

SPECpower 委員会

サーバー効率評価ツール (Server Efficiency Rating Tool)  
(SERT) <sup>TM</sup>

設計文書  
第 3 公開草案

Standard Performance Evaluation Corporation

## 目次

<b>1</b>	<b>序論</b> .....	<b>6</b>
1.1	概要.....	6
1.2	SPEC について.....	6
1.2.1	SPEC 会員資格.....	6
1.2.2	SPEC 一般策定指針.....	7
1.3	EPA の ENERGY STAR コンピュータサーバー基準と SPEC.....	7
1.4	従来型ベンチマークと SERT の違い.....	7
1.5	意見募集方法.....	8
1.6	文書管理.....	8
1.7	商標.....	8
<b>2</b>	<b>対象範囲と目的</b> .....	<b>9</b>
2.1	概要.....	9
2.2	ソケットとノード.....	9
2.3	拡大縮小能力 (スケーリング).....	9
2.4	サーバーのオプションと拡張能力.....	10
2.5	IO 構成装置.....	10
2.5.1	ストレージ IO.....	10
2.5.2	ネットワーク IO.....	10
2.6	冗長性.....	11
2.7	実行時間.....	11
2.8	プラットフォーム.....	11
2.8.1	出荷時構成による試験.....	11
2.9	実行言語.....	11
2.10	負荷水準.....	11
2.11	小作業負荷 (worklet).....	12
2.12	作業負荷 (workload).....	12
2.13	暫定試験スケジュール.....	13
2.14	スケジュールの妥協点.....	13
<b>3</b>	<b>SERT 基本設計概念</b> .....	<b>14</b>
3.1	システム概要.....	14
3.2	SERT の実行.....	15
<b>4</b>	<b>小作業負荷の実行段階</b> .....	<b>16</b>
<b>5</b>	<b>小作業負荷の設計指標</b> .....	<b>19</b>
5.1	有効アイドル小作業負荷.....	19
5.2	CPU 小作業負荷.....	19
5.3	メモリ小作業負荷.....	19
5.4	ネットワーク IO 小作業負荷.....	20
5.5	ストレージ IO 小作業負荷.....	20
5.6	システム小作業負荷.....	20
<b>6</b>	<b>消費電力と温度の測定</b> .....	<b>21</b>
6.1	環境条件.....	21
6.2	温度センサー仕様.....	21
6.3	消費電力測定器の要件.....	21
6.4	SPEC PTDaemon.....	22

6.5	対応および準拠装置.....	22
6.6	消費電力測定器の設置.....	22
6.7	直流線間電圧.....	22
<b>7</b>	<b>指標(metric)／成績(score)、報告、ロギング.....</b>	<b>23</b>
7.1	指標／成績.....	23
7.2	報告と出力のファイル.....	23
7.2.1	報告1:「概要報告」.....	23
7.2.2	報告2:「消費電力と性能のデータシート」.....	24
7.3	妥当性確認 (Validation) ／実証 (Verification).....	24
7.4	ロギング.....	25
<b>8</b>	<b>将来的な強化／拡大目標.....</b>	<b>26</b>
8.1	グラフィカルユーザーインターフェース (GUI).....	26
8.2	試験ソフトウェア.....	26
<b>9</b>	<b>SERT および EPA の ENERGY STAR サーバー基準バージョン 2.0.....</b>	<b>27</b>
9.1	測定.....	27
9.2	SERT バイナリおよび再コンパイル.....	27
9.3	手動介入.....	27
9.4	SERT 情報の公正使用.....	27
9.4.1	公正使用規則.....	27
9.5	認定独立試験所.....	27
9.6	供給電圧公差.....	28
<b>10</b>	<b>小作業負荷候補.....</b>	<b>29</b>
10.1	CPU 小作業負荷: Compress.....	30
10.1.1	概要.....	30
10.1.2	シーケンス実行方法.....	30
10.1.3	指標.....	30
10.1.4	所要初期化.....	30
10.1.5	構成設定値.....	30
10.1.6	トランザクションコード.....	30
10.2	CPU 小作業負荷: CryptoAES.....	31
10.2.1	概要.....	31
10.2.2	シーケンス実行方法.....	31
10.2.3	指標.....	31
10.2.4	所要初期化.....	31
10.2.5	構成設定値.....	31
10.2.6	トランザクションコード.....	31
10.3	CPU 小作業負荷: FFT.....	32
10.3.1	概要.....	32
10.3.2	シーケンス実行方法.....	32
10.3.3	指標.....	32
10.3.4	所要初期化.....	32
10.3.5	構成設定値.....	32
10.3.6	トランザクションコード.....	32
10.4	CPU 作業負荷: LU.....	33
10.4.1	概要.....	33
10.4.2	シーケンス実行方法.....	33

10.4.3	指標.....	33
10.4.4	所要初期化.....	33
10.4.5	構成設定値.....	33
10.4.6	トランザクションコード.....	33
10.5	CPU 作業負荷：SOR.....	34
10.5.1	概要.....	34
10.5.2	シーケンス実行方法.....	34
10.5.3	指標.....	34
10.5.4	所要初期化.....	34
10.5.5	構成設定値.....	34
10.5.6	トランザクションコード.....	34
10.6	CPU 作業負荷：XmlValidate.....	35
10.6.1	概要.....	35
10.6.2	シーケンス実行方法.....	35
10.6.3	指標.....	35
10.6.4	所要初期化.....	35
10.6.5	構成設定値.....	35
10.6.6	トランザクションコード.....	35
10.7	メモリ小作業負荷：Flood.....	36
10.7.1	概要.....	36
10.7.2	シーケンス実行方法.....	36
10.7.3	指標.....	37
10.7.4	所要初期化.....	37
10.7.5	構成設定値.....	37
10.7.6	トランザクションコード.....	37
10.8	メモリ作業負荷：XmlValidate.....	38
10.8.1	概要.....	38
10.8.2	シーケンス実行方法.....	38
10.8.3	指標.....	38
10.8.4	所要初期化.....	38
10.8.5	構成設定値.....	38
10.8.6	トランザクションコード.....	39
10.9	ストレージ IO 作業負荷.....	40
10.9.1	概要.....	40
10.9.2	シーケンス実行方法.....	40
10.9.3	指標.....	40
10.9.4	所要初期化.....	40
10.9.5	構成設定値.....	40
10.9.6	トランザクションコード 1-RandomRead.....	41
10.9.7	トランザクションコード 1-RandomWrite.....	41
10.9.8	トランザクションコード 2-SequentialRead.....	41
10.9.9	トランザクションコード 2-SequentialWrite.....	41
10.10	システム小作業負荷：CSSJ.....	42
10.10.1	概要.....	42
10.10.2	シーケンス実行方法.....	42
10.10.3	指標.....	42

10.10.4	所要初期化.....	42
10.10.5	構成設定値.....	42
10.10.6	新規注文トランザクション .....	42
10.10.7	支払トランザクション.....	43
10.10.8	注文状況トランザクション .....	44
10.10.9	配達トランザクション.....	44
10.10.10	在庫水準トランザクション .....	44
10.10.11	顧客報告トランザクション .....	44

## 1 序論

### 1.1 概要

EPAのENERGY STAR策定チームは、現在コンピュータサーバー基準バージョン 2.0 について作業を進めている<sup>1</sup>。バージョン 2.0 は、稼働モード効率評価ツール (Active Mode Efficiency Rating Tool) を介して、実際の演算作業を実行している間のサーバーの全体効率を測定する方法を追加し、本プログラムを発展させることを目的としている。

