work accomplished by each worklet is clearly identifiable as “important” but is not required to cover “all important” types of work.

In order to follow these guidelines, most worklets use the concept of batches of discrete work, where each batch constitutes a transaction. The different load levels will be achieved by scheduling the required number of transactions, after an initial calibration phase that estimates the 100% performance level.

SERT worklets are not designed to explicitly exercise General Purpose Graphics Processing Units (GPGPUs).

7.1. CPU Worklets

The defining characteristics of the CPU worklets are:

- The worklet requires consistent processor characteristics per simulated “user” regardless of number of processors, cores, enabled threads, etc.
- At the 100% load level, the performance bottleneck is the processor subsystem.
- The worklet’s performance should increase with more processor resources, including the number of processors, the number of cores, possibly the number of logical processors, increased frequency, larger available cache, lower latency, and faster interconnect between CPU sockets.

A more detailed discussion of these worklets follows in the next chapter; generally the group includes code patterns used in compression, cryptography, numerical processing, sorting, and the processing of HTML.

7.2. Memory Worklets

The defining characteristics of the memory worklets are:

- The worklet contains consistent memory access characteristics per simulated “user” regardless of size and number of memory DIMMs.
- At the 100% load level, the performance bottleneck is the memory subsystem.
- The worklet's performance should measure a higher (better) performance score with improved memory characteristics (e.g. higher bandwidth, lower latency, total memory size), and
- The worklets as a group should reflect a combination of random and sequential reads and writes, and small and large memory accesses.

As discussed in more detail below, the worklets in this category consist of a memory throughput workload, and a memory capacity workload.

7.3. IO Worklets

Disk and Network IO components are a key part of any well-rounded picture of system performance and power. Worklets in this category are intended to reflect the performance and power usage of modern storage subsystems with higher performance, larger capacity, and extensive reliability and availability features.

SPEC recognizes that some of the items in the next two sections may not be reasonable or practical to test or measure in a meaningful way. In those cases the use of "Configuration Power/Performance Modifier" to compensate for the extra power draw associated with extra functionality is recommended.

The measurements of power and performance of either optional add-in storage controller cards or server blade enclosure storage are not in the scope of the SERT.

7.3.1. Network IO Worklets

No Network IO worklet will be included in the first release of the SERT; instead, a Network IO Configuration Power/Performance Modifier has to be established.

The main reasons for this decision are:

- The cost of testing all reasonable external test system configurations
- Initial measurements show that there are no significant differences in power utilization between 100% and 0% network utilization for today's technology.

7.3.2. Storage IO Worklets

The SERT does include Storage IO Worklets, whose key characteristics are:

- The worklets reflect consistent IO characteristics per simulated "user" regardless of system size and number of disks or the installed memory.
- The worklets consist of a combination of random and sequential accesses, reads and writes, and small and large IOs.
- At the 100% load level, the performance bottleneck is the storage subsystem.
- The worklets should score a higher (better) performance result for higher bandwidth and lower latency.
- The worklets are limited to testing individual internal storage devices only. RAID arrays and external storage devices are not supported.

7.4. Hybrid Worklets

The key characteristics of a Hybrid worklet are:

- The worklet reflects a combination of a wide variety of processor and memory-intensive tasks.
- At the 100% load level, the performance bottleneck is due to multiple subsystems.
- The combined worklets should measure a higher (better) performance score for improved processor and memory characteristics.

7.5. Idle Worklet

During idle measurements, the SUT must be in a state in which it is capable of completing workload transactions. Therefore, the idle worklet is treated in a manner consistent with all other worklets, with the exception that no transactions occur during the measurement interval.

8. Worklet Details

The following table lists the current worklets and their load levels for each of the workloads.

Workload	Load Level	Worklet Name
CPU	100%, 75%, 50%, 25%	Compress
		CryptoAES
		LU
		SHA256
		SOR
		SORT
		XMLValidate
Memory	Flood: Full, Half Capacity: 4GB, 8GB, 16GB, 128GB, 256GB, 512GB, 1024GB (see details in sections 8.8 and 8.9)	Flood2
		Capacity2
Storage	100%, 50%	Random
		Sequential
Hybrid	100%, 87.5%, 75%, 62.5%, 50%, 37.5%,25%, 12.5%	SSJ
Idle	idle	Idle

Note: Several of the worklets utilize a method named input data caching (IDC) which is described in Section 8.9.

8.1. CPU Worklet: Compress

8.1.1. General Description

The Compress workload implements a transaction that compresses and decompresses data using a modified Lempel-Ziv-Welch method (LZW). Essentially, it finds common substrings and replaces them with a variable size code. This is both deterministic and done on the fly. Thus, the decompression procedure needs no input table, but tracks the way the table was built. The algorithm is based on "A Technique for High Performance Data Compression", Terry A. Welch, IEEE Computer Vol. 17, No. 6 (June 1984), pp 8-19.

8.1.2. Sequence Execution Methods

Graduated Measurement Sequence

8.1.3. Metric

Transactions Per Second

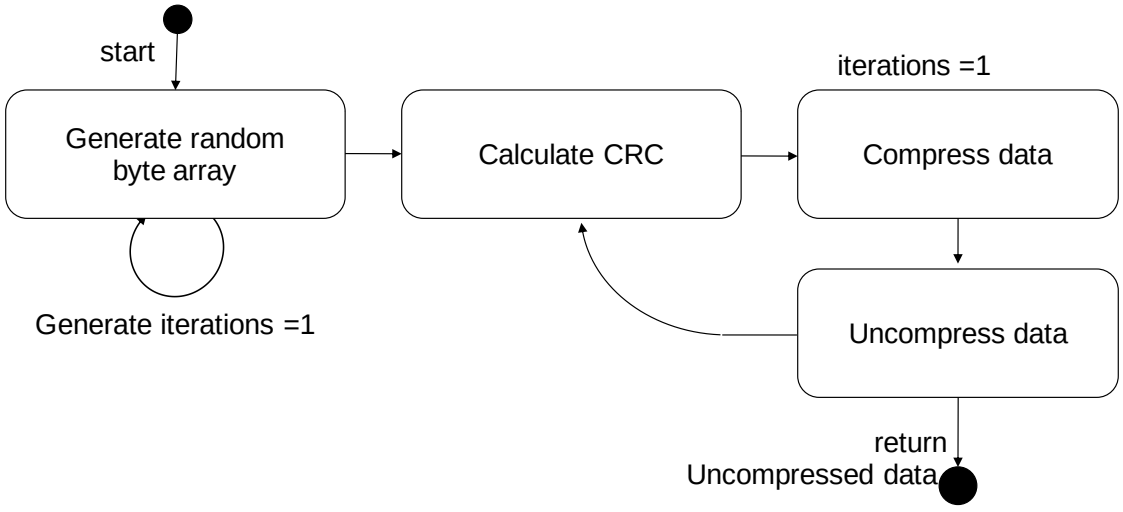
8.1.4. Required Initialization

A constant size byte array is generated on the fly before for each transaction execution. The contents of the byte array are randomly generated.

8.1.5. Configuration Parameters

`size`	Size of the input byte array for each transaction execution.
`enable-idc`	Enables/disables memory scaling using input data caching (IDC). Must be set to false.
`iterations`	Number of executions per transaction
`debug-level`	Value governs the volume of debug messages printed during execution.
`input-generate-iterations`	Number of random byte array assignment iterations

8.1.6. Transaction Code



8.2. CPU Worklet: CryptoAES

8.2.1. General Description

The CryptoAES workload implements a transaction that encrypts and decrypts data using the AES (or DES) block cipher algorithms. Which algorithm is a configurable parameter, but the current candidate version uses AES with CBC and no PKCS5 padding. Encryption and decryption are done using the Java Cryptographic Extension (JCE) framework, and the Cipher class, in particular.

