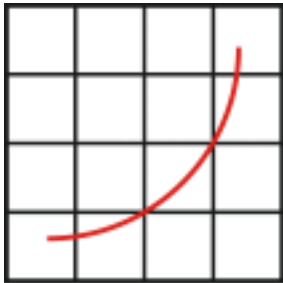


SPEC – Power and Performance



spec[®]

Run and Reporting Rules

SPECpower_ssj2008

Standard Performance Evaluation Corporation

Table of Contents

1	Introduction	4
1.1	Philosophy.....	4
1.1.1	Applicability.....	4
1.1.2	Optimizations	4
1.2	Caveats	5
1.3	Research and Academic Usage	5
2	Run Rules	5
2.1	Measurement	5
2.2	Initializing and Running Benchmark.....	5
2.3	Workload.....	5
2.3.1	Manual Intervention	5
2.3.2	Sequence of Target Loads	6
2.3.2.1	The Active Idle Interval.....	6
2.4	SUT Configuration Parameters.....	6
2.5	Benchmark Control Parameters.....	6
2.5.1	Warehouse Count and Override.....	7
2.5.2	Validity Checks	7
2.6	Optimization Flags	9
2.7	Testbed Configuration.....	9
2.8	Line Voltage Source {VoltageValidator}.....	9
2.9	Environmental Conditions	10
2.9.1	Air Flow	10
2.10	General Availability	10
2.10.1	SUT Availability for Historical Systems.....	11
2.11	System(s) Under Test (SUT)	11
2.11.1	Electrical Equivalence.....	11
2.11.2	Hardware	11
2.11.2.1	Network Interfaces.....	12
2.11.3	Software.....	12
2.12	Java Specifications	13
2.12.1	Feedback Optimization and Precompilation	13
2.12.2	Benchmark Binaries and Recompilation.....	13
2.13	Power and Temperature Measurement	13
2.13.1	Power Analyzer Setup	13
2.13.2	Power Analyzer Requirements	14
2.13.3	Temperature Sensor Requirements	14
2.13.4	Supported and Compliant Devices	14
2.13.5	Acceptance Process for New Measurement Devices.....	15
3	Reporting Rules	15
3.1	Reporting Metric and Result	15
3.1.1	Publication	16
3.1.1.1	Disclosure Requirement	16
3.1.2	Estimates	16
3.1.3	Comparison to Other Benchmark Suites	16
3.1.4	Addendum to OSG Fair Use Policy	16
3.2	Reproducibility.....	16
3.3	Testbed Configuration Disclosure.....	16
3.3.1	General Availability Dates.....	16
3.3.2	FDR Headline	17
3.3.3	Benchmark Results Summary	17
3.3.3.1	Aggregate SUT Data	17
3.3.4	SUT.....	17
3.3.4.1	Shared Hardware	18
3.3.4.2	Set: 'N'	18
3.3.4.3	SUT Hardware - single node(s).....	18
3.3.4.4	SUT Software	19

3.3.4.5	System Under Test Notes	20
3.3.5	Controller System	20
3.3.5.1	Power Analyzer and Temperature Sensor	20
3.3.6	Disclosure Notes.....	21
3.3.7	Electrical and Environmental Data.....	21
4	Submission Requirements for SPECpower_ssj2008	21
5	SPECpower_ssj2008 Benchmark Kit Overview	22
5.1	Documents overview.....	22
5.2	Trademark.....	22

SVN Revision: 1168

SVN Date: 2016/05/11 18:25:20

1 Introduction

SPECpower_ssj2008 is the first generation SPEC benchmark for evaluating the AC power and performance of server class computers. This document specifies the guidelines on how SPECpower_ssj2008 V1.12 is to be run for measuring and publicly reporting AC power and performance results of servers. These rules abide by the norms laid down by SPEC in order to ensure that results generated with this benchmark are meaningful, comparable to other generated results, and repeatable, with documentation covering factors pertinent to reproducing the results.

Per the SPEC license agreement, all results publicly disclosed must adhere to these Run and Reporting Rules.

To check for possible updates to the Run and Reporting Rules, please see http://www.spec.org/power/docs/SPECpower_ssj2008-Run_Reporting_Rules.pdf.

1.1 Philosophy

SPEC believes the user community will benefit from an objective series of benchmark results, which can serve as a common reference and be considered as part of an evaluation process. SPEC expects that any public use of results from this benchmark suite must be for the Systems Under Test (SUTs) and configurations that are appropriate for public consumption and comparison. For results to be publishable, SPEC requires:

- Proper use of the SPEC benchmark tools as provided.
- Availability of an appropriate full disclosure report (FDR).
- Availability of the Hardware and Software used (see section 3.3.1).
- Support for all of the appropriate protocols.

1.1.1 Applicability

SPEC intends that this benchmark measures the AC power and performance of systems providing environments for running server-side Java applications. It is not a J2EE benchmark and therefore it does not measure Enterprise Java Beans (EJBs), servlets, Java Server Pages (JSPs), etc.

The AC power consumption measured by this benchmark should not be assumed to represent the AC power consumption of other applications on the same hardware.

While this benchmark was designed to be a measure of computer servers, SPEC acknowledges that it may also be possible to measure other classes of computing devices. Given the speed of technology advances in the industry, SPEC does not arbitrarily restrict the type of system on which the benchmark is measured. However, since it would be misleading to draw comparisons between systems that are intended for substantially different uses, this document includes rules to promote fair comparisons between systems that are intended for similar purposes. Also restrictions are imposed on creating results with systems that rely on a battery for power for extended periods (See section 2.8).

Note that while it may be possible to run this benchmark on personal systems, SPEC provides a substantial suite of benchmarks intended for evaluation of workstations that should be considered (<http://www.spec.org/benchmarks.html#gwpg>).

1.1.2 Optimizations

SPEC is aware of the importance of optimizations in producing the best system power and performance. SPEC is also aware that it is sometimes difficult to draw an exact line between legitimate optimizations that happen to benefit SPEC benchmarks and optimizations that specifically target a SPEC benchmark. However, with the rules below, SPEC wants to increase the awareness of implementers and end users of issues of unwanted benchmark-specific optimizations that would be incompatible with SPEC's goal of fair benchmarking.

- Hardware and software used to run the SPECpower_ssj2008 benchmark must provide a suitable environment for running typical server-side Java programs. (Note: this may be different from a typical environment for client Java programs)
- Software optimizations must generate correct code for a class of programs, where the class of programs must be larger than a single SPEC benchmark.

- Hardware and/or software optimizations must improve AC power and/or performance for a class of programs, where the class of programs must be larger than a single SPEC benchmark.
- The vendor encourages the implementation for general use.
- The implementation is generally available, documented, and supported by the providing vendor(s).