SPECpower委員会は現在、コンピュータサーバーのエネルギー効率を測定し評価する次世代ツールである、サーバー効率評価ツール (SERT : Server Efficiency Rating Tool) <sup>TM</sup>の設計、実施および配信に取り組んでいる。この公開草案は、EPAの関係者および協力者による検討用に、SERTの設計を説明するものである。

最新情報については、[http://www.spec.org/sert/docs/SERT-Design\\_Doc.pdf](http://www.spec.org/sert/docs/SERT-Design_Doc.pdf)にて確認してほしい。

### 1.2 SPEC について

Standard Performance Evaluation Corporation (SPEC) は、演算性能を測定するための業界規格を策定するために、1988 年に業界により設立された。その後 SPEC は、最大かつ最も影響力のある世界的なベンチマーク共同事業者に成長した。この団体の目的は、最新世代の IT 機器を分析するための公正かつ有用な指標が、確実に市場に存在するようにすることである。

SPEC communityは、様々な応用分野におけるシステム性能評価のために、30 を超える業界標準のベンチマークを策定し、世界中の企業、情報センター、および教育機関に多くのベンチマーク利用許可を与えてきた。これらベンチマークを利用する組織は、20,000 を超える専門家検証性能報告書 (peer-reviewed performance report) をSPECのウェブサイトに公開している (<http://www.spec.org/results.html>)。

SPEC は、様々な業界部門における業界標準のコンピュータシステム性能ベンチマークの設計、策定と公表、専門家によるベンチマーク実行結果の検証に長く従事してきた。性能のベンチマーク評価や、新たなベンチマークの策定および公表に必要な業務は、参加者の間に不一致をもたらす可能性がある。そのため SPEC は、多様かつ競合する組織間の協力や公平性を推奨する運用指針と、様々な規範的行動を策定した。

エネルギー効率 IT 機器の需要増加により、消費電力と性能のベンチマークの必要性が生じている。これに応じて、SPEC community は、消費電力/消費電力量の測定により既存の業界標準ベンチマークを増補する新たな取り組みとなる、SPECpower を設立した。ベンチマーク策定とエネルギー消費効率分野の優れた技術者と科学者が、この業務に取り組むことを公約した。サーバー級の演算機器の消費電力と性能特性を測定する、最初の業界標準ベンチマークの策定は、2006 年 1 月 26 日に始まった。2007 年 12 月には、CPU、キャッシュ、メモリ階層、および複数負荷水準における共用メモリプロセッサの拡張機能を働かせる SPECpower\_ssj2008 が公表された。このベンチマークは、多種多様なオペレーティングシステムやハードウェア上で実行することができる。2009 年 4 月 15 日に公表されたバージョン 1.10 において、SPEC は、複数ノード対応 (例：ブレード対応) を追加し、SPECpower\_ssj2008 を増補した。

#### 1.2.1 SPEC 会員資格

SPECへの入会は、関心を有するあらゆる企業または事業者に対して開かれている。OSG会員および準会員は、公表されたOSGベンチマークすべての複製許可と、SPEC公開ウェブサイトへの無制限の結果公表について権利が与えられる。会員には、入会金および年会費の支払いが求められる。非営利団体や教育機関には、年間費割引料金体系が適用される。会員資格情報の追加詳細は、<http://www.spec.org/osg/joining.html>で見ることができ、[info@spec.org](mailto:info@spec.org) に請求することができる。また、現在の SPEC 会員一覧は、<http://www.spec.org/spec/membership.html>で見ることができる。

<sup>1</sup> 米国環境保護庁 (US Environmental Protection Agency) - コンピュータサーバーのENERGY STARプログラム要件。  
[http://www.energystar.gov/index.cfm?c=revisions.computer\\_servers](http://www.energystar.gov/index.cfm?c=revisions.computer_servers)

## 1.2.2 SPEC 一般策定指針

SPEC の信条と参加基準が、SERT 策定の基礎となっている。本ツールは、多種多様な競合する企業を代表する委員が協力して策定している。以下の項目は、業界にとって有用であり広く採用されるツールの策定において、委員の指針となる。

- 決定は全会一致により達成される。動議の可決には、特定多数決が必要とされる。
- 決定は事実に基づく。実験による結果は、意見よりも重要性を有する。データおよび実証は断定より優る。
- 公正なベンチマークは、透明性のある市場において、すべての業界参加者間の競争を可能にする。
- ツールおよびベンチマークは、アーキテクチャに中立的であり移植可能である。
- 貢献する意思のある者すべてが参加することができる。様々な範囲の利用可能なソリューションについて結果が広く入手できることにより、最終使用者は、適切な IT 機器を判断することができる。

同様の指針は、性能と消費電力/性能の業界における SPEC ベンチマークの成功と普及という結果をもたらしており、SERT の成功にとっても不可欠である。

## 1.3 EPA の ENERGY STAR コンピュータサーバー基準と SPEC

SPEC は、IT 機器のさらなるエネルギー消費効率の向上という EPA の目標を称賛し、EPA の ENERGY STAR プログラムのことを、本取り組みにおける業界パートナーと考えている。稼働モード効率評価ツール (Active Mode Efficiency Rating Tool) の策定は、世界規模の消費電力量削減のための継続的な取り組みにおいて不可欠な要素であり、社会的成功を収めた ENERGY STAR が、世界中のエネルギー効率化プログラムを調和させる潜在性を持ったコンピュータサーバープログラムとなる道筋を与える。

SPEC は、コンピュータサーバーの ENERGY STAR 基準を支持し、SERT について EPA と協力して取り組む機会を歓迎し、また EPA の ENERGY STAR 基準策定チームとの長期にわたる関係が継続することを期待している。

## 1.4 従来型ベンチマークと SERT の違い

性能ベンチマークとエネルギー消費効率ベンチマークは、特定の業務モデルまたは応用分野におけるコンピュータサーバーの能力を中心に扱う傾向がある。SERT は、広範なアプリケーション環境にわたるエネルギー消費効率の第 1 次近似<sup>2</sup>を提供することに重点を置いている。

- 絶対的成績 (スコア) は、特定の応用能力を反映していないため、最終使用者にとってはあまり適切ではない。
- 合否または [レベル 1/レベル 2/レベル 3] 合否評価をもたらす評価ツールは、指標に複数桁の精度を伴う一般的なベンチマークの結果よりも、コンピュータサーバーに関する EPA の ENERGY STAR 環境に適している。
- 本プログラムに対するより多くの参加を促すために、絶対的成績 (スコア) の販促利用は認められない。

ベンチマークは、ベンチマークビジネスモデルの用途に合わせて構成とソフトウェアを特注対応にする選択肢の調整を含めて、最適な状態を重視する傾向がある。優位なベンチマーク結果を得る必要があるために、ベンチマーク処理に多額の投資を行うことが頻繁に見られる。SERT は、以下の方法によって最低限の機器と技能を必要とする、経済的で利用が容易なツールになるように設計されている。

- 高度に自動化された処理過程と既存の SPEC 方法の利用
- サーバーの出荷時における初期設定の重視
- 過度な調整を含まない

ベンチマークが固定基準点を表している場合において、ENERGY STAR プログラムは、業界が発展するとともに厳しくなる基準値を利用し、継続的な改善を促すように設計されている。SERT は、以下の内容を含めて、この論理的枠組みに合うように設計されている。

- 新たな演算技術の早期採用
- SERT バージョン更新の迅速な実施

<sup>2</sup> Andrew Fanara, Evan Haines, Arthur Howard

[http://www.energystar.gov/ia/partners/prod\\_development/downloads/State\\_of\\_Energy\\_and\\_Performance\\_Benchmarking\\_for\\_Enterprise\\_Servers\\_Final.pdf](http://www.energystar.gov/ia/partners/prod_development/downloads/State_of_Energy_and_Performance_Benchmarking_for_Enterprise_Servers_Final.pdf)