8.2.2. Sequence Execution Methods

Graduated Measurement Sequence

8.2.3. Metric

Transactions Per Second

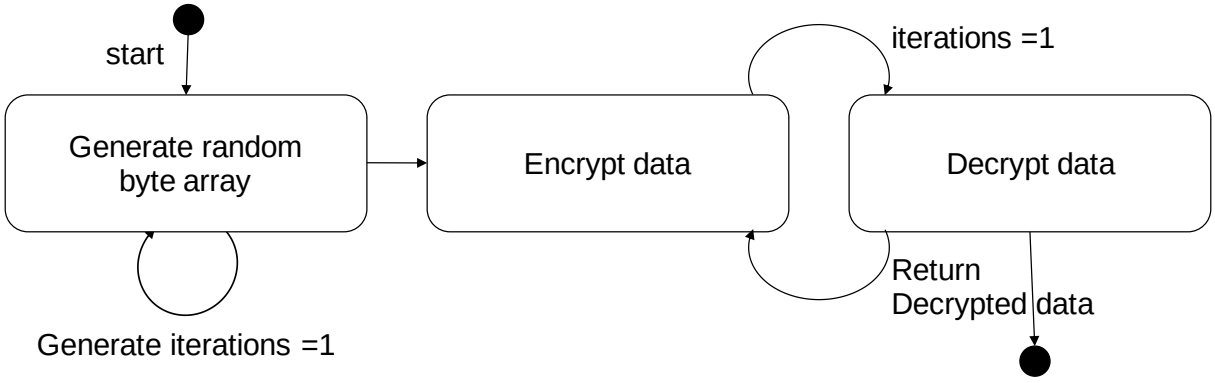
8.2.4. Required Initialization

A constant size byte array is generated on the fly before for each transaction execution. The contents of the byte array are randomly generated.

8.2.5. Configuration Parameters

`size`	Size of the input byte array for each transaction execution
`key-generator`	Key generator algorithm (AES or DESede)
`key-size`	Key size. (128 for AES, 168 for DES)
`algorithm`	Encryption algorithm. (E.g., AES/CBC/NoPadding, AES/CBC/PKCS5Padding, DESede/CBC/NoPadding, DES/CBC/PKCS5Padding)
`Level`	Number of times to perform the encryption
`enable-idc`	Enables/disables memory scaling using input data caching (IDC). Must be set to false.
`iterations`	Number of executions per transaction
`debug-level`	Value governs the volume of debug messages printed during execution.
`input-generate-iterations`	Number of random byte array assignment iterations

8.2.6. Transaction Code



8.3. CPU Worklet: LU

8.3.1. General Description

The LU workload implements a transaction that computes the LU factorization of a dense matrix using partial pivoting. It exercises linear algebra kernels (BLAS) and dense matrix operations. The algorithm is the right-looking version of LU with rank-1 updates. (Adapted from the NIST-developed SciMark benchmark).

8.3.2. Sequence Execution Methods

Graduated Measurement Sequence

8.3.3. Metric

Transactions Per Second

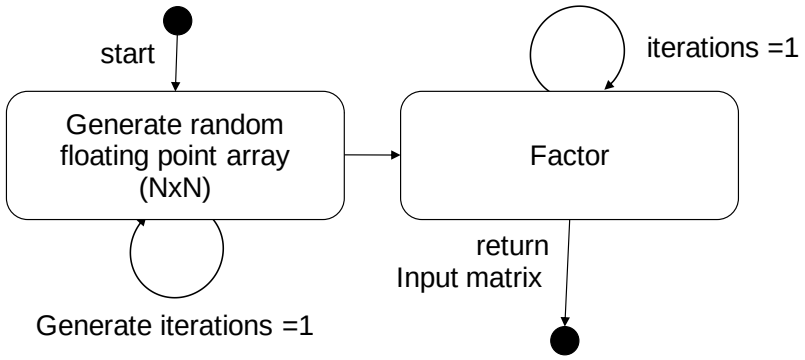
8.3.4. Required Initialization

A constant size matrix of floating point numbers is generated on the fly before for each transaction execution. The contents of the matrix are randomly generated.

8.3.5. Configuration Parameters

`matrix-dimen`	Dimension of the input floating point matrix for each transaction execution (NxN)
`enable-idc`	Enables/disables memory scaling using input data caching (IDC). Must be set to false.
`iterations`	Number of executions per transaction
`debug-level`	Value governs the volume of debug messages printed during execution.
`input-generate-iterations`	Number of random matrix assignment iterations

8.3.6. Transaction Code



8.4. CPU Worklet: SHA256

8.4.1. General Description

Hashing and encryption/decryption are two pillars of modern computer security. The SHA-256 workload utilizes standard Java functions to perform SHA-256 transformations on a byte array. This byte array is perturbed by one byte for each transaction.

8.4.2. Sequence Execution Methods

Graduated Measurement Sequence

8.4.3. Metric

Transactions Per Second

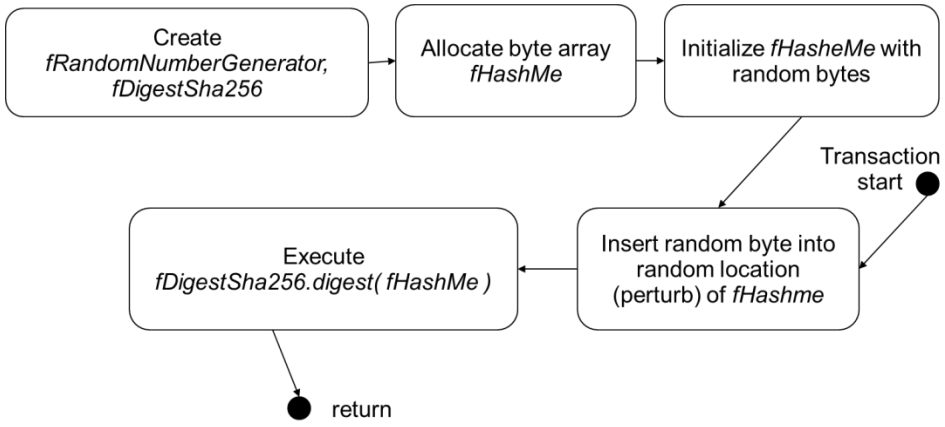
8.4.4. Required Initialization

None

8.4.5. Configuration Parameters

`Debug-level`	Detailed diagnostic information can be enabled through the debug parameter. Valid values are 0 = no additional debug information (default), -1 = debug information turned on.

8.4.6. Transaction Code



8.5. CPU Worklet: SOR

8.5.1. General Description

The Jacobi Successive Over-relaxation (SOR) workload implements a transaction that exercises typical access patterns in finite difference applications, for example, solving Laplace's equation in 2D with Drichlet boundary conditions. The algorithm exercises basic "grid averaging" memory patterns, where each `A(i,j)` is assigned an average weighting of its four nearest neighbors. Some hand-optimizing is done by aliasing the rows of `G[][]` to streamline the array accesses in the update expression. (Adapted from the NIST-developed SciMark benchmark).

8.5.2. Sequence Execution Methods

Graduated Measurement Sequence

8.5.3. Metric

Transactions Per Second

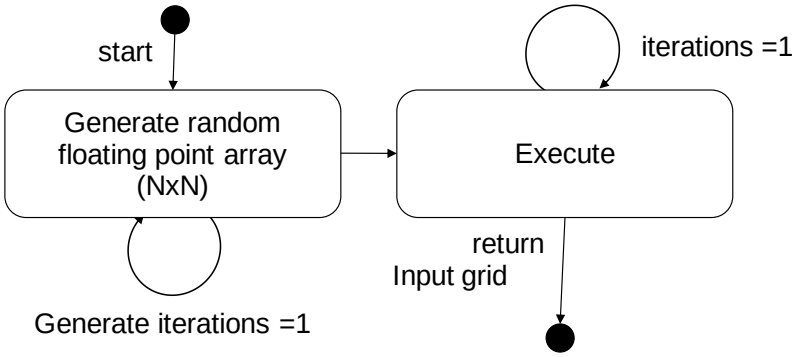
8.5.4. Required Initialization

A constant size grid of floating point numbers is generated on the fly before for each transaction execution. The contents of the grid are randomly generated.