Furthermore, SPEC expects that any public use of results from this benchmark must be for configurations that are appropriate for public consumption and comparison. In the case where it appears that the above guidelines have not been followed, SPEC may investigate such a claim and take action in accordance with current policies.

1.2 Caveats

SPEC reserves the right to investigate any case where it appears that these guidelines and the associated benchmark run and reporting rules have not been followed for a public SPEC benchmark claim. SPEC may request that the claim be withdrawn from the public forum in which it appears and that the benchmarker correct any deficiency in product or process before submitting or publishing future results.

SPEC reserves the right to adapt the benchmark codes, workloads, and rules of SPECpower_ssj2008 as deemed necessary to preserve the goal of fair benchmarking. SPEC will notify members and licensees whenever it makes changes to the benchmark and may rename the metrics. In the event that the workload and/or metrics are changed, SPEC reserves the right to republish, in summary form, "adapted" results for previously published systems, converted to the new metric. In the case of other changes, a republication may necessitate retesting and may require support from the original test sponsor.

Relevant standards are cited in these run rules as URL references, and are current as of the date of publication. Changes or updates to these referenced documents or URLs may necessitate repairs to the links and/or amendment of the run rules. SPEC will notify members and licensees whenever it makes changes to the suite.

1.3 Research and Academic Usage

Please consult the SPEC Fair Use Rule for Research and Academic Usage (<http://www.spec.org/fairuse.html#Academic>) for SPECpower_ssj2008.

2 Run Rules

2.1 Measurement

The provided SPECpower_ssj2008 tools must be used to run and produce measured SPECpower_ssj2008 results. The SPECpower_ssj2008 metric is a function of the SPECpower_ssj2008 workload (see section 2.3), and the defined benchmark control parameters (see section 2.5). SPECpower_ssj2008 results are not comparable to power and performance metrics from any other application.

2.2 Initializing and Running Benchmark

For guidance, please consult the latest User Guide) and Measurement Setup Guide on the SPEC's website (http://www.spec.org/power_ssj2008).

2.3 Workload

SPECpower_ssj2008 exercises a Java application workload. A detailed description can be found in the latest version of the design document on SPEC's website (http://www.spec.org/power_ssj2008).

2.3.1 Manual Intervention

No manual intervention or optimization to the controller, SUT or its internal and external environment is allowed during the benchmark run.

2.3.2 Sequence of Target Loads

The benchmark runs at multiple target loads to determine the AC power consumption of the SUT under varying processing loads. First, the maximum throughput achievable by the SUT is determined by running the workload unconstrained for at least 3 calibration intervals. The maximum is set as the arithmetic average of the throughputs achieved during the final two calibration interval runs. The workload is then run in a controlled manner, with delays inserted into the workload stream, to obtain total throughputs of 100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, and 10% of the maximum throughput. The delays inserted into workload streams are exponentially random with a fixed maximum of 10 seconds. During each of these target loads, the power characteristics of the SUT as well as the temperature are recorded. Finally, the power characteristics and temperature are measured and recorded during an idle interval during which the SUT processes no Java transactions. The preceding sequence is automatically implemented by the benchmark harness and must not be changed for a compliant run.

2.3.2.1 The Active Idle Interval

During active idle, the SUT must be in a state in which it is capable of completing workload transactions. The active idle measurement interval is treated in a manner consistent with all other target load levels, with the exception that no transactions occur during the active idle interval.

The intent in defining and automating active idle power measurement within the SPECpower_ssj2008 benchmark is to prevent manipulation of idle power measurements. The benchmark workload and the JVM process in which it is running are to remain active without interruption for the duration of this phase.

2.4 SUT Configuration Parameters

The "SPECpower_ssj_[name].props" file contains configuration information used to generate the final report, and values must be populated appropriately by the tester to reflect the SUT for a compliant run.

2.5 Benchmark Control Parameters

There are a number of parameters which control the operation of SPECpower_ssj2008. The "SPECpower_ssj.props" file is used to control the parameters of the benchmark run. The properties in the "Changeable Input Parameters" section of the benchmark parameters properties file may be set to values other than default for a compliant run. These are marked gray in the table below. For a compliant run, the properties in the "Fixed Input Parameters" section of the properties file being used must not be changed from the values as provided by SPEC. All workload JVM instances must use the same parameters in a multi-JVM environment.

Parameter name	Compliant Value	Adjustable for a Compliant Run
calibration.interval_count	10 >= integer >= 3	Yes
calibration.length_seconds	240	No
ccs.enabled	true	No
director.connect_timeout	any	Yes
director.enabled	true	No
director.hostname	any	Yes
deterministic_random_seed	false	No
idle.length_seconds	240	No
idle.post_calibration	false	No
idle.post_run	true	No
idle.pre_calibration	false	No
idle.settle_seconds	0	No
include_file	any	Yes
load_level.count	10	No
load_level.delay_between	10	No
load_level.length_seconds	240	No
load_level.number_warehouses	availableProcessors (see 2.5.1)	Yes
load_level.percentage_sequence	none	No
load_level.post_measurement_seconds	30	No
load_level.pre_measurement_seconds	30	No
load_level.target_max_throughput	-1	No
load_level.throughput_sequence	none	No
log_level	INFO	No

orderlines_per_order	10	No
output_directory	any	Yes
override_itemtable_size	20000	No
power_meter.enabled	false	No
power_meter.hostname	any	Yes
power_meter.port	any	Yes
scheduler.batch_size	1000	No
scheduler.log_arrival_rates	false	No
scheduler.max_arrival_delay	10	No
scheduler.number_threads	availableProcessors / jvmInstances	No
scheduler.single_queue	false	No
screen_write	false	No
show_warehouse_detail	false	No
status.port	any	Yes
steady_state	true	No
suite	SPECpower_ssj	No
transaction_mix.cust_report	10	No
transaction_mix.delivery	1	No
transaction_mix.new_order	10	No
transaction_mix.order_status	1	No
transaction_mix.payment	10	No
transaction_mix.stock_level	1	No
transaction.response_time	None	No
warehouse_population	60	No

Table 2.5-1 Compliant Values for Benchmark Parameters

2.5.1 Warehouse Count and Override

The SPECpower_ssj2008 benchmark runs a fixed number of warehouses, N, equal to the number of logical processors in the system under test. This number is, by default, the value returned by the `java.lang.Runtime.getRuntime.availableProcessors` API. The value may be overridden by setting the `input.load_level.number_warehouses` property. Results for which the value of the `input.load_level.number_warehouses` property has been overridden must be submitted and reviewed by SPEC to determine compliance. An acceptable reason must be disclosed in the `config.sw.notes` section of the disclosure report. An example of an acceptable reason to override the default value would be if `System.availableProcessors()` does not return an accurate or valid value for the hardware architecture of the SUT. An example of an unacceptable reason would be to decrease the value of N from the default to hide scalability problems and artificially obtain a higher score. Future publications using an overridden `input.load_level.number_warehouses` do not require a review for this override unless the technical reason for setting the flag differs from what was previously accepted by the subcommittee.