## 1.5 意見募集方法

SERT 策定班は、EPA の ENERGY STAR プログラムとのパートナーシップを活用し、策定過程の全体を通して広範囲にわたる業界専門家から得られた意見を評価する予定である。本班は、EPA のバージョン 2.0 の枠組みにおける作業負荷、指標およびその他すべての要件について協調して取り組む予定である。

貴方の詳細な意見を [servers@energystar.gov](mailto:servers@energystar.gov) を通じて EPA に提供してほしい。EPA は提供された意見を収集、分類、匿名化、そして優先順位を付けて、SPEC 策定班に提供する。

## 1.6 文書管理

SERT に適用される、使用許可取得者や価格構成、更には支援と保守の規範は作成中である。

## 1.7 商標

本書において言及される製品およびサービスの名称は、各所有者の商標の可能性がある。



## 2 対象範囲と目的

コンピュータサーバーの ENERGY STAR 基準バージョン 2.0 の現在の対象範囲には、1~4 プロセッサソケットを有するサーバーが含まれており、同様の範囲に入るブレード技術を含めるよう拡大するという目標が公言されている。SERT の設計目標は、これら技術と更に大型の技術に対応することである。

大型システムへの対応に関する課題の中には、作業を完了するためのシステム全体の能力や、付加的なプロセッサ、メモリ、ネットワークインターフェースカード、ディスクドライブ等の追加に伴い増減する作業負荷を設計する能力がある。様々な作業負荷の特徴が、これら構成装置それぞれについて有効性を実証するために必要とされる。構成装置が存在していないことを不当に示さないようにしながら、その存在を正しく表す作業負荷を提供することは、難しい課題である。これらの問題は、小型サーバーよりも、多くの拡張機能を有する大型システムにおいて多く見られる。

消費電力要件に対する構成装置の価値が適切に表されないと結論付けられた分野については、「構成の消費電力/性能変更因子 (configuration power/performance modifier)」の追加に対応するように、本ツールを設計する予定である。設計目標は、変更因子が追加されたという詳細な記録を含む追加情報が、ENERGY STAR 適合結果の評価に自動的に含まれるようにすることである。

### 2.1 概要

以下の表には、SERT が提供を予定する、あるいは予定していない、設計目標の一部が集約されている。

設計目標	非設計目標
総エネルギー消費効率の評価ツール	ベンチマークあるいは能力計画ツール
消費電力、性能および吸気温度の測定ツール	気流、気圧、排気温度の測定ツール
一般演算環境評価基準	特定用途ベンチマーク評価基準
交流給電サーバー対応	直流給電サーバー対応
サーバー環境に基づいた単一 OS インスタンスにおける使用	仮想ハイパーバイザー技術 <sup>3</sup> の強調
ENERGY STAR 評価ツール	販促用ツール
アーキテクチャおよび OS に対する中立性	開発、試験、および試験を遂行するために任意で提供された情報が不十分であるアーキテクチャおよび OS 環境における実施

### 2.2 ソケットとノード

SERT 1.0.0.0 は増減可能に設計されており、(同種のサーバーまたはブレードサーバーに限定して) 最大 8 ソケットおよび最大 64 ノードまで試験する予定である。試験されるサーバー (SUT : server under test) は、単一の独立型サーバーまたは複数ノードサーバーである。複数ノード SUT は、バックプレーン、電源装置、送風機または他の要素のような共用設備から独立して動作できないサーバーノードで構成されている。これら共用設備システムは、一般に「ブレードサーバー」または「複数ノードサーバー」として知られている。同一サーバーのみが複数ノード SUT 構成において認められる。

### 2.3 拡大縮小能力(スケーリング)

任意のサーバーのサーバー効率評価が SERT の主要目的であるため、SERT の主要設計目標の一つは、システム構成に比例して、そのシステムの性能を増減できるようにすることである。より多くの構成装置 (プロセッサ、メモリ、およびディスクストレージ) がサーバーに追加されるにつれて、小構成の同じサーバーの性能と比較したときに結果として得られる性能が大きくなるように、作業負荷は追加資源を利用すべきである。同様に、任意のサーバーについて構成装置がより高速な同一装置にアップグレードされた場合も、その性能は、それに応じて向上すべきである。構成装置の追加やアップグレードは一般的に、サーバーの総合効率結果に影響を及ぼす総消費電力を増加させることから、本件は本ツールの非常に重要な点である。CPU の数/速度に基づき性能を増減させるツールの策定は、最も容易に達成可能であるが、他の構成装置の場合には、このようなツールの実施の複雑度は大きく増大する。

<sup>3</sup> 仮想化は、消費電力量削減の重要な手段になる可能性がある。SERT のような一次近似ツールにおいて、仮想化された環境による影響は、より大きな負荷水準における結果を調査することにより判断することができる。

SERT は、SUT の追加ハードウェア資源に応じて性能が増減するように設計されるが、追加ハードウェアに関係ないシステム構成装置において性能障害がある場合には、SUT 自体が高い性能を維持できない可能性がある。このような場合、SUT に対する構成装置の追加は通常、これに比例した性能の向上が無いまま、消費電力が大きくなる。また、小型システム用の作業負荷一式は、大型システムを評価する場合において適切に増大しない可能性がある。

## 2.4 サーバーのオプションと拡張能力

サーバーは、応用の幅を広げるように設計されている多くの任意特性を有する可能性がある。これら特性は、追加消費電力の要因となるだけでなく、電源装置と冷却システムに多くの容量を必要とする。一部の SERT 作業負荷の構成要素は、これら特性が提供する強化能力を実証するように設計される予定である。しかし、本ツールは、対応する拡張作業負荷に対するこれら能力を認める必要がある一方で、実質的な拡張の選択肢が設計されていない高効率サーバーを不利にすることはできない。強化された構成に対して強化された評価を提供することと、すべての状況において不必要と思われる特性の単なる追加によるサーバーの容易な適合を防ぐことを、両立しなければならない。

SERT の目標は、サーバーに大きな拡張性があることを評価する一方で、低拡張性に設計されているサーバーを不必要に不利にすることを回避することである。例えば、PCI スロットに I/O アダプタが 4 つ存在する構成は、そのようなアダプタが 1 つしかない構成よりも効率的に本ツールの作業負荷を実行する可能性がある。その一方で、ネットワークアダプタが 2 つある構成と同程度にしか効率的に本ツールの作業負荷を実行しないかもしれない。4 つのアダプタを有する構成は、アダプタが 2 つだけの構成よりも実際の作業負荷を効率的に実行する可能性があるために、EPA は、追加 PCI スロット対応に要する電力設備に控除を与えるために、ある種の「構成変更因子 (configuration modifier)」を考慮することを選ぶ可能性がある。

「構成の消費電力ー性能変更因子 (configuration power-performance modifier)」の控除が含まれる場合、本ツールは、これに確実に対応するように設計され試験される予定である。

## 2.5 IO 構成装置

ディスクおよびネットワーク IO 構成装置は、CPU 中心の試験よりも、システムの性能と消費電力の適切な包括的状況を提供することが強く求められている。SPEC は、SERT 用 IO 作業負荷評価の初期段階にあるために、本節では、必ずしも結論ではなく、多くの検討課題が提供されている。