8.5.5. Configuration Parameters

`grid-dimen`	Dimension of the input floating point grid for each transaction execution (NxN)
`enable-idc`	Enables/disables memory scaling using input data caching (IDC). Must be set to false.
`iterations`	Number of executions per transaction
`debug-level`	Value governs the volume of debug messages printed during execution.
`input-generate-iterations`	Number of random grid assignment iterations

8.5.6. Transaction Code



8.6. CPU Worklet: SORT

8.6.1. General Description

Sorting is one of the most common and important operations in computing. The SORT worklet sorts a randomized 64-bit integer array during each transaction.

8.6.2. Sequence Execution Methods

Graduated Measurement Sequence

8.6.3. Metric

Transactions Per Second

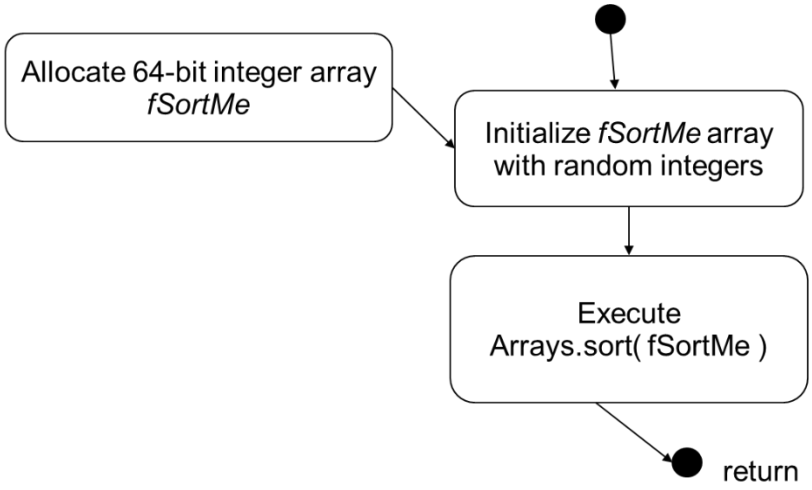
8.6.4. Required Initialization

None

8.6.5. Configuration Parameters

`Debug-level`	Detailed diagnostic information can be enabled through the debug parameter. Valid values are 0 = no additional debug information (default), -1 = debug information turned on.

8.6.6. Transaction Code



8.7. CPU Worklet: XMLValidate

8.7.1. General Description

The XML validate workload implements a transaction that exercises Java's XML validation package javax.xml.validation. Using both SAX and DOM APIs, an XML file (.xml) is validated against an XML schemata file (.xsd). To randomize input data, an algorithm is applied that swaps the position of commented regions within the XML input data.

8.7.2. Sequence Execution Methods

Graduated Measurement Sequence

8.7.3. Metric

Transactions Per Second

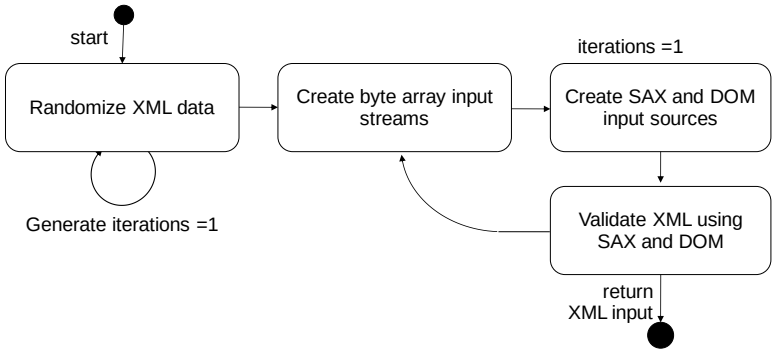
8.7.4. Required Initialization

At initialization time, both XML and XML schemata files are read in from disk and saved in a buffer for future use. (There will be no further disk IO once this is completed.) A randomization algorithm is applied to the original XML data on the fly before each transaction execution to create variations in parsing without modifying file size or complexity.

8.7.5. Configuration Parameters

`xml-schema-dir`	Specifies the directory of the XML schema file
`xml-schema-file`	Specifies the name of the XML schema file
`xml-dir`	Specifies the directory of the XML file
`xml-file`	Specifies the name of the XML file
`enable-idc`	Enables/disables memory scaling using input data caching (IDC). Must be set to false.
`iterations`	Number of executions per transaction
`debug-level`	Value governs the volume of debug messages printed during execution.
`input-generate-iterations`	Number of XML file randomization iterations

8.7.6. Transaction Code



8.8. Memory Worklet: Flood2

8.8.1. General Description

The Flood2 workload is based upon STREAM, a popular benchmark that measures memory bandwidth across four common and important array operations. For the *long* (64-bit) integer arrays used in Flood2, the following amounts of memory are involved per assignment:

1. **COPY:** *a(i) = b(i)*
   -- 8 bytes read + 8 bytes write per assignment = 16 bytes / assignment
2. **SCALE:** *a(i) = k * b(i)*
   -- 8 bytes read + 8 bytes write per assignment = 16 bytes / assignment
3. **ADD:** *a(i) = b(i) + c(i)*
   -- 16 bytes read + 8 bytes write per assignment = 24 bytes / assignment
4. **TRIAD:** *a(i) = b(i) + k * c(i)*
   -- 16 bytes read + 8 bytes write per assignment = 24 bytes / assignment

The Flood2 score is based upon the aggregate system memory bandwidth calculated from the average of these four tests multiplied by the amount of physical memory installed in the SUT. While Flood2 is based upon STREAM, it uses no STREAM code and is implemented wholly in Java.

Flood2 enhances STREAM in a variety of important ways:

1. Flood2 rewards systems with large memory configurations by scaling results based upon physical memory size.
2. Flood2 is designed to fully exploit the memory bandwidth capabilities of modern multi-core servers. Flood2 is multi-threaded and threads are scheduled to operate concurrently during bandwidth measurements ensuring maximum throughput and minimizing result variability.
3. Flood2 requires little to no user configuration, yet automatically expands the data set under test to fully utilize available memory.

Measuring aggregate system memory bandwidth on large servers with many cores and multiple memory controllers is challenging. In particular, run-to-run variability is often unmanageable with existing memory bandwidth benchmarks. Flood2 minimizes run-to-run variation by taking three memory bandwidth tests back-to-back and discarding the first and last tests. This ensures that all threads are running under fully concurrent conditions during the middle measurement which is used in Flood2 scoring calculations.

Flood2 scores scale with a SUT's aggregate memory bandwidth as well as with the SUT's physical memory configuration. CPU, storage and network performance have little to no impact on Flood2 scores.

Since the Flood2 workload always deploys a fixed number of iterations and the amount of memory under test will automatically adjust to fully utilize installed DRAM, run time will vary depending upon system configuration. On a 2.2GHz, 24-core SUT with 24 threads and 48GB of physical memory, Flood2 takes about 20 minutes to complete. Run time varies proportionally with the amount of physical memory installed in the SUT. Run time is also impacted by the overall thread count.

8.8.2. Sequence Execution Methods

FixedIterationsDirectorSequence – Flood2 is executed for a given set of iterations specified within *config.xml.*

8.8.3. Metric

By default, *Flood2* runs the four memory bandwidth tests that are also part of STREAM, but were independently implemented (i.e. no code was borrowed from STREAM). Note that several other memory bandwidth tests are also implemented for future versions of *Flood2,* but are not currently utilized (e.g. STREAM2 tests).

Each test returns a score that is based upon the measured raw memory bandwidth scaled by the square-root of the amount of physical memory:

```
score = bandwidthGbs * sqrt(fPhysicalMemorySizeInGB);
```

Note: In SERT 1.1.0 the Flood worklet was replaced with the Flood2 worklet which used an updated method to calculate the metric.