2.5.2 Validity Checks

At the beginning of each run, the benchmark parameters are checked for conformance to the run rules. Warnings are displayed for non-compliant properties and printed in the final report; however, the benchmark will run to completion producing a report that is not valid for publication.

The following are required for a valid run and are automatically checked:

- Input Properties {InputPropertiesValidator}:
 - Verify that all input properties meet the criteria defined in section 2.5.
- Contains all required intervals as defined in section 2.3.2
- The measurement interval {IntervalLengthValidator} specified in the properties file must be 240 seconds. The actual measurement interval for each load point must be no less than 238.8 seconds (-0.5%) and no greater than 242.4 seconds (+1.0%). This rule allows for some variation in communicating the end of measurement to the threads.
- In order to ensure that the measurement interval for each JVM occurs during the time all instances are running {InstanceTimingValidator} the following requirement must be met:
 - No JVM can start the measurement interval before all JVMs have entered the pre-measurement interval.
 - No JVM can end the measurement interval after any JVM has completed the post-measurement interval.
 - Ensure that all measurement intervals fully overlap across all JVMs.

- Interval Length {IntervalLengthValidator}
 - Verify that the elapsed time for each measurement interval is at least 99.5% but no more than 101% of the configured interval length.
 - Note: For calibration interval 0, the upper boundary of the elapsed time can reach up to 105% for a valid result, since the calibration interval 0 is a warm-up phase and has no impact on setting actual load levels.
 - Results marked INVALID, because the elapsed time of Calibration 0 is up to 105%, can still be submitted to SPEC and the resulting INVALID message will be corrected to a WARNING message.
- Temperature {MinimumTemperatureValidator}
 - Verify that the minimum temperature reading is $\geq 20^{\circ}\text{C}$ (from the beginning of the run until the end of Active Idle -- including calibration).
- Overall Target Load Throughput {MultiVMTargetThroughputValidator}
 - Ensure that the combined throughput at each load level is within a limit
 - +2%, -2.5% for the 100% and 90% target loads
 - $\pm 2\%$ for the 80% though 10% target loads
- Per Host Target Load Throughput {MultiVMTargetThroughputValidator}
 - Ensure that the combined throughput on each host at each load level is within a limit
 - +2%, -2.5% for the 100% and 90% target loads
 - $\pm 2\%$ for the 80% though 10% target loads
- Set Throughput Variation {SetThroughputVariationValidator}
 - Ensure that the variation of the calibrated throughput of each node in a set is within a limit
 - 5% between the highest and lowest calibration throughput of a set
- Recompilation {NoRecompilationValidator}
 - Make sure that ssj.jar is first in the classpath, and verify that the code has not been recompiled.
- Power Analyzer {MeasurementDeviceValidator}
 - Ensure that a power analyzer was used, and that the SPEC PTDaemon marked it as a compliant device.
- Power Error Readings {PowerErrorReadingsValidator}
 - Validate the percentage of error readings from the power analyzer.
 - Threshold is 1% for Power and 2% for Volt rms, Ampere rms and Power Factor, measured only during measurement interval.
- Single Node Target Load Deviations {TotalThroughputDeviationValidator}
 - The sum of the throughputs on a single node at all load levels must be within 1% of the sum of all targets.
- Multi Node Target Load Deviations {TotalThroughputDeviationValidator}
 - The sum of the throughputs of all nodes at all load levels must be within 1% of the sum of all targets.
- Temperature Error Readings {TemperatureErrorReadingsValidator}
 - Validate the percentage of error readings from the temperature sensor. Threshold is 2%, measured from beginning of the run until the end of Active Idle, including calibration.
- Temperature Sensor {MeasurementDeviceValidator}
 - Ensure that a temperature sensor was used, and that the SPEC PTDaemon marked it as a compliant device.
- Power Uncertainty Validator {PowerUncertaintyValidator}
 - Per measurement interval (not calibration, nor ramp-up) per power analyzer
 - No more that 5% of all samples can have an uncertainty of greater than 1%
 - No more that 1% of all samples can have an unknown uncertainty
 - The average uncertainty of all samples cannot exceed 1%
- Single Set {HomogenousValidator}
 - Ensures that only one set is defined

2.6 Optimization Flags

Both JVMs and native compilers are capable of modifying their behavior based on flags. Flags which do not break conformance to section 2.12 are allowed. All command-line flags used must be reported. All flags used must be documented and supported within the time frame specified in this document for general availability. At the time a result is submitted to SPEC, descriptions of all flags used but not currently publicly documented must be available to SPEC for the review process. When the result is published, all flags used must be publicly documented, either in the vendor's public documentation, in the disclosure, or in a separate flags file.

2.7 Testbed Configuration

These requirements apply to all hardware and software components used in producing the benchmark result, including the System under Test (SUT), network, and controller.

- Any deviations from the standard default configuration for the testbed components must be documented so an independent party would be able to reproduce the configuration and the result without any further assistance.
- The controller system must be run on a physically different system than the SUT.
- There is no restriction on which machine the JVM director must run.

2.8 Line Voltage Source {VoltageValidator}

The preferred Line Voltage source used for measurements is the main AC power as provided by local utility companies. Power generated from other sources often has unwanted harmonics which are incapable of being measured correctly by many power analyzers, and thus would generate inaccurate results.

- The AC Line Voltage Source needs to meet the following characteristics:
 - Frequency: (50Hz or 60Hz) \pm 1%
 - Voltage in rms: (100V, 110V, 120V, 200V, 208V, 220V, 230V, 240V or 400V) \pm 5%

The usage of an uninterruptible power source (UPS) as the line voltage source is allowed, but the voltage output must be a pure sine-wave. For placement of the UPS, see 2.13.1. This usage must be specified in the Note section of the FDR.

Systems that are designed to be able to run normal operations without an external source of power cannot be used to produce valid results. Some examples of disallowed systems are notebook computers, hand-held computers/communication devices, and servers that are designed to frequently operate on integrated batteries without external power.