SPEC は、下記の 2 つの節における課題の一部について、有意義な形で試験または測定することが合理的または実際的でない可能性があることを認識している。これらの場合において、我々は、追加機能に関連して引き込まれる追加消費電力を補うために、「構成の消費電力ー性能変更因子 (configuration power-performance modifier)」の使用を提案する。検討中の他の課題には以下のものが含まれる。

- 異なるサーバー区分に対する異なる種類/数量の IO
- ディスクおよびネットワークサブシステムの自己校正性能測定値

### 2.5.1 ストレージ IO

理想的には、SERT のストレージ IO 構成装置には以下のものを認める。

- 高性能ストレージサブシステム
- 大容量ストレージサブシステム
- 信頼性および可用性特性 (RAID、バッテリー付キャッシュ等)

### 2.5.2 ネットワーク IO

理想的には、SERT のネットワーク IO 構成装置には以下のものを認める。

- 高性能ネットワークインターフェース
- 高速転送ネットワークインターフェース:
- 信頼性および可用性特性

## 2.6 冗長性

多くのサーバーは、電源装置および冷却送風機に冗長性が組込まれている。一部のサーバーには、メモリ、ディスクおよびプロセッサについても異なる水準の冗長性が含まれている。設計目標は、冗長構成装置への対応能力を含めることであるが、冗長構成装置の一つが不能になったときの障害耐性状態における消費電力量を測定する具体的試験は予定されていない。

## 2.7 実行時間

結果の高い再現性、広範なサブシステム対象範囲、および低度の資源配分に適正なバランスを取ることが望ましい。実行時間は、合意された一連の小作業負荷によって決まる。目標実行時間は約3時間である。

## 2.8 プラットフォーム

SERT 1.0.0.0 は、資源を未決定にしたまま、以下のプラットフォーム/OS/JVM の組み合わせ (64bit のみ) について実施され、試験される予定である。1 つ以上の JVM が一般的に入手可能であり、1 つの JVM を選択することにより特定プロセッサアーキテクチャまたはオペレーティングシステムが不当に不利になる可能性がある場合、SPEC は、2 つ以上の JVM の使用を推奨する。

HW プラット フォーム	x86 AMD	x86 AMD	x86 AMD	x86 Intel	x86 Intel	x86 Intel	Itanium Intel	Power IBM	Power IBM	Power IBM	SPARC Oracle	SPARC Fujitsu
OS	Windows Server 2008 R2	LINUX	Solaris	Windows Server 2008 R2	LINUX	Solaris	HP-UX 11i	AIX	IBM i	LINUX	Solaris	Solaris
JVM	IBM j9 Oracle HS	IBM j9 Oracle HS	Oracle	IBM j9 Oracle HS	IBM j9 Oracle HS	Oracle	HP HS	IBM-j9	IBM-j9	IBM-j9	Oracle	Oracle

注記：OS は、SERT の公表時点における最新のバージョン（サービスパックおよびパッチ等）を指す。

現在一覧に示されていないプラットフォーム/OS/JVM の組み合わせは、これらに割り当てられる資源がない。追加アーキテクチャまたは OS への対応が望まれる場合には、要求する団体からの積極的な参加が必須である。JVM の追加は、SERT に対する JVM の無制限の使用に関する JVM 供給事業者の合意に基づいている。SPECpower 委員会に対して SERT 策定用に追加情報を提供する企業は、スケジュール的に余裕を持つことができると思われる。

### 2.8.1 出荷時構成による試験

顧客環境を代表する結果を提供するために、「出荷時」状態においてシステムを試験することを目標とする。極度の調整は認められないが、構成に対する妥当と判断される設定値の変更や一般的な最適化は、限定的に認められている。その他の変更は、試験実行が非遵守と示される要因になる。SERT は、追加調整を制限するために、本ツールにおいて JVM を起動する予定である。

許容可能な設定値の一覧は、本書の今後のバージョンと本ツールの操作資料に記載される予定である。本一覧は、SERT 公表前に EPA から合意をとり、SERT 実行規則の一部として明確に文書化されることが考えられている。

## 2.9 実行言語

クロスプラットフォーム対応の負担を軽減するために、コード本体は Java で書かれている。しかし、本フレームワークは、他の言語による実行にも対応できるように設計されている。

## 2.10 負荷水準

複数負荷水準は SERT に求められる目標であり、その設計には、複数水準対応能力が含まれる予定である。動的アイドル負荷水準や 100% 作業負荷水準（最大消費電力ではない）は、既に有力な候補である。模擬試験により、どの水準を含むのか、また重み付けが必要であるかが示される予定である。

## 2.11 小作業負荷(worklet)

実際の用途に基づき従来の SPEC による方法で作業負荷を策定することは、特に IO 集約型作業負荷に関して、複雑な試験環境や多くの実行時間の原因になると考えられている（例：多くのクライアントシステムは、ネットワーク IO を必要とし、ストレージ IO 用に大型ディスクサブシステムを必要とすると考えられている）。このような試験の実施による費用は、評価ツールにとって非常に高額になる可能性があった。そのため SERT の作業負荷は、様々な異なる負荷シナリオ用の人工的な小作業負荷の一群になる予定である。

## 2.12 作業負荷(workload)

既存の SPEC ベンチマークは、1 つの応用分野における一般的な作業負荷、または既存のサーバー実行の分析から得られた人工的な作業負荷を代表する、実際の用途を調整したバージョンに基づいている。これらベンチマークは、消費電力測定値が含まれている場合において、サーバーの総合的な性能または効率における様々な下位項目を評価するのに適している。これらは、サーバーの総合的な性能または効率の代表的評価を示すようには設計されていない。

しかし SERT の設計目標は、サーバーアーキテクチャのすべての主要項目を含めることであり、これにより、サーバーがある作業負荷では良い結果を示し、他の作業負荷では不利な結果を示す可能性がある、特定アーキテクチャ特性の選好を回避する。

その代わりに、SERT 作業負荷は、様々な負荷パターンを使用して異なるサーバー機能を活用し、サーバーのすべての主要構成装置に様に負荷を与えることが意図されている。

各負荷パターンによって一部の構成装置に適切に負荷を与えることができない場合は、これら構成装置に対する境界値を調整する（例：負荷パターンに使用されない追加構成装置に対する消費電力許容値を増大させる）ことにより、これを補うことができる。

特に EPA の ENERGY STAR サーバー基準バージョン 2.0 のスケジュールにおける時間的制約を考慮すると、上記の目標を達成する単一作業負荷を設計できる可能性は極めて低い。そのため、SERT 作業負荷は、いくつかの異なる小作業負荷で構成され、それぞれがサーバーの特定機能に負荷を与えることになる。なおこの方法は、総合的システム成績に加えて、サーバー構成装置に対する個別の効率成績の作成を支援する。