These scores are scaled (based upon load level), aggregated, and averaged. The load level is 100% in *Flood_Full* and 50% in *Flood_Half* (which uses data sets reduced down to 50% of *Flood_Full*):

```
//measurement:

result = Math.abs( RunCopySystem() * fLoadLevel );

result += Math.abs( RunScale() * fLoadLevel );

result += Math.abs( RunAdd() * fLoadLevel );

result += Math.abs( RunTriad() * fLoadLevel );

result /= 4; // take average
```

In turn, SERT aggregates these results for all *Flood2* threads.

If desired, *Flood2* can be made to return raw memory bandwidth by specifying the following run parameter:

```
<parameter name="return-bandwidth">true</parameter>
```

This can be useful for diagnostic purposes, but the same value can be derived by dividing normal *Flood2* results by the square root of SUT memory capacity in GB.

Typically, *Flood2* achieves raw memory bandwidth results around 60%-90% that of highly tuned, multithreaded (OMP) C++ STREAM binaries, but usually with much better reproducibility. Furthermore, *Flood2* utilizes nearly all available memory whereas STREAM uses much smaller data sets.

8.8.4. Required Initialization

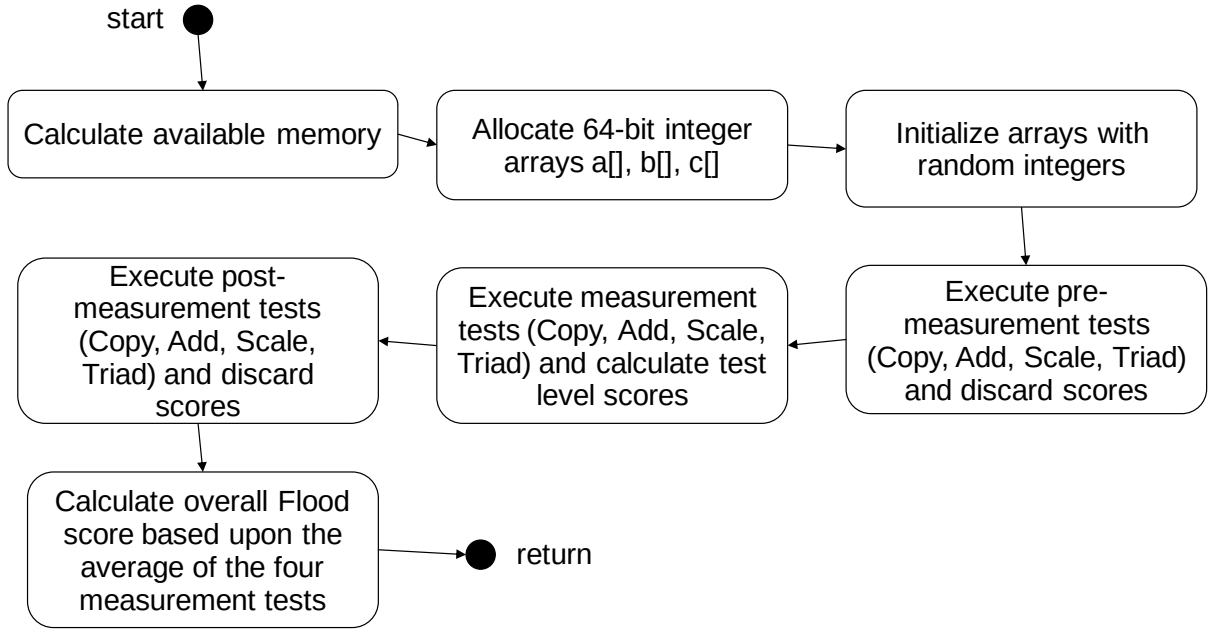
Flood2 calculates the amount of memory available to the thread and creates three 64-bit (*long*) integer arrays, a[], b[], and c[], to completely utilize all available space. These arrays are initialized with random data. To ensure full load concurrency during bandwidth measurements, a complete set of pre-measurement tests is launched prior to an identical measurement period followed by identical post-measurement tests. Only the test results for the measurement period are utilized for Flood2 score generation.

8.8.5. Configuration Parameters

`memory-under-test`	The default value of "-1 MB" turns on automatic configuration of the data set size. However, the user can override this behavior and explicitly define the amount of memory to test per JVM. Valid values are (san quotation marks): "200 MB", "1.1 GB", "10000000 B".
`iterations`	Flood2 internally iterates the number of memory bandwidth tests based upon the value of the iterations parameter. The default is 100.

`debug-level`	Detailed diagnostic information can be enabled through the debug parameter. Valid values are 0 = no additional debug information (default), 1 = debug information turned on, 2 = detailed debug information.
`return-bandwidth`	The raw, aggregate system memory bandwidth calculated by Flood2 can be obtained by setting the parameter return-bandwidth to "true" in which case Flood2 will return measured memory bandwidth instead of a score. The default value is "false".

8.8.6. Transaction Code



8.9. Memory Worklet: Capacity2

8.9.1. General Description

The XML validate workload implements a transaction that exercises Java's XML validation package javax.xml.validation. Using both SAX and DOM APIs, an XML file (.xml) is validated against an XML schemata file (.xsd). To randomize input data, an algorithm is applied that swaps the position of commented regions within the XML input data.

Memory scaling in Capacity2 is done through a scheme known as input data caching (IDC). In IDC, the universe of possible input data (here, randomized XML file data) is pre-computed and then cached within memory before the start of the workload. During workload execution, the input data for a particular transaction instance is then chosen randomly and retrieved from this cache rather than computed on the fly.

The data store size is increased incrementally with each interval. If the data store size is less than the amount of physical memory available to the worklet, data is retrieved from the cache. Once the data store size is larger than the max size of the data cache, a 'cache miss' penalty is incurred when the transaction randomly chooses a data store element that is not currently in the cache. When this occurs, multiple iterations of re-generating a cache element are performed to apply a cache miss penalty and the transaction rate decreases. The more memory the system has, the larger of a data store size can be executed before the transaction rate begins to lower as a result of cache misses.

In addition to the transaction characteristics, the maximum cache size is applied to the scoring algorithm. Cache size is computed as:

Physical Memory * data-cache-to-heap-ratio (currently .6)

While this worklet does contain transactions that are memory oriented, there is still a component that is influenced by CPU performance.

8.9.2. Sequence Execution Methods

A Modified Parameters Sequence. Each interval consists of a No Delay Series where the parameter data-store-size changes with each interval.

8.9.3. Metric

Transactions Per Second * sqrt(Physical Memory size (GB))

The overall score will scale with memory size.

Note: In SERT 1.1.0 the Capacity worklet was replaced with the Capacity2 worklet which used an updated method to calculate the metric.

8.9.4. Required Initialization

At initialization time, both XML and XML schemata files are read in from disk and saved in a buffer for future use. (There will be no further disk IO once this is completed.) IDC initialization follows during which all possible input data sets are pre-computed and cached in memory. For each input data set, a randomization algorithm is applied to the original XML data to create variations in parsing without modifying file size or complexity.