Systems with batteries intended to preserve operations during a temporary lapse of external power, or to maintain data integrity during an orderly shutdown when power is lost, can be used to produce valid benchmark results. For SUT components that have an integrated battery, the battery must be fully charged at the end of each of the measurement intervals described in clause 2.3.2, or proof must be provided that it is charged at least to the level of charge at the beginning of the interval.

Note that integrated batteries that are intended to maintain such things as durable cache in a storage controller can be assumed to remain fully charged. The above paragraph is intended to address "system" batteries that can provide primary power for the SUT.

If an unlisted AC line voltage source is used, a reference to the standard must be provided to SPEC. DC line voltage sources are currently not supported.

For situations in which the appropriate voltages are not provided by local utility companies (e.g. measuring a server in the United States which is configured for European markets, or measuring a server in a location where the local utility line voltage does not meet the required characteristics), an AC power source may be used, and the power source must be specified in the notes section of the disclosure report. In such a situation the following requirements must be met, and the relevant measurements or power source specifications disclosed in the general notes section of the disclosure report:

- Total Harmonic Distortion of source voltage (loaded), based on IEC standards: < 5%
- The AC Power Source needs to meet the frequency and voltage characteristics previously listed in this section.
- The AC Power Source must not manipulate its output in a way that would alter the power measurements compared to a measurement made using a compliant line voltage source without the power source.

The intent is that the AC power source does not interfere with measurements such as power factor by trying to adjust its output power to improve the power factor of the load.

2.9 Environmental Conditions

SPEC requires that power measurements 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 benchmark run.

SPECpower_ssj2008 requires the following environmental conditions to be met:

- Ambient temperature range: 20°C or above {MinimumTemperatureValidator}
- Elevation: within documented operating specification of SUT
- Humidity: within documented operating specification of SUT

2.9.1 Air Flow

Overtly directing air flow in the vicinity of the measured equipment to improve the benchmark score in a way that would be inconsistent with normal data center practices is not allowed.

2.10 General Availability

The entire testbed must be comprised of components that are generally available on or before date of publication, or must be generally available within **three** months of the first publication of these results.

Products are considered generally available if they are orderable by ordinary customers and ship within a reasonable time frame. This time frame is a function of the product size and classification and common practice. Some limited quantity of the product must have shipped on or before the close of the stated availability window. Shipped products do not have to match the tested configuration in terms of CPU count, memory size, and disk count or size, but the tested configuration must be available to ordinary customers. The availability of support and documentation of the products must be coincident with the release of the products.

Hardware products that are still supported by their original or primary vendor may be used if their original general availability date was within the last **five** years. The **five-year** limit is waived for hardware used in controller systems.

Software products that are still supported by their original or primary vendor may be used if their original general availability date was less than **3** years (for Java runtime environments) or **4** years (for all other software) prior to the availability date of the CPU family in use in the result. For Java runtime environments, original GA is defined as the release date (if documented publicly) or the date found in the Java version information output. For operating systems, original GA is generally defined as the release date of the service pack or minor version in use; specifically: for most operating systems, including RHEL, SLES, Ubuntu, Solaris, Mac OS, Debian, and BSD based versions, the minor version release date; for Windows, the base OS or service pack release date; for AIX, the technology level release date; for Fedora, the major version release date. A CPU family is defined as the group of CPUs from the same silicon vendor with the same architecture, socket, and brand name.

Information must be provided in the disclosure to identify any component that is no longer orderable by ordinary customers.

See <http://www.spec.org/osg/policy.html#AppendixC> – OSG Policy / Appendix C - Guidelines for General Availability

2.10.1 SUT Availability for Historical Systems

Please see OSG Policy section 2.3.5 on SUT Availability for Historical Systems

<http://www.spec.org/osg/policy.html#s2.3.5>

Also see section 3.3.4.3 of this document for proper declaration of a historical model.

2.11 System(s) Under Test (SUT)

The SUT may be a single stand-alone server or a multi-node set of servers as described below in the following sections. The SPECpower_ssj2008 benchmark metric *overall_ssj_ops/watt* applies to the entire SUT.

A multi-node SUT will consist of server nodes that cannot run independent of shared infrastructure such as a backplane, power-supplies, fans or other elements. These shared infrastructure systems are commonly known as “blade servers”.

Only identical servers are allowed in a multi-node SUT configuration; each must be identically configured. This requirement is for servers that execute the workload of the benchmark, and it does not include components that support these servers, e.g. storage-blades, controllers and shared appliances, which must be included in the power measurement.

All installed server-nodes must run the benchmark code, e.g. a multi-node SUT with 8 installed servers must run the benchmark code on all 8 nodes.

All software required to run the SPECpower_ssj2008 benchmark must be installed on and executed from a stable storage device which is considered part of the SUT.

Storage external to the enclosure is only allowed if no other means of storage is available, e.g. server internal storage, storage blade, or enclosure storage. The power consumption of this external storage must be measured as part of the SUT.

2.11.1 Electrical Equivalence

Many other SPEC benchmarks allow duplicate submissions for a single system sold under various names. Each SPECpower_ssj2008 result submitted to SPEC or made public must be for an actual run of the benchmark on the SUT named in the result. Electrically equivalent submissions are not allowed.

2.11.2 Hardware

Any hardware configuration of one or more systems and supporting components that is sufficient to install, start, and run the benchmark to completion in compliance with these run rules (including the availability requirements in section 2.10 and multi-system requirements in section 2.11) must be considered a compliant configuration. Any device configured at the time the benchmark is started must remain configured for the duration of the benchmark run. Devices which are configured but not needed for the benchmark (e.g. additional on-board NICs) may be disabled prior to the start of the benchmark run. Manual intervention to change the configuration state of components after the benchmark run has begun is not allowed.

External devices required for initial setup or maintenance of the SUT, but not required for normal operation or for running the benchmark (e.g. an external optical drive used for OS installation) may be removed prior to the benchmark being started.

If the model name or product number implies a specific hardware configuration, these specific components can not be removed from the hardware configuration but may be upgraded. Any upgrades are subject to the support, availability and reporting requirements of this document. For example, if the SUT is available from the vendor only with dual power supplies, both supplies must be installed and measured during the benchmark run. The power supplies may be upgraded if the vendor offers and supports such an upgrade, and the upgrade must be documented in the benchmark disclosure report.

For systems designated as a Server (see clause 3.3.2), a video monitor, if configured, may be powered by a separate power source and need not be included in the power measurement of the

SUT. For systems designated as a Personal System, a user display device and a user input device must be included in the power measurement and steps must be taken to ensure that the display device is actively displaying information for the duration of each measurement interval. All other configured devices must receive their power from the measured power source.

The components are required to be:

- specified using customer-recognizable names,
- documented and supported by the providing vendor, and
- of production quality.