図 1 は、SERT 試験一式の全体構造と構成要素を説明している。

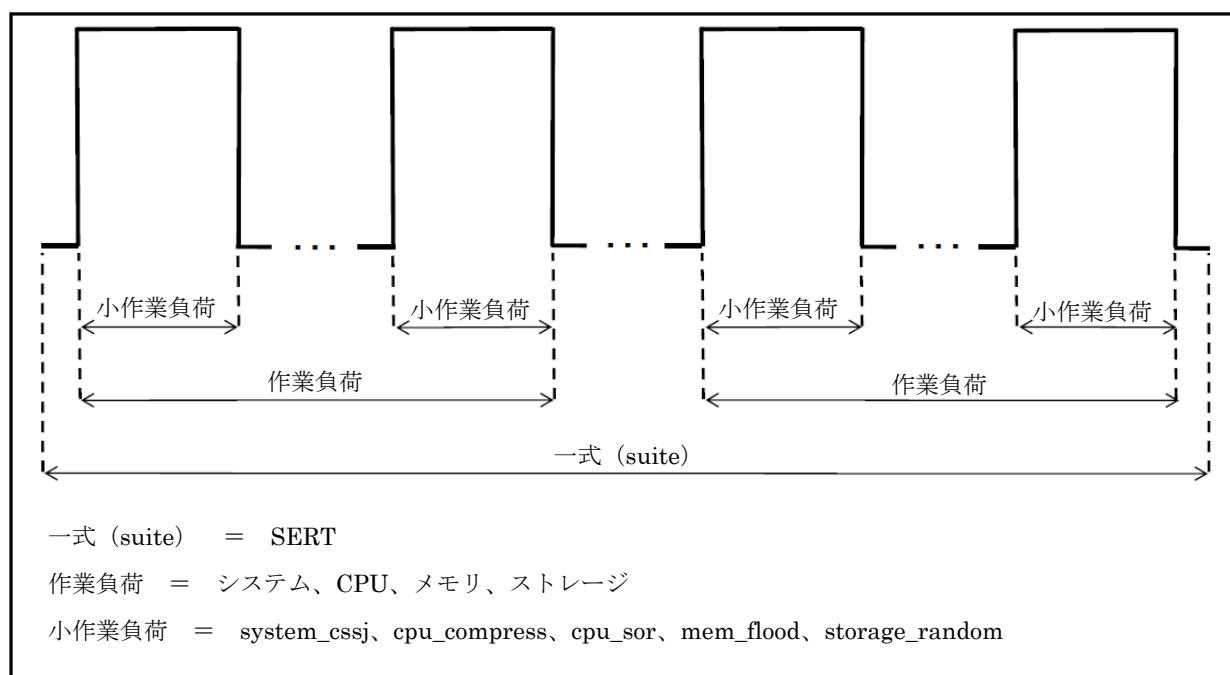


図 1: SERT 一式の構成要素

### 2.13 暫定試験スケジュール

初期（アルファ）試験段階は、2011年3月に開始される予定であり、各段階の開始には、前段階が問題無く完了していることが必要となる。予測スケジュールは、すべての設計詳細を決定した後に策定される。

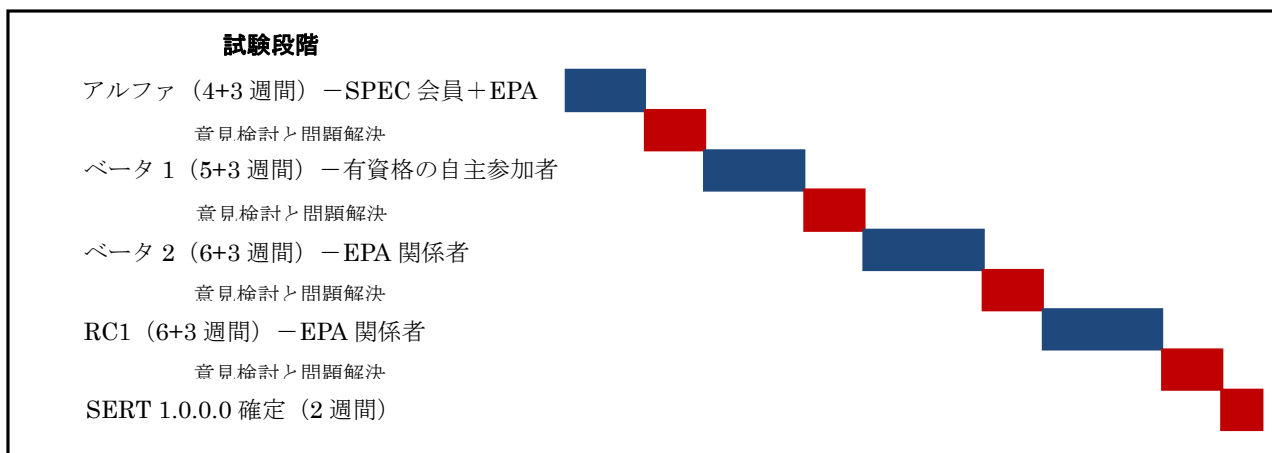


図 2: 暫定試験スケジュール

### 2.14 スケジュールの妥協点

SPEC ベンチマークは、可能な範囲において、複数のハードウェアとソフトウェアのアーキテクチャを直接比較可能な結果を生成するという目的を持って策定されている。この同一の基本目標は、本書に示されているように SERT の設計の方向性を示している。

SERT はアーキテクチャに依存しないという目標のもとに設計されているが、各作業負荷、およびすべての対応するアーキテクチャに対するツールハネネスについては、コードの実行が必要である。更に、公正かつ比較可能な結果を生成する、機能的に同等のバイナリー式を確保するために、このコードは、すべてのアーキテクチャに対して集中的に試験されなければならない。単なる移植可能なプログラミング言語の使用は、これら目標の達成において十分ではない。その結果、策定作業に大幅な複雑さが追加される。

SERT が一次近似評価ツールとして設計されていることを考慮した場合、比較可能性は、競争マーケティングに利用されるベンチマーク（二次近似ツール）の場合とは異なる方法で扱われる可能性がある。それでも、最低限の比較可能性を確保することは不可欠である。

SPECpower 委員会において利用可能な資源は限られており、単一アーキテクチャ用ツールの適時開発は非常に困難である。追加アーキテクチャへの対応によって、コードの移植に資源が必要となるために、基本試験手順の策定から移動させることになる。更に追加試験の取り組みは、比較可能性を確保するため、新規アーキテクチャに対してだけでなく、初代の実装に対しても必要とされる。そのため、各追加アーキテクチャが、現在の不確定な時間的スケジュールの原因となっている。SERT は、HW アーキテクチャごとに選択されたオペレーティングシステム（OS）上で最初に実施される予定である。

### 3 SERT 基本設計概念

#### 3.1 システム概要

SERT は、全体構造において SPECpower\_ssj2008 の設計理念と構成要素を共有している。SERT は、複数のソフトウェア構成要素で構成されている。

最も基本的な SERT ハードウェア測定設定については、以下の各項目が必要とされる。

- 被試験システム (SUT : system under test) - 測定が行われる実際のシステム。制御装置および SUT は、イーサネット接続を介して相互に接続されている。
- 制御装置 (例 : サーバー、PC、ラップトップ) - 消費電力測定器および温度センサーが接続されているシステム。
- 消費電力計測器 - 制御装置に接続されており、SUT の消費電力を測定するために使用される。
- 温度センサー - 制御装置に接続されており、SUB が設置されている周囲温度を測定するために使用される。

SERT は、以下を含めた幾つかの構成要素で構成されている。

- 試験ハーネス (Chauffeur) - SUT に設定されたソフトウェアおよび制御装置システム自体を制御すると共に、消費電力データの測定と記録の論理的側面を処理する。
- ディレクター - SUT に作業負荷の実行を指示する。
- 作業負荷 (一連の小作業負荷) - 試験ハーネスが消費電力と温度のデータを収集する間、SUT を動作させる。
- SPEC PTDaemon - 消費電力測定器と温度センサーに接続され、作業負荷を実行している間のこれら計測器の読み取り値を収集する。
- レポーター - 実行完了後の環境、消費電力、および性能のデータを収集し、読み取りが容易な形式に編集する。
- キットの設置や実行を容易にするために、キットの今後のバージョンには、GUI も含まれる予定である。