8.9.5. Configuration Parameters

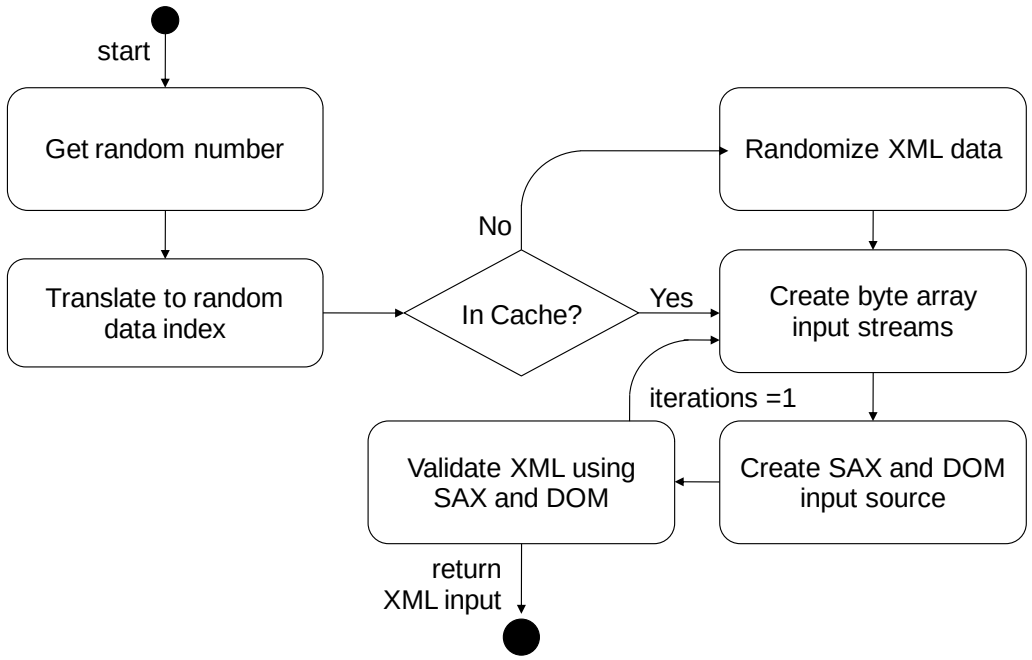
`xml-schema-dir`	Specifies the directory of the XML schema file
`xml-schema-file`	Specifies the name of the XML schema file
`xml-dir`	Specifies the directory of the XML file
`xml-file`	Specifies the name of the XML file

`enable-idc`	Enables/disables memory scaling using input data caching (IDC). Must be set to false.
`iterations`	Number of executions per transaction.
`debug-level`	Value governs the volume of debug messages printed during execution.
`input-generate-iterations`	Number of XML file randomization iterations.

Additional IDC configuration parameters:

`store-type`	Specifies the algorithm to use in generating data when a cache miss occurs
`locality-distribution`	Specifies the probability distribution to use when randomly choosing input data indices
`data-store-size`	Specifies the size of the universe of possible input data
`data-cache-size`	Specifies the size of the input data cache
`data-cache-report-interval`	Governs the frequency of output messages on cache hit/miss ratio
`custom-score-policy`	Specifies the algorithm to use in computing custom score reflecting cache size configuration.
`data-cache-size-scale factor`	Specifies the scaling factor to use in the DataCacheSizeMultiplierGB custom scoring algorithm
`data-cache-to-heap-ratio`	Ratio of cache size to JVM heap size used in automatic cache sizing

8.9.6. Transaction Code



8.10. Storage IO Workload

8.10.1. General Description

The Storage-Workload has four different transactions, two random and two sequential transaction-pairs. Each pair has a write and a read transaction.

8.10.2. Sequence Execution Methods

Graduated Measurement Sequence

8.10.3. Metric

Transactions (IO operations) per second.

8.10.4. Required Initialization

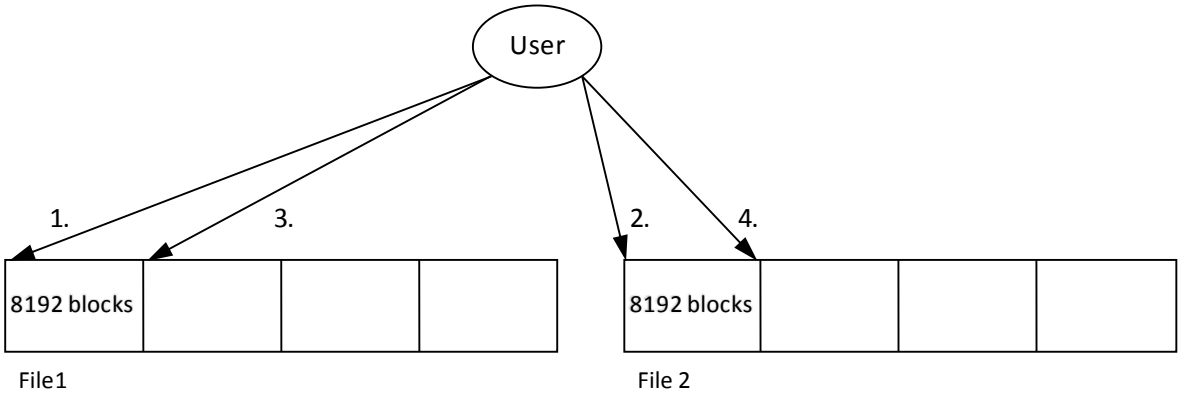
A set of files is created before execution of the transaction

8.10.5. Configuration Parameters

`file-size-bytes`	size of a file
`file-per-user`	number of files opened by each user
`file-path`	location of the files - In this example the path is “D:\data\”, please note that the files always reside in a subfolder called “data”.
`sequential-max-count`	amount of blocks that are accessed by the sequential transaction in one file before the next file is addressed

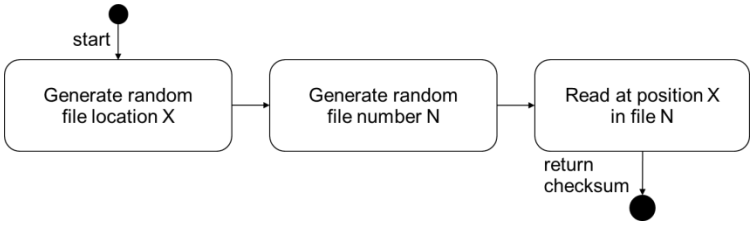
Example:

```
<file-size-bytes>1000000<file-size-bytes>

<file-path>D:\</file-path>

<file-per-user>2</file-per-user>

<sequential-max-count>8192</sequential-max-count>
```

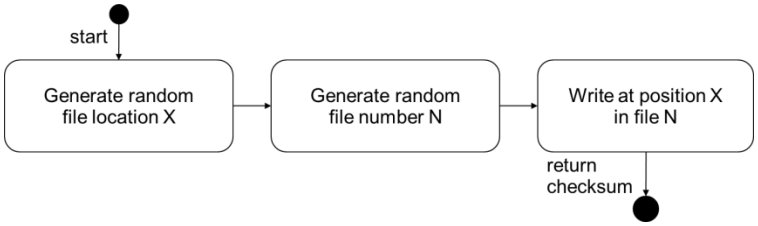


[File Example (2 files per user and max-count of 8192)]

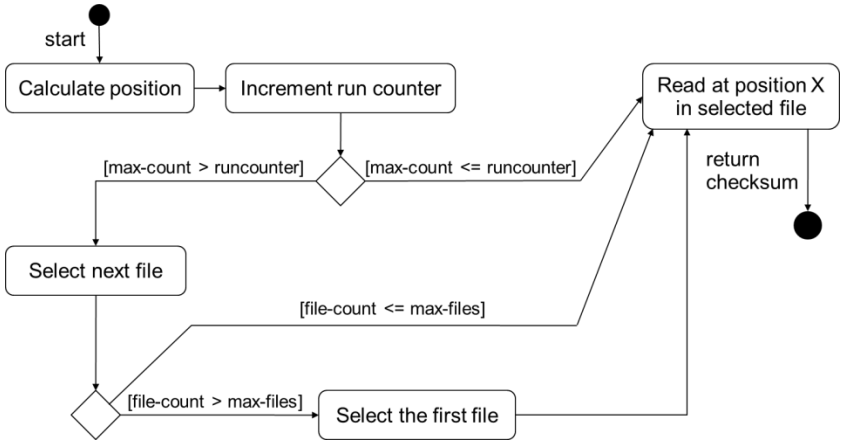
8.10.6. Transaction – Code 1 - RandomRead