Any tuning or deviation from the default installation or configuration of hardware components is allowed by available tools only and must be reported. This includes BIOS settings, power saving options in the system board management, or upgrade of default components. Only modifications that are documented and supported by the vendor(s) are allowed.

2.11.2.1 Network Interfaces

At least one port of the SUT's fastest network interface controller must be connected and operating at its full rated speed or 1Gb.

Automatically reducing network speed and power consumption in response to traffic levels is allowed for network interface controllers with such capabilities, as long as they are also capable of increasing to their full rated speed automatically.

2.11.3 Software

Required software components per server (host) are

- Exactly one single operating system (with one user space) including all modules that are installed during the installation process.
- A Java run time environment including one or more instances of a Java Virtual Machine (JVM).

Optional power management software, when installed, must be reported. The operating system must be in a state sufficient to execute a class of server applications larger than the benchmark alone. The majority of operating system services should remain enabled. Disabling operating system services may subject disclosures to additional scrutiny by the benchmark subcommittee and may cause the result to be found non-compliant. Any changes from the default state of the installed software must be disclosed in sufficient detail to enable the results to be reproduced. Examples of tuning information which must be documented include, but are not limited to:

- Description of System Tuning (includes any special OS parameters set, changes to standard daemons or services)
- List of Java parameters and flags used
- Any special per-JVM tuning for multi-JVM running (e.g. associating JVMs to specific processors)

These changes must be "generally available", i.e., available, supported and documented. For example, if a special tool is needed to change the OS state, it must be available to users and documented by the vendor.

The tester is expected to exercise due diligence regarding the reporting of tuning changes, to ensure that the disclosure correctly records the intended final product.

The software environment on the SUT is intended to be in a state where applications other than the benchmark could be supported. Disabling of operating system services is therefore discouraged but not explicitly prohibited. Disabled services must be disclosed.

The submitter/sponsor is responsible for justifying the disabling of service(s).

Services that must not be disabled include but are not limited to logging services such as cron or event logger.

A list of active operating system services may be required to be provided for SPEC's results review. The submitter is required to generate and keep this list for the duration of the review period. Such a list may be obtained, for example, by:

- Windows: *net start*
- Solaris 10: *svcs -a*
- Red Hat Linux: */sbin/runlevel; /sbin/chkconfig --list*

2.12 Java Specifications

Tested systems must provide an environment suitable for running typical server-side J2SE 5.0 (or higher) applications. Any tested system must include an implementation of the Java (tm) Virtual Machine as described by the following references, or as amended by SPEC for later Java versions:

- Java Virtual Machine Specification (second edition/ ISBN-13: 978-0201432947)

The following are specifically allowed, within the bounds of the Java Platform:

- Precompilation and on-disk storage of compiled executables are specifically allowed. However, support for dynamic loading is required. Additional rules are defined in section 2.12.2. See section 2.6 for details about allowable flags for compilation.

The system must include a complete implementation of those classes that are referenced by this benchmark as in the [J2SE 5.0 specification \(http://www.oracle.com/technetwork/java/javase/index-jsp-135232.html\)](http://www.oracle.com/technetwork/java/javase/index-jsp-135232.html). SPEC does not intend to check for implementation of APIs not used in this benchmark. For example, the benchmark does not use AWT (Abstract Window Toolkit, <http://download.oracle.com/javase/1.5.0/docs/guide/awt/index.html>), and SPEC does not intend to check for implementation of AWT. Note that the reporter does use AWT, however it is not necessary to run the reporter on the SUT.

2.12.1 Feedback Optimization and Precompilation

Feedback directed optimization and precompilation from the Java bytecodes are allowed, subject to the restrictions regarding benchmark-specific optimizations in section 1.1.2. Precompilation and feedback-optimization before the measured invocation of the benchmark are also allowed. Such optimizations must be fully disclosed.

2.12.2 Benchmark Binaries and Recompilation

The SPECpower_ssj2008 benchmark binaries are provided in jar files containing the Java classes. Valid runs must use the provided jar files and these files must not be updated or modified in any way. While the source code of the benchmark is provided for reference, the benchmarker must not recompile any of the provided .java files. Any runs that use recompiled class files are marked invalid and can not be reported or published.

2.13 Power and Temperature Measurement

The SPECpower_ssj2008 benchmark tool set provides the ability to automatically gather measurement data from accepted power analyzers and temperature sensors and integrate that data into the benchmark result. SPEC requires that the analyzers and sensors used in a submission be supported by the measurement framework, and be compliant with the specifications in this section.

2.13.1 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.

2.13.2 Power Analyzer Requirements

To ensure comparability and repeatability of power measurements, SPEC requires the following attributes for the power measurement device used during the benchmark. 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 discouraged.

- **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 rms 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 3 times the maximum amperage measured during any 1-second sample of the benchmark run. A minimum of 3 was chosen in order to align with EPA's Energy Star Program requirements.
- **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 1 set of measurements per second, where set is defined as watts and at least 2 of the following readings: volts, amps and power factor. The data averaging interval of the analyzer must be either 1 (preferred) or 2 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.

For example:

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

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

2.13.3 Temperature Sensor Requirements

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 benchmark:

- **Logging** - The sensor must have an interface that allows its measurements to be read by the benchmark harness. The reading rate supported by the sensor must be at least 4 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 benchmark run.

2.13.4 Supported and Compliant Devices

See Device List (http://www.spec.org/power/docs/SPECpower-Device_List.html) for a list of currently supported (by the benchmark software) and compliant (in requirements) power analyzers and temperature sensors.

2.13.5 Acceptance Process for New Measurement Devices

Adding a new measurement device to the SPEC power measurement framework includes three components:

- Providing documentation that the device meets the requirements of Section 2.13.2.
- Adding a new source code module to SPEC's Power and Temperature Daemon to allow the benchmark software to control the device.
- Performing tests with SPECpower_ssj2008 and SPEC tools to evaluate the actual behavior of the device.

Documentation to prove compliance with all required attributes must be provided. Publicly available documentation is preferred, but in special cases where a device vendor does not wish to disclose information perceived as proprietary, the device vendor may request its documentation remain SPEC Confidential.

For new device modules, all source code submitted to SPEC must include a signed SPEC Permission to Use Form (http://www.spec.org/spec/docs/permission_to_use.pdf) and must be freely available for use by other members and licensees of the benchmark. Supporting documentation must be provided as needed for the review. Once the code has been submitted, SPEC will then review the code. Barring any issues, SPEC will then incorporate the device module into a new version of the benchmark. Compliant runs must be done with SPEC provided binaries only.