基本システム概略図において、これら構成要素と相互関係が示されている。

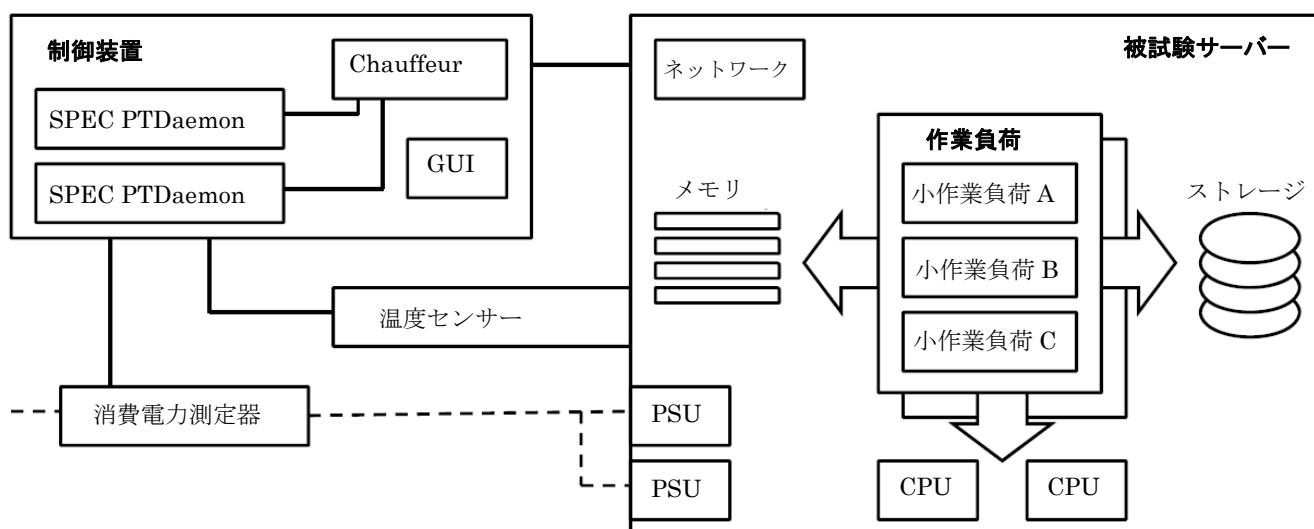


図 3: SERT 概要

## 3.2 SERT の実行

これら基本手順が、SERT キットの実行に必要とされる。

1. 消費電力測定器および関連する SPEC PTDaemon スクリプトを設定する。
  - SUT のアンペア数と電圧を正確に測定できるように消費電力測定器を構成する。
  - 適切な消費電力測定器のモデルが特定され、正しい通信およびネットワークポートが使用されることを確保できるように、`runpower.bat/sh` スクリプトファイルを加工する。
  - SPEC PTDaemon が消費電力測定器と接続し通信していることを確保する。
2. 温度センサーおよび関連する SPEC PTDaemon スクリプトを設定する。
  - 適切な温度センサーのモデルが特定され、正しい通信およびネットワークポートが使用されることを確保できるように、`runtemp.bat/sh` スクリプトファイルを加工する。
  - SPEC PTDaemon が温度センサーと接続し通信していることを確保する。
3. ディレクタスクリプトファイルを加工する。
  - 適切なシステム構成に合わせて `director.bat/sh` スクリプトファイルを加工する。
  - 適切な Java パスが特定されていることを確保する。
  - LOCAL\_DIRECTOR 変数に適切な情報が含まれていることを確保する。
  - DIRECTOR\_HOST 変数に適切な情報が含まれていることを確保する。
4. SERT スクリプトファイルを加工する。
  - 適切なシステム構成に合わせて `sert.bat/sh` スクリプトファイルを加工する。
  - 適切な Java パスが特定されていることを確保する。
  - 適切な数の JVM が特定されていることを確保する。
  - LOCAL\_DIRECTOR 変数に適切な情報が含まれていることを確保する。
  - DIRECTOR\_HOST 変数に適切な情報が含まれていることを確保する。
5. SPEC PTDaemon、ディレクタ、および SERT スクリプトを実行する。
  - `runpower.bat/sh`、`runtemp.bat/sh`、`director.bat/sh` および `sert.bat/sh` スクリプトを実行する。

キットの実行完了後には、`results.xml` ファイルが `\result\chauffeur-xxxx` ディレクトリ (xxxx は、実行反復数) に作成される。

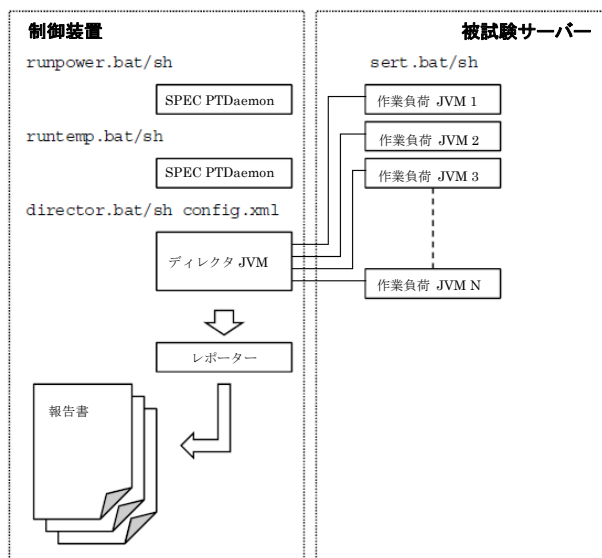


図 4: SERT 設定手順

## 4 小作業負荷の実行段階

SERT 試験一式は、CPU、メモリ、ストレージ等の被試験システム (SUT) の様々な構成装置に負荷を与えるように設計された、幾つかの作業負荷で構成されている。各作業負荷には、これらシステム構成装置の 1 つに負荷を与える特定のコードシーケンスを実行する、1 つ以上の小作業負荷が含まれている。SERT の全体設計構造は、上記の図 1 に示されている。

現行設計において、小作業負荷は、各段階において連続的に実行される予定である。これにより、総合指標に統合することが可能な、異なる負荷水準における個別の成績を得ることができる。

同時実行用に特別に設計されている、選択された小作業負荷または追加の小作業負荷は、追加段階において同時実行される可能性がある。小作業負荷の同時実行には、特に IO 負荷モジュールを使用した、より現実的なタスク切り替えを導入することになる。

小作業負荷の実行は、SERT 用語における「シーケンス (sequence)」と呼ばれる段階に分割される。現在 SERT ハーネスは、2 つの小作業負荷シーケンスすなわち「段階的測定 (Graduated Measurement)」と「固定反復 (Fixed Iteration)」に対応している。

低負荷水準に調整可能な小作業負荷には、図 5 に説明されているように段階的測定の実行計画を用いる。常に最大容量で実行する小作業負荷には、図 6 に示されている固定反復実行計画を使用しなければならない。