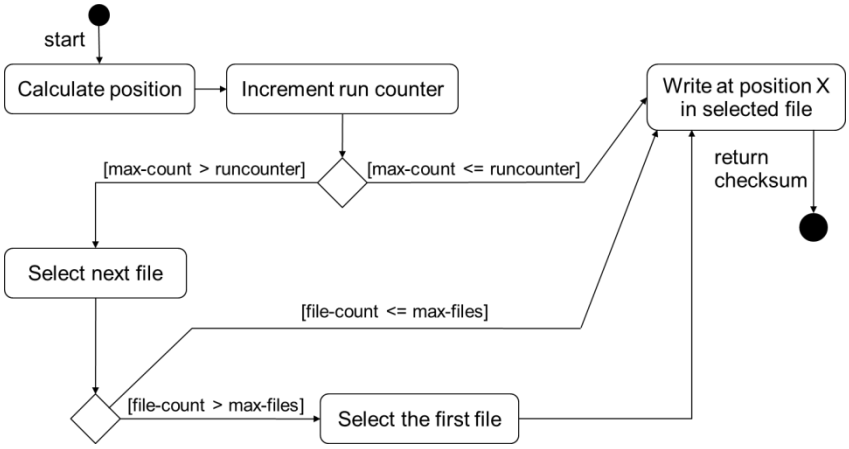
8.10.7. Transaction – Code 1 - RandomWrite



8.10.8. Transaction – Code 2 – SequentialRead



8.10.9. Transaction – Code 2 – SequentialWrite



8.11. Hybrid Worklet: SSJ

8.11.1. General Description

SSJ is a simulated Online Transaction Processing (OLTP) workload, and represents a Server Side Java application. It is based on the SSJ workload in SPECpower_ssj2008, which was based on SPECjbb2005, which was inspired by the TPC-C specification; however, there are several differences between all of these workloads, and SSJ results are not comparable to any of these benchmarks.

The Hybrid Worklet exercises the CPU (s), caches, and memory of the SUT. The peak throughput level is determined by maximum number of transaction of the above type the system can perform per second. Once the peak value of the transactions is determined on a given system, the worklet is run from peak (100%) down to the system idle in a graduated manner.

The performance of the Hybrid Worklet depends on the combination of the processor type, number of processors, their operating speed, and the latency and bandwidth of the memory subsystem of the system.

SSJ includes 6 transactions, with the approximate frequency shown below:

- New Order (30.3%) – a new order is inserted into the system
- Payment (30.3%) – records a customer payment
- Order Status (3.0%) – requests the status of an existing order
- Delivery (3.0%) – processes orders for delivery
- Stock Level (3.0%) – finds recently ordered items with low stock levels
- Customer Report (30.3%) – creates a report of recent activity for a customer

8.11.2. Sequence Execution Methods

Graduated Measurement Sequence

8.11.3. Metric

Transactions Per Second

8.11.4. Required Initialization

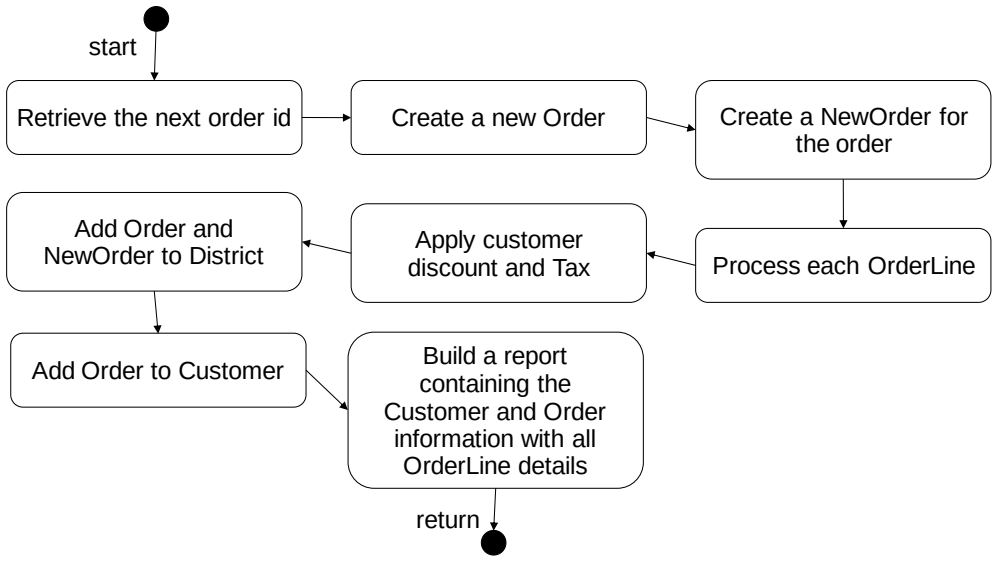
Each user represents a warehouse. During initialization, each warehouse is populated with a base set of data, including customers, initial orders, and order history.

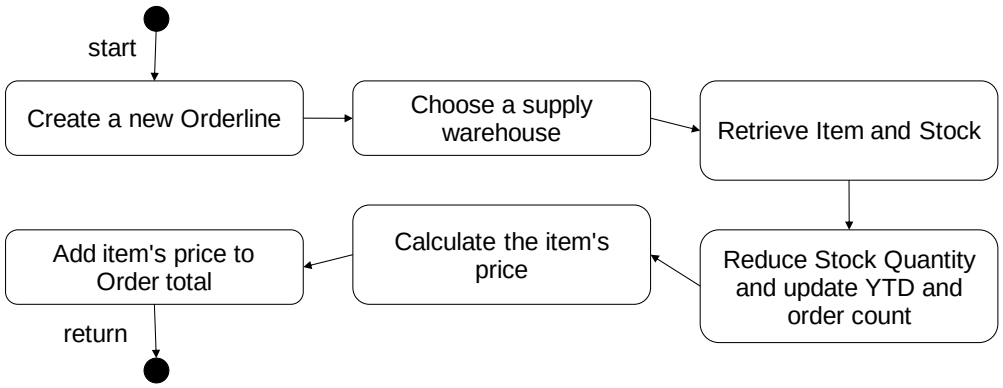
8.11.5. Configuration Parameters

The SSJ workload does not have any supported configuration parameters.

8.11.6. New Order Transaction

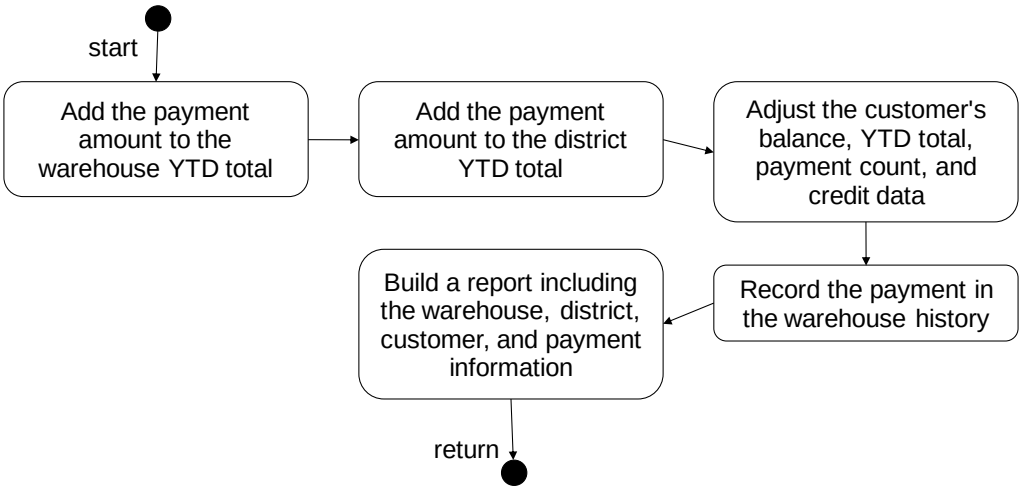
The input for a New Order Transaction consists of a random district and customer ID in the user's warehouse, and a random number of orderlines between 5 and 15.