The final step is testing of the device to verify that it meets the run rules requirements of section 2.13. The intent of this testing is to ensure that results obtained with the device are comparable to results obtained with other measurement devices.

SPEC provides a series of tests (see SPEC's Power Analyzer Acceptance Testing) that must be performed to determine power analyzer behavior under dynamic benchmark conditions. The preferred method of running these tests is to connect the new measurement device in series with another power analyzer that has already been accepted as compliant with the run rules requirements. These tests should be run by the submitter. In cases where the submitter does not have a currently-accepted power analyzer, a member of SPEC may volunteer to run those tests if a device is provided to them.

SPEC will review the test results against a set of criteria specified (see SPEC's Power Analyzer Acceptance Testing). If questions arise, SPEC may ask that additional testing be performed. Once a set of satisfactory results is produced, the device will be accepted as compliant and incorporated into the next release of the benchmark software.

Note: Since only SPEC-provided binaries may be used for compliant results, it is recommended that the device acceptance process be started well in advance of any benchmark use of a new device.

3 Reporting Rules

In order to publicly disclose SPECpower_ssj2008 results, the tester must adhere to these reporting rules in addition to having followed the run rules above. The goal of the reporting rules is to ensure the system under test is sufficiently documented so that someone could reproduce the test and its results and to ensure that the tester has complied with the run rules.

3.1 Reporting Metric and Result

SPECpower_ssj2008 expresses power and performance in the terms of overall ssj_ops/watt. Overall ssj_ops/watt represents the sum of the performance measured at each target load level (in ssj_ops) divided by the sum of the average power (in W) at each target load including active idle.

The report of results is an HTML file (*ssj.wxyz-main.html*) generated by the tools provided by SPEC. These tools must not be changed, except for portability reasons with prior SPEC approval. The tools perform error checking and will flag some error conditions as resulting in an "invalid run". However,

these automatic checks are only there for debugging convenience, and do not relieve the benchmarker of the responsibility to check the results and follow the run and reporting rules.

The section of the *ssj.wxyz.raw* file that contains actual test measurement must not be altered. Corrections to the SUT descriptions may be made as needed to produce a properly documented disclosure.

3.1.1 Publication

Any entity choosing to make statements using SPECpower_ssj2008 must follow the SPEC Fair Use Rule. Fair Use: Consistency and fairness are guiding principles for SPEC. To help assure that these principles are met, any organization or individual who makes public use of SPEC benchmark results must do so in accordance with the SPEC Fair Use Rule, as posted at (<http://www.spec.org/fairuse.html>).

3.1.1.1 Disclosure Requirement

Please see OSG Policy section 2.3.7 on Required Disclosure for Independently Published Results: (<http://www.spec.org/osg/policy.html#s2.3.7>).

3.1.2 Estimates

This rule, formerly present in this document, is now covered in SPEC Fair Use Rule (<http://www.spec.org/fairuse.html>).

3.1.3 Comparison to Other Benchmark Suites

This rule, formerly present in this document, is now covered in SPEC Fair Use Rule (<http://www.spec.org/fairuse.html>).

3.1.4 Addendum to OSG Fair Use Policy

This rule, formerly present in this document, is now covered in SPEC Fair Use Rule (<http://www.spec.org/fairuse.html>).

3.2 Reproducibility

SPEC is aware that power or performance results for pre-production systems may sometimes be subject to change, for example when a last-minute bug fix reduces the final performance.

If the sponsor becomes aware that the SPECpower_ssj2008 metric of a typical released system is more than 5% lower than that reported for the pre-release system, the tester is required to submit a new result for the production system, and the original result must be marked non-compliant (NC).

By submitting or publishing a benchmark disclosure (report) to SPEC, the test sponsor implicitly states that the system performance and power measured is representative of such systems. Power consumption is dependent on many factors that may vary over time within a specific vendor model. It can also vary from system to system due to well-known variability in electronic component fabrication processes.

3.3 Testbed Configuration Disclosure

The system configuration information that is required to reproduce published power and performance results must be reported. The principle is that if anything affects power or performance or is required to duplicate the results, it must be described. Any deviations from the standard, default configuration for the SUT must be documented so an independent party would be able to reproduce the result without any further assistance.

For the following configuration details, there is an entry in the configuration file, and a corresponding entry in the tool-generated HTML result page. If information needs to be included that does not fit into these entries, the Notes sections must be used.

3.3.1 General Availability Dates

The dates of general customer availability must be listed for the major Hardware components (`config.hw.available`) and server software (`config.sw.available`), by month and year. All the system, hardware and software features are required to be available within three months of the

first publication of these results. With multiple sub-components of the major components having different availability dates, the latest availability date must be listed for that major component. The benchmark Software components are not included in this date.

3.3.2 FDR Headline

The reporting page must list:

- The name of the organization or individual that sponsored the test. (`config.test.sponsor`).
- SPEC license number of that organization (`config.test.spec_license`)
- The test method "Single Node" or "Homogeneous Multi-Node" (`config.test.method`)
- Licensee which is reporting the results (`config.test.tested_by`)
- The name of the city, the state and country the test took place. (`config.test.location`).
- Date the test was performed, by month and year (automatically included)
- The System Source, either "Single Supplier" or "Parts Built" (`config.hw.system_source`)
 - Single Supplier
 - "Single Supplier" is defined as a SUT configuration where all hardware is provided by a single supplier.
 - Parts Built
 - "Parts Built" is defined as a SUT configuration where hardware is provided by multiple suppliers. A "Parts Built" system disclosure must include enough detail to procure and reproduce all aspects of the submission, including performance and power.
 - The usage of 3rd party switches (network or KVM) does not force a parts-built classification.
- System Designation "Server" or "Personal System" (`config.hw.system_designation`)
 - For the purposes of this designation, a 'Server' is defined as a computer system that is marketed to support multiple tasks from multiple users, simultaneously. A 'Personal System' is a computer system that is primarily marketed for use by a single individual, even though multiple tasks may execute simultaneously. (please see also 2.11.2)
 - The test sponsor must declare whether the system under test is a Server or a Personal System, depending on the primary use for the SUT and how the test sponsor intends it to be compared. In the case where a sponsor wishes to designate a computer system as both a server and a personal system, two separate benchmark measurements must be run and documented with separate designations.
- Power Provisioning "Line-powered" or "Battery-powered" - (`config.hw.power_provisioning`)

3.3.3 Benchmark Results Summary

The reporter automatically populates the Benchmark Result Summary.

For each Target Load:

- Performance: Target Load in %
- Performance: Actual Load in %
- Performance: Actual Load in ssj_ops
- Average Power in W
- Performance to Power Ration

Also a graphical representation of these values is automatically rendered.