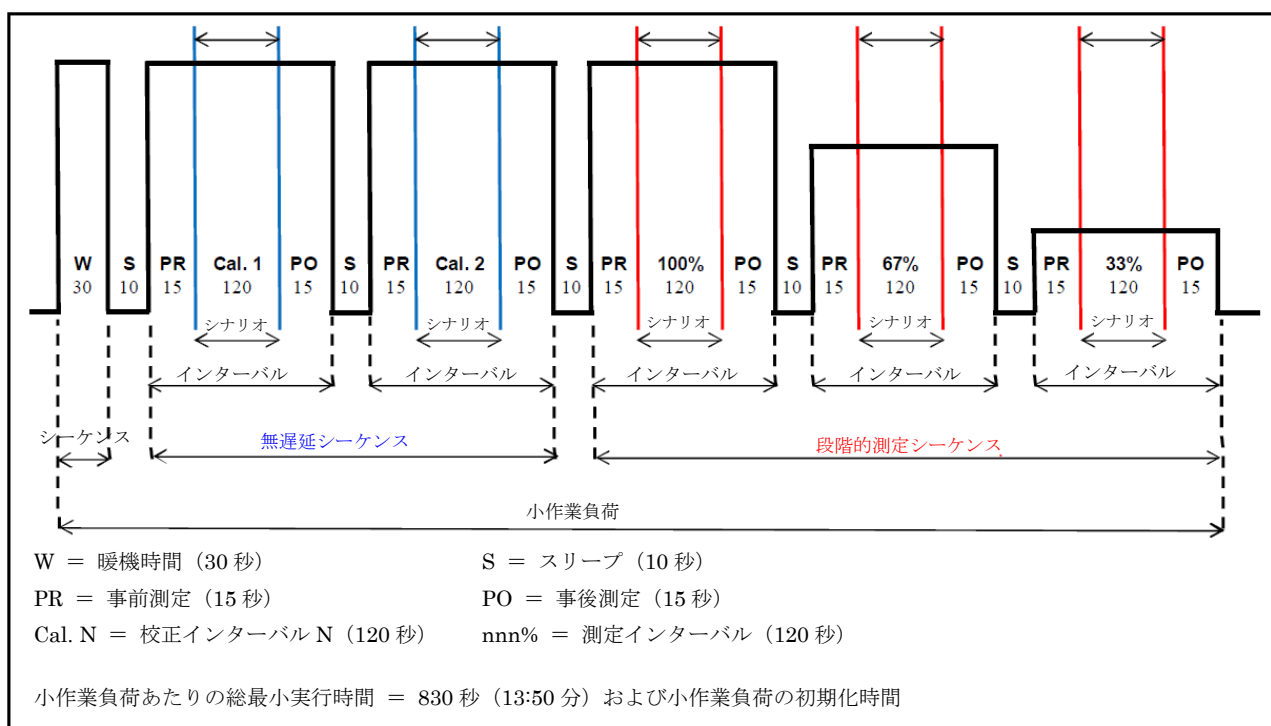


図 5: 段階的測定の実行

以下の一連の段階は、一般的に段階的測定の実行のために実施され、短いスリープ段階により中断される。

1. 暖機シーケンス
2. 校正のための無遅延シーケンス
3. 性能および消費電力のデータを判断するための段階的測定シーケンス

暖機および校正の段階において小作業負荷コードは無制限に実行されるが、その一方で SERT ハーネスは、測定シーケンスにおいて負荷水準を制御する。

各シーケンスは、図 5 に示されるとおり、固定時間インターバルに分割可能である。インターバルの数とその時間、および求められる負荷水準は、SERT 構成ファイルにおいて各小作業負荷に対し個別に判断することができる。

段階的測定実行を使用する小作業負荷の例は、System\_CSSJ、CPU\_Compress、Storage\_Random である。



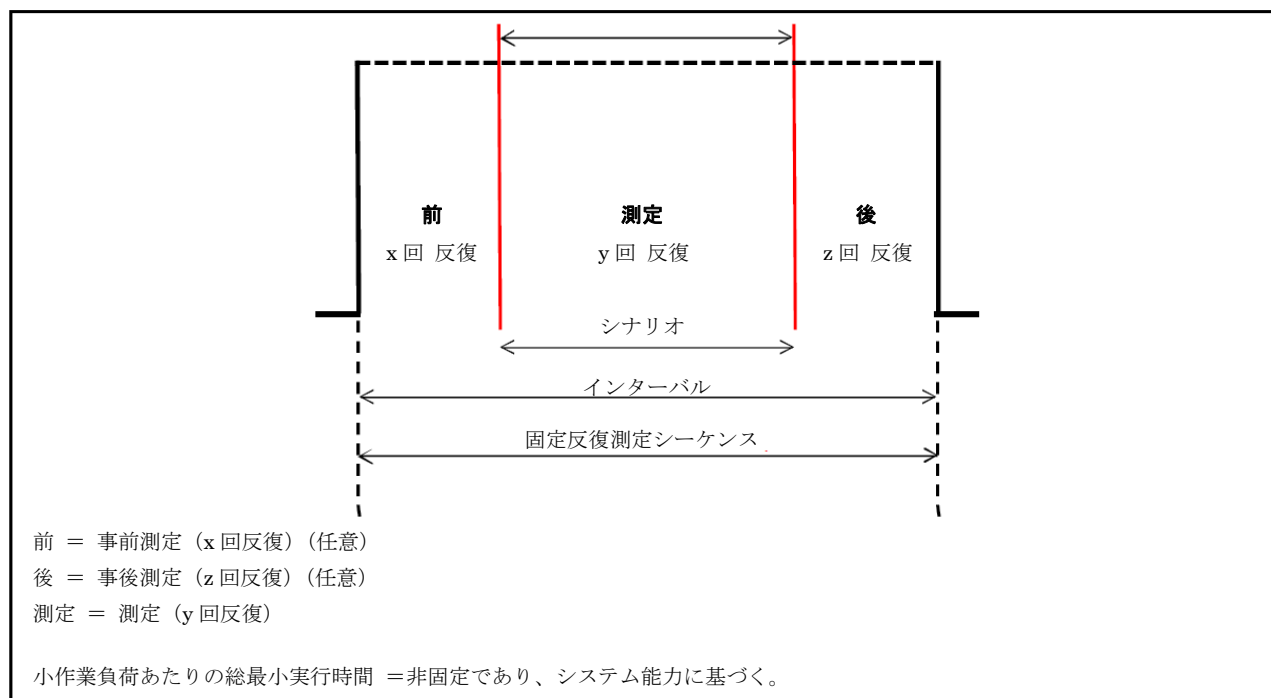


図 6: 固定反復の実行

固定反復実行計画には、一般的に 1 つのシーケンスと 1 つのインターバルのみが含まれている。インターバル時間は既定されておらず、システムの能力、すなわちそのシステムが固定量の作業を実行するために要する時間により判断される。

インターバルおよびシナリオの数は、SERT 構成ファイルにおいて、各作業負荷に対し個別に判断することができる。

現在、Mem\_Flood 小作業負荷のみがこの実行計画を使用する。

ユーザー	ユーザーとは、作業を開始することができる外部主体の象徴 (例: 人間)。
	各ユーザーは、識別情報を維持することができる。
	例: 各ユーザーは、Warehouse を意味する。
	各ユーザーは、状態情報を維持することができる。
	1 つのトランザクションから別のトランザクションまで継続する一時的情報 単一作業負荷に対して、異なる種類のユーザーが存在する可能性がある。
トランザクション	トランザクションは、ユーザーおよびトランザクション特定の入力を、変数 (パラメータ) として受け取る。 結果を一部生成する。
	一部のトランザクションは、結果を検証できる可能性がある。これは検査を目的に、トランザクションの小部分に使用することができる。
	シナリオ
シナリオ	小作業負荷は、特定種類のユーザーにより実行されるトランザクション一式である。 作業負荷は、複数の小作業負荷を含む可能性がある。
	各小作業負荷は、ユーザーとの一連の交流を表すことができる。
	トランザクション間に思考時間が生じる可能性がある。
インターバル	シーケンスにおける各インターバルには、事前測定が含まれている。
	各インターバル内において、各ユーザーは小作業負荷の実行を予定する。
	シナリオの予定時間に達したとき、シナリオは、トランザクションを通じて繰り返し適用される。各トランザクションは、JVM-全体のスレッドプールに送信される。小作業負荷における次のトランザクションは、進行中のトランザクションが完了した後に送信される。