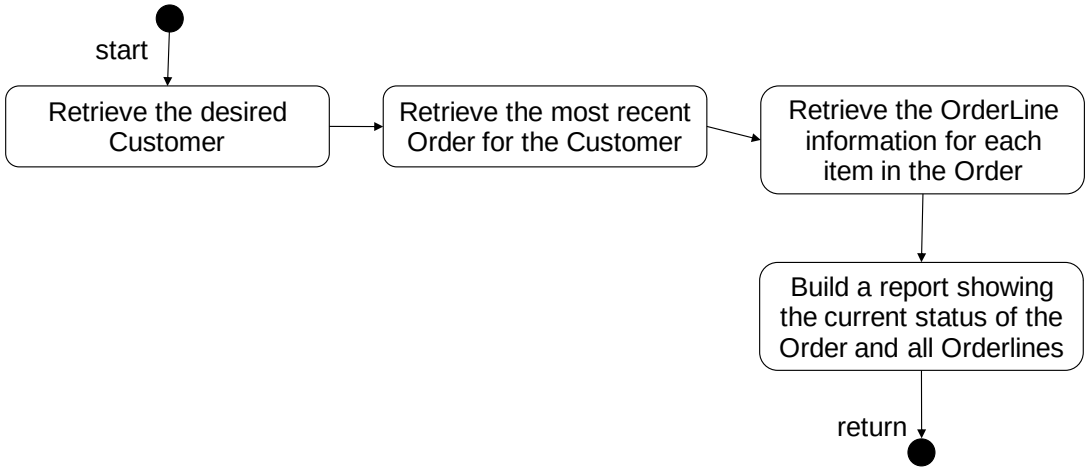
8.11.7. Payment Transaction

The input for a Payment Transaction consists of a random district from the user's warehouse, a random customer id or last name (from either the user's warehouse or a remote warehouse), and a random payment amount.



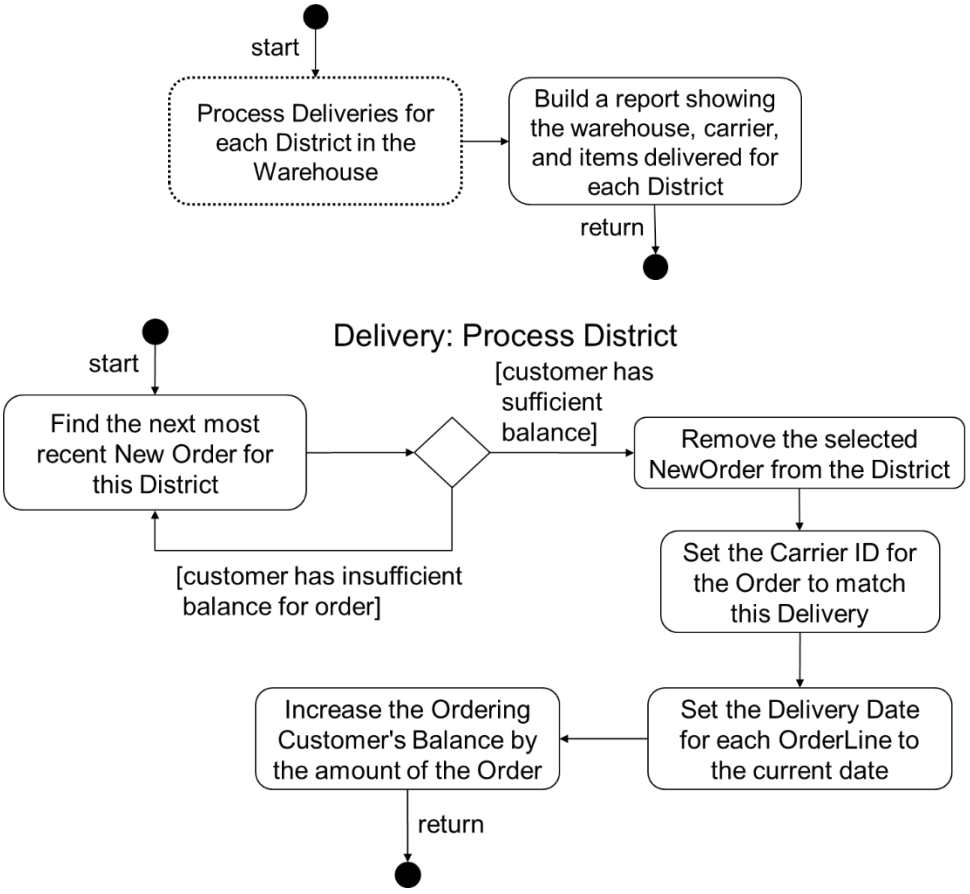
8.11.8. Order Status Transaction

The input for an Order Status Transaction consists of a random district and either a customer ID or last name from the user's warehouse.



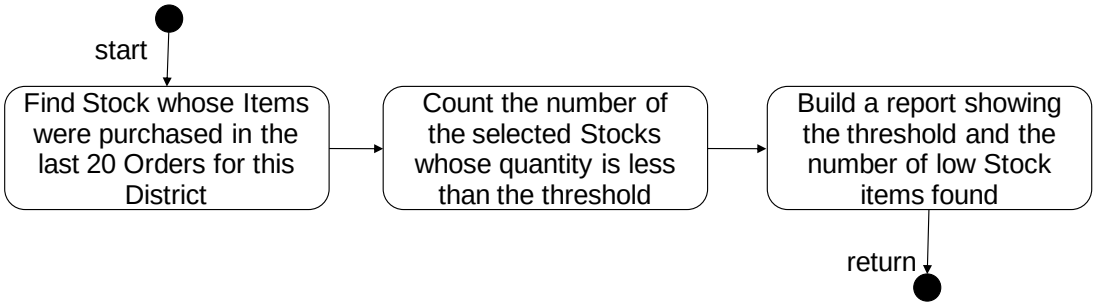
8.11.9. Delivery Transaction

The input for a Delivery Transaction is a random carrier ID.



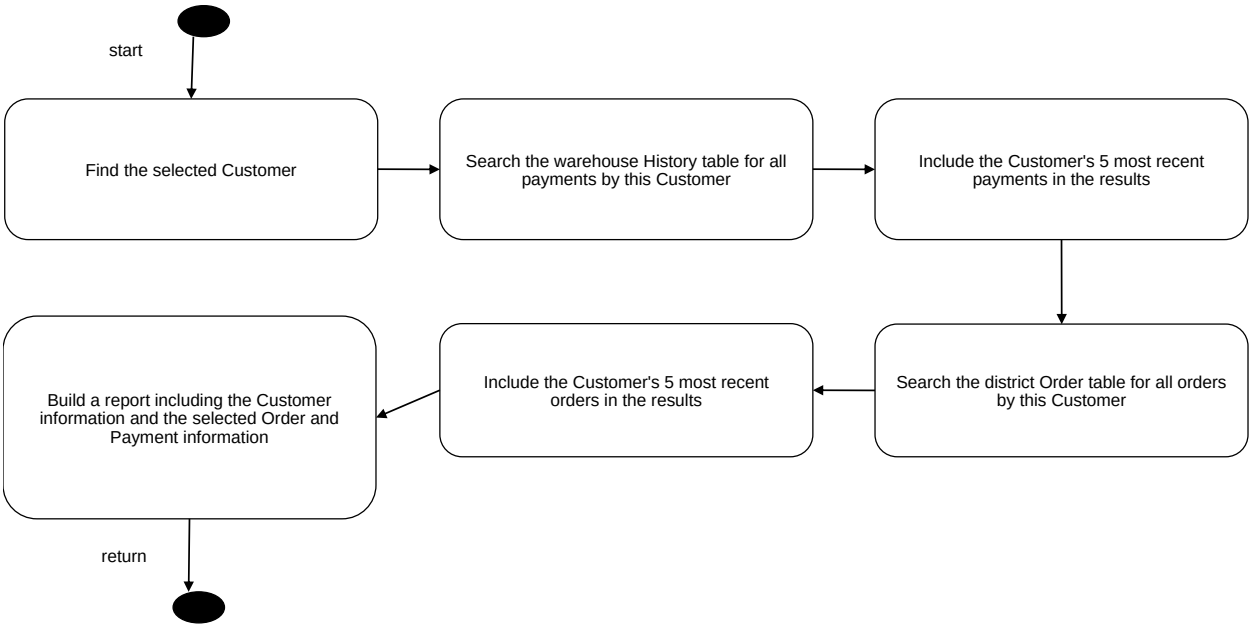
8.11.10. Stock Level Transaction

The input for a Stock Level transaction is a random district from the user's warehouse and a random “low level” threshold between 10 and 20.