3.3.3.1 Aggregate SUT Data

The reporter automatically populates the Aggregate SUT Data. In this section aggregated values for several system configuration parameters are reported. The section will be displayed only if more than one node is configured.

3.3.4 SUT

In this section hardware components common to all nodes will be described. The section will be displayed only if more than one node is configured.

3.3.4.1 Shared Hardware

A table including the description of the shared hardware components.

- Cabinet/Housing/Enclosure - The model name identifying the enclosure housing the tested nodes (`config.shared.enclosure`).
- Form Factor - The full SUT form factor (including all nodes and any shared hardware). (`config.shared.form_factor`). For rack-mounted systems, specify the number of rack units. For other types of enclosures, specify "Tower" or "Other".
- Power Supply Quantity and Rating (W) - The number of power supplies that are installed in the tested configuration (`config.shared.psu.installed`) and the power rating for each power supply (`config.shared.psu.rating`). Both values are set to 0 if there are no shared power supplies.
- Power Supply Details - The manufacturer of the power supply and the model number to identify it. (`config.shared.psu.description`) "N/A" if there are no shared power supplies.
- Network Switch - The number of network switches used to run the benchmark (`config.shared.network.switch`). "N/A" if there is no network switch.
- Network Switch Details - The manufacturer of the network switch and the model number to identify it (`config.shared.network.switch.description`). "N/A" if there is no network switch.
- KVM Switch - The number of KVM switches used to run the benchmark (`config.shared.kvm`) "N/A" if there is no KVM switch.
- KVM Switch Details - The manufacturer of the KVM switch and the model number to identify it. (`config.shared.kvm.description`) "N/A" if there is no KVM switch.
- Other Hardware - Any additional shared equipment added to improve performance and required to achieve the reported scores (`config.shared.other`).
- Comment - Description of additional performance or power relevant components not covered in the fields above (`config.shared.comment`)

Switches (network or KVM) are not included in the power measurement of the SUT for a multi-node configuration.

3.3.4.2 Set: 'N'

Detailed hardware and software description of the identically configured nodes which constitute this set.

- **Set Identifier** - A unique identifier for this set of nodes. (see (`SETID`) in `runssj.bat/runssj.sh`)
- **Set Description** - A textual description of this set of nodes, e.g. the model name of a blade server (`config.set.description`).
- **# of Identical Nodes** - The number of identically configured nodes which constitute this set. This number is read by the benchmark program from the "-numHosts" commandline parameter used to start the director code of the benchmark and reported here. (see (`NUM_HOSTS`) in `rundirector.bat/rundirector.sh`)
- **Comment** - Additional comments related to this set of nodes (`config.set.comment`).

3.3.4.3 SUT Hardware - single node(s)

The following SUT Hardware components must be reported:

- Hardware Vendor and Model (`config.hw.vendor`) and (`config.hw.model`)
 - For a historical system the declaration "(Historical)" must be added to the model
- Form Factor / [xU | Tower | Blade | Others] for SUT (`config.hw.form_factor`).
- CPU Name - (`config.hw.cpu`),
- CPU Characteristics - (`config.hw.cpu.characteristics`)
- CPU Frequency (MHz) - (`config.hw.cpu.mhz`)
- CPU(s) enabled - (`config.config.hw.cpu.cores`)
- the number of chips - (`config.config.hw.cpu.chips`)
- the number of cores per chip - (`config.config.hw.cpu.cores_per_chip`)

- the number of Hardware Threads / Core - (`config.hw.cpu.threads_per_core`)
- CPU(s) orderable - (`config.hw.cpu.orderable`)
- Primary Cache - (`config.hw.cache.primary`)
- Secondary Cache - (`config.hw.cache.secondary`)
- Tertiary Cache - (`config.hw.cache.tertiary`)
- Other Cache - (`config.hw.cache.other`).
- If a level of cache is shared among processors in a system that must be stated in the notes section of the disclosure.
- Memory Amount (GB) - (`config.hw.memory.gb`)
- # and size of DIMM(s) (`config.hw.memory.dimms`)
- Memory configuration - (`config.hw.memory.description`) if this is an end-user option that may affect the metric, e.g. interleaving and access time.
- Power Supply Quantity and Rating (W) - (`config.hw.psu.installed`)
 - “None” if shared by multiple nodes
- Power Rating for each power supply - (`config.hw.psu.rating`)
 - “None” if shared by multiple nodes
- Power Supply Details - (`config.hw.psu.description`)
 - “Shared” if shared by multiple nodes
- Number, type, model, and capacity of Disk Drive - (`config.hw.disk`)
- Number, type, Disk Controller model - (`config.hw.disk.controller`)
- Number and type of Network Interface Cards (NICs) Installed - (`config.hw.network.controller`).
- NICs Enabled in Firmware - (`config.hw.network.controller.enabled.firmware`)
- NICs Enabled in the OS - (`config.hw.network.controller.enabled.os`)
- NICs connected - (`config.hw.network.controller.connected`)
- Network Speed (Mbit) - (`config.hw.network.speed`)
- The connectivity to keyboard (`config.hw.keyboard`), mouse (`config.hw.mouse`) and monitor (`config.hw.monitor`) must be stated (KVM, USB, PS2, or none)
- Optical Drives - (`config.hw.optical`)
- Other Hardware, e.g. write caches, or other accelerators - (`config.hw.other`)

Please note that the method that started the benchmark needs to be disclosed in the “System under Test Notes” when no keyboard was used (e.g. Run was started via Remote Desktop).

3.3.4.4 SUT Software

The following SUT software components must be reported:

- All Power Management Options used - (`config.sw.power_management`)
- The Operating System (OS) name and version - (`config.sw.os`)
- Type of Filesystem – (`config.sw.filesystem`)
- Java Virtual Machine (JVM):
 - Vendor (`config.sw.jvm.vendor`)
 - Version (`config.sw.jvm.version`)
 - Command-line Options (`config.sw.jvm.options`)
 - Affinity settings (`config.sw.jvm.affinity`)
 - Instances (automatically supported)
 - Initial Heap (MB) (`config.sw.jvm.heap.initial`)
 - Maximum Heap (MB) (`config.sw.jvm.heap.max`)
 - Address Bits (`config.sw.jvm.bitness`).
- The version (`config.sw.boot_firmware.version`) and settings (`config.sw.boot_firmware.settings`) of the Boot Firmware.
- The version (`config.sw.mgmt_firmware.version`) and settings (`config.sw.mgmt_firmware.settings`) of the Management Firmware.
- Benchmark Version (automatically supported)

- The location of the JVM director (`config.director.location`).
- Other Software, e.g. management components - (`config.sw.other`)
- Any other software packages used during the benchmarking process.
- Other clarifying information as required to reproduce benchmark results; e.g. non-default kernel parameters, must be stated in the notes section of the disclosure.
- Additionally, the submitter must be prepared to make available a description of the tuning features that were utilized; e.g. kernel parameters and software settings, including the purpose of that tuning feature. Where possible, it must be noted how the values used differ from the default settings for that tuning feature.

3.3.4.5 System Under Test Notes

The System Under Test Notes (`config.sut.notes`) section is used to document:

- System tuning parameters other than default.
- Process tuning parameters other than default.
- Background load, if any.
- Critical customer-identifiable firmware or option versions such as network and disk controllers.
- Definitions of tuning parameters must be included or a pointer supplied to a separate document hosted by SPEC.
- Part numbers or sufficient information that would allow the end user to order the SUT configuration.
- Identification of any components used that are supported but that are no longer orderable by ordinary customers.
- The presence of an integrated battery (or batteries) that can provide primary power for the SUT.
- Additional information may be required. If:
 - the hardware vendor offers additional configuration options or features with the identical model name to this SUT,
 - and the options or features of those configurations are publicly documented,
 - and the options or features of this SUT are not publicly documented,
 - then any significant differences in capabilities or features of this SUT not already disclosed elsewhere in the FDR must be documented in this section.

3.3.5 Controller System

The following properties must be reported:

- Controller System - Hardware Vendor - (`ccs.config.hw.vendor`)
- Controller System - Hardware Model - (`ccs.config.hw.model`)
- Controller System CPU description - (`ccs.config.hw.cpu`)
- Controller System Total memory amount - (`ccs.config.hw.memory.gb`)
- Controller System OS Type and Version - (`ccs.config.sw.os`)
- Controller System JVM Vendor (`ccs.config.sw.jvm.vendor`)
- Controller System JVM - (`ccs.config.sw.jvm.version`)
- CCS Version - (automatically supported)

3.3.5.1 Power Analyzer and Temperature Sensor

The following properties must be reported:

- Power Analyzer Vendor - (`ptd.pwrN.config.analyzer.vendor`)
- Power Analyzer Model - (`ptd.pwrN.config.analyzer.model`)
 - Power Analyzer Serial number - (`ptd.pwrN.config.analyzer.serial`)
 - The connectivity to the Power Analyzer (RS-232 (serial port), USB, GPIB etc.) - (`ptd.pwrN.config.analyzer.connectivity`)
 - Input connection to the Power Analyzer - (`ptd.pwrN.config.analyzer.input_connection`).
 - Calibration of the Power Analyzer:

- Institute (`ptd.pwrN.config.calibration.institute`)
- Accredited by (`ptd.pwrN.config.calibration.accredited_by`)
- Calibration label (`ptd.pwrN.config.calibration.label`).
- Calibration date (`ptd.pwrN.config.calibration.date`).
- Power analyzer voltage and current range in rms (see `ccs.props`)
- Since auto-ranging is discouraged, if auto-ranging is used, an explanation of the reason must be provided. Also, the rms ranges used by the analyzer during auto-ranging must be readable by the SPEC PTDaemon in order to ensure that an uncertainty calculation can be made.
- PTDaemon host system, OS - (`ptd.pwrN.config.ptd.system`), (`ptd.pwrN.config.ptd.os`)
 - PTDaemon Version - (automatically supported)
 - Setup Description - (`ptd.pwrN.config.analyzer.setup_description`)
- Temperature Sensor Vendor (`ptd.tempN.config.sensor.vendor`)
- Temperature Sensor Model (`ptd.tempN.config.sensor.model`)
 - Driver version - (`ptd.tempN.config.sensor.driver`)
 - The connectivity to the Temperature Sensor - (`ptd.tempN.config.sensor.connectivity`)
 - PTDaemon host system, OS - (`ptd.tempN.config.ptd.os`)
 - Setup Description - (`ptd.tempN.config.sensor.setup_description`)

3.3.6 Disclosure Notes

The Notes (`config.notes`) section is used to document:

- Additional important information required to reproduce the results from other reporting sections that require a larger text area.

3.3.7 Electrical and Environmental Data

The reporter automatically populates (values from measurements) the following table entries.

For each Target Load:

- Average Voltage (V rms)
- Average Current (A rms)
- Average Power (W)
- Minimum Ambient Temperature (°C)

The reporter also automatically populates measured values for the Minimum Temperature (°C).

- The Line Standard must be reported manually.
 - Voltage rms - (`config.line.standard.voltage`)
 - Frequency - (`config.line.standard.frequency`)
 - Number of Phase(s) - (`config.line.standard.phase`)
 - Number of Wires - (`config.line.standard.wires`)
- The Elevation must be reported with an accuracy of 50m or better - (`config.test.elevation`)
- The Humidity needs not be reported.

4 Submission Requirements for SPECpower_ssj2008

When a potentially-compliant run is completed and acceptance by SPEC is desired, the raw results file must be submitted. The required file should be e-mailed to SPEC as an attachment. The committee may request additional benchmark output files from the submitter as well. The submitter should be prepared to participate in discussion during the review cycle and at the subcommittee meeting in which the result is voted on for final acceptance, to answer any questions raised about the result. The submitter is also required to keep the log files for the SUT and Controller from the run for the duration of the review cycle and make them available upon request. Licensees of the benchmark wishing to

submit results for acceptance may be required to pay a fee. The complete submission process is documented in “Submitting OSG Benchmark Results to SPEC”.

(http://www.spec.org/osg/submitting_results.html). Please ensure that the latest SPEC PTDaemon was used before submitting a result in order to prevent potential complication during the review process. (http://www.spec.org/power/docs/SPECpower-Device_List.html)

5 SPECpower_ssj2008 Benchmark Kit Overview

The benchmark kit includes tools for running the benchmark and reporting its results. The workload and CCS components are written in Java; precompiled class files are included with the kit, so no build step is necessary. This software implements various checks for conformance with these run and reporting rules, therefore the SPEC software must be used.

Any new SPEC PTDaemon device modules will be evaluated by the sub-committee according to the acceptance process (see section 2.13.5). Once the code is accepted by the sub-committee, it will be made available for any licensee to use in their measurements and submissions.

5.1 Documents overview

The benchmark related documents (Run and Reporting Rules, User Guide, Measurement Setup Guide, Design Document, Methodology, FAQ, etc.) can be found as part of the benchmark distribution.

For the latest versions, please consult SPEC’s website (http://www.spec.org/power_ssj2008/).

5.2 Trademark

Product and service names mentioned herein may be the trademarks of their respective owners.