8.11.11. Customer Report Transaction

The input for a Customer Report transaction consists of a random district from the user's warehouse and a random customer ID or last name (from either the user's warehouse or a remote warehouse).



9. Energy Efficiency Regulatory Programs and the SERT

The SERT is designed to be utilized in Energy Efficiency Regulatory Programs world-wide. The EPA's ENERGY STAR development team is currently working on Version 2.0 of their Computer Server Specification[2]. Version 2.0 aims to evolve the program by adding a means to measure the overall efficiency of the server while it is performing actual computing work via an Active Mode Efficiency Rating Tool. The Korean's government has tasked the Telecommunications Technology Association (TTA) to create the Korean's Server Energy Efficiency Labeling Program (EELP).

SPEC applauds both efforts for their goal of driving towards greater energy efficiency in IT Equipment, and SPEC considers these programs an industry partner in this effort. The development of an Active Mode Efficiency Rating Tool is an essential component in the ongoing effort to reduce world-wide energy consumption and paves the way for successful regulatory programs that have the potential to harmonize energy efficiency programs worldwide.

SPEC recognizes that there are other Energy Efficiency Regulatory Programs around the globe that have an interest in establishing criteria for energy efficient computer servers. SPEC welcomes the opportunity to work with the EPA, the TTA, and other agencies as they use the SERT in support of their regulatory programs.

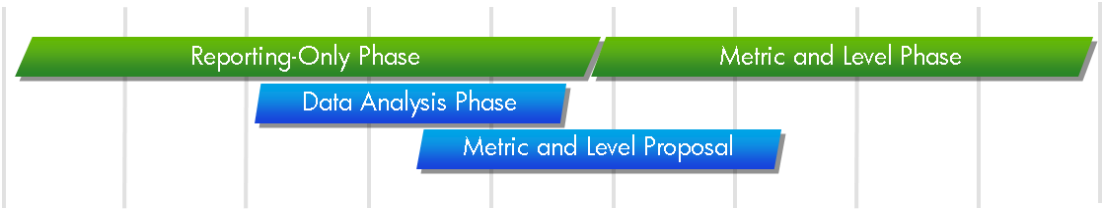
The breadth of the SERT's functional coverage allows for a broad span of configurations and shows the different aspects of computer servers at a variety of load levels. Also, the near out-of-box tuning provides relevance to end-consumers. As more programs adopt the use of the SERT, the base of measured data will grow. The SERT must be used in the intended matter in order to ensure the accurate and repeatable results; therefore, we recommend the inclusion of the following items in every Energy Efficiency Regulatory Program:

9.1. Measurement

The provided SERT test kit must be used to run and produce measured SERT results. Its results are not comparable to power and performance metrics from any other application.

9.1.1. Scoring and Metric

The complexity of performance and power measures across components at multiple target load levels makes creation of a metric difficult. Therefore, it is recommended to implement a ~9-12 month reporting-only phase first. Once a sufficient level of data is collected from report-only submissions, SPEC plans to recommend a data-driven metric and scoring algorithm.



9.1.2. Configuration Power/Performance Modifier

These are "substitutions" for real measurements for items the SERT cannot measure or for which the performance cannot be determined (e.g., redundant power supplies) and need to be created during the *Metric and Level Proposal* phase. The design allows for modifiers; nevertheless, SERT 1.0.0 does not implement this feature. The intention is to build on data collected from the first version of the SERT to create modifier proposals for a future revision.

[2] US Environmental Protection Agency – Energy Star Program Requirements for Computer Servers. http://www.energystar.gov/index.cfm?c=revisions.computer_servers

9.2. SERT Binaries and Recompilation

Valid runs must use the provided binary files and these files must not be updated or modified in any way.

9.3. Manual Intervention

No manual intervention or optimization for the SUT or its internal and external environment is allowed during the test measurement.

9.4. Public Usage of SERT Results Information

In general, a clear goal of every Energy Efficiency Regulatory Programs is to have the broadest possible participation among vendors. Experience in the computer industry's performance benchmark community demonstrates that when performance details become available for marketing purposes, only vendors with superior (at the time of publication) products are incented to publish results. To encourage broader participation across the industry, a set of strong rules must be in place that will restrict marketing use of any of the detailed information generated by the tool. No data besides the actual qualification should be utilized in Energy Efficiency Regulatory Programs Partners' marketing collateral. These rules will be stipulated in both the license for the tool and the Partner agreement.

Note that, while these rules are not strictly a part of the tool "design", the existence of these rules are necessary to allow the flexibility of the design and the delivery of detailed consumer information that is desired.

Public Usage Rules:

- Competitive comparisons that promote the use of one product over another and use numeric data generated by the SERT are expressly disallowed.
- The only information provided by the SERT that can be used for marketing collateral is the qualification of a server configuration or server family for an energy efficiency program such as EPA ENERGY STAR.
- The only information provided by the SERT that can be used for public comparison that promotes the use of one product over another when removed from the context of the full Power and Performance Datasheet is the ENERGY STAR qualification of a server configuration or server family, or a similar qualification defined by another Agency. All other publicly available information is provided in the datasheet and references must be made to this document in its entirety.
- If the tool is used for research to generate information outside of the ENERGY STAR program or similar programs, the information may not be compared to the results that are associated with an official energy efficiency program, such as ENERGY STAR, and competitive comparisons may not be made using the data generated
- 32-bit and 64-bit results cannot be compared to each other and need to be in separate categories.

9.5. General Availability (GA)

The implementation of the System under Test must be generally available, documented, and supported in order to ensure that the systems and their hardware and software components actually represent real products that solve real business and computational problems.

9.6. Accredited, Independent Laboratory

The requirement to use accredited, independent laboratories may place a large burden on the partners of Energy Efficiency Regulatory Programs, especially smaller companies. We recommend the use of an independent laboratory as an option, but are not implementing this as a requirement.

9.7. Supply Voltage Tolerance

SERT is designed for environments that have a supply voltage tolerance of ± 5%.

10. Logistics

The licensing and pricing structures as well as the support and maintenance models that will be used for the SERT are established by the SPEC Board of Directors.

10.1. Future Enhancement Ideas

10.1.1. Worklets

Worklet improvements (e.g., data output verification) and additional worklets are being investigated in order to keep up with upcoming technologies.

10.1.2. DC Power

If sufficient testing and support commitments are provided, SPEC will consider including support for DC-powered server measurements in the SERT. This is dependent on similar support for DC-powered servers in the SPEC PTDaemon.

10.1.3. Additional Operating Environments

Please see section 2.11.

10.2. Design Feedback Mechanism

The SERT development team will evaluate input from a broad spectrum of industry experts during the entire development process. Please provide your detailed feedback to the SPECpower Committee via http://www.spec.org/sert/feedback/issuereport.html.

10.3. Trademark

SPEC and the names SERT, SPECpower_ssj, and SPEC PTDaemon are trademarks of the Standard Performance Evaluation Corporation. Additional product and service names mentioned herein may be the trademarks of their respective owners.

10.4. Copyright Notice