<b>シーケンス</b>	各段階は、一連のインターバルで構成されている。	
	シーケンスにおけるインターバルは、共通部分を有する（ただし、「共通部分」は、作業負荷または構成に基づき変化する可能性がある）。	
	無遅延シーケンス	無制限のシナリオを実行する固定時間インターバル
	段階的測定シーケンス	管理されたシナリオの実行を伴う固定時間インターバル
	固定反復測定シーケンス	シナリオごとに既定の反復回数を実行する。成績は実行時間から算出される。事前および事後のインターバル段階は任意であり、一部の小作業負荷については無い可能性がある。
<b>小作業負荷</b>	作業負荷は、ユーザーと小作業負荷の組み合わせを定義する。	
	作業負荷の実行には、以下の段階が含まれる。 暖機 校正 1つ以上の測定段階	
	これらの各段階は、実際には一連の測定インターバルである。	
	複数の測定段階は、様々な混合トランザクションやユーザー等のために使用される可能性がある。	

## 5 小作業負荷の設計指標

すべての小作業負荷と広範な技術から一定の結果を得るために、以下の指針を遵守すること。

- 各小作業負荷は、多様な性能水準（例：100%（最大負荷）から0%（アイドル）の間の所定水準）に調整可能でなければならない。
- 各小作業負荷は、それ自体の最大性能水準に調整しなければならない。すなわち、試験実施者による100%水準の定義はない。
- 複数のプログラミング言語が使用される可能性がある。
- 可能であれば試験プログラムのコンパイル済みバイナリを使用すること。
- 各小作業負荷は、利用可能なハードウェア資源に応じて増減すること。より多くの資源は、より高い性能成績（スコア）をもたらす。例えば、より多くのプロセッサ／メモリ／ディスク容量、または追加のプロセッサ／メモリ／ディスクモジュールは、効率評価における性能に関する部分において、優れた結果をもたらす。
- 使用許可、再使用、および適応に関するすべての SPEC 規則に従う移植コード。
- 様々なアーキテクチャおよび／または OS に対応するために、アーキテクチャおよび OS に中立であるか、あるいは「if-def」機能を有するかのいずれかであること。
- 各小作業負荷により達成される作業は、「重要（important）」として明確に識別可能であるが、「極めて重要（all-important）」な種類の作業を対象にする必要はない。

これら指針に従うために、恐らく作業負荷は、各バッチが1つのトランザクションを構成する、個別作業のバッチに基づく予定である。異なる負荷水準は、必要なトランザクション数を計画することにより達成される。

### 5.1 有効アイドル小作業負荷

有効アイドル測定の間、SUT は、作業負荷トランザクションを完了できる状態でなければならない。有効アイドル小作業負荷は、有効アイドルインターバルにおいてトランザクションが発生しない場合を除き、その他すべての小作業負荷と整合する方法で取り扱われる。

### 5.2 CPU 小作業負荷

特定され適切とされる文字列操作、タスク管理、Java「commercial」処理、C「commercial」処理、数値処理、および他のタスクを含む、多種多様なプロセッサ集中タスクの組み合わせ。

- プロセッサ、コア、有効スレッド等の数に関係無く整合する模擬「ユーザー」あたりのプロセッサ特性
- 100%における障害はプロセッサであり、ストレージまたはメモリではない。
- 負荷を100%からアイドルまで段階的に水準を下げるができるような方法で、プロセッサタスクまたはタスク群を予定することができる。
- CPU 小作業負荷は、以下の項目について高い（優れた）性能成績を測定すること。
  - 高いCPU数、高いコア数、高い論理プロセッサ数、高い周波数、大きな総キャッシュ、低減された待ち時間、CPUソケット間における高速相互接続。

### 5.3 メモリ小作業負荷

無作為的と順次的、読み込みと書き込み、小容量と大容量のメモリ利用の組み合わせ。

- メモリ DIMM のサイズや数に関係無く整合する模擬「ユーザー」あたりのメモリ利用特性
- 100%における障害はメモリ自体であり、プロセッサまたはストレージではない。
- 負荷を100%からアイドルまで段階的に水準を下げるができるような方法で、メモリ負荷タスクまたはタスク群を予定することができる。
- メモリ小作業負荷は、メモリ特性（例：大きな帯域幅、低減された待ち時間、総メモリサイズ）に基づき、高い（優れた）性能成績を測定すること。

## 5.4 ネットワーク IO 小作業負荷

構成の消費電力／性能変更因子は、ネットワーク IO を利用するために確立される。

- 高額かつ大規模な外部試験システム構成を回避する。
- 測定値は、現代の技術に関して、100%および 0%のネットワーク利用に、大幅な消費電力の違いは無いことを示している。

## 5.5 ストレージ IO 小作業負荷

無作為的と順次的、読み込みと書き込み、小型と大型 IO の組み合わせ。

- システムサイズやディスクまたは搭載メモリの数に関係無く整合する模擬「ユーザー」あたりの I/O 特性
- 100%における障害はストレージサブシステムであり、プロセッサまたはメモリではない。
- 負荷を 100%からアイドルまで段階的に水準を下げるができるような方法で、I/O タスクまたはタスク群を予定することができる。
- ストレージ小作業負荷は、大きな帯域幅および低減された待ち時間について、高い（優れた）性能成績を測定すること。

任意の拡張ストレージ制御装置カードまたはサーバーのブレード筐体ストレージの消費電力と性能の測定は、SERT の対象ではない。

## 5.6 システム小作業負荷

多種多様なプロセッサおよびメモリ集中タスクの組み合わせ。

- 100%における障害は、プロセッサとメモリである。
- 負荷を 100%からアイドルまで段階的に水準を下げるができるような方法で、プロセッサタスクまたはタスク群を予定することができる。
- システム小作業負荷は、以下の項目について高い（優れた）性能成績を測定すること。
  - 高い CPU 数、高いコア数、高い論理プロセッサ数、高い周波数、大きな総キャッシュ、低減された待ち時間、CPU ソケット間における高速相互接続。
  - 大きな帯域幅、低減された待ち時間、総メモリサイズ

## 6 消費電力と温度の測定

SERT は、一般的な消費電力測定器と温度センサーから測定データを自動的に収集し、そのデータを SERT 結果に自動的にまとめる能力を提供する。この測定器とセンサーは測定フレームワークにより支持されており、本章における規定を遵守していることが必要とされる。

### 6.1 環境条件

消費電力は、利用環境の大部分を代表する環境で測定される必要がある。その目的は、SERT 実行前および実行中の、サーバーの消費電力または性能に人為的に影響を与える可能性がある極端な環境を防ぐことである。

以下の環境条件を満たす必要がある。

- 周囲温度下限値：20°C
- 周囲温度上限値：SUT の文書化された動作規定範囲以内
- 標高：SUT の文書化された動作規定範囲以内
- 湿度：SUT の文書化された動作規定範囲以内
- 通常データセンター業務と整合しない方法による、あからさまな被測定機器付近の気流の管理は認められない。

### 6.2 温度センサー仕様

温度は、SUT の主要吸気口の正面から最大 50mm（風上）の位置で測定されなければならない。温度測定値の比較可能性と再現性を確保するために、SPEC は、ベンチマーク中に使用される温度測定装置に対して、以下の特性を求める。

- ログイング — センサーは、ベンチマークハーネスがその測定値を読み取れるようにするインターフェースを有していなければならない。そのセンサーが対応する読み取り速度は、少なくとも 4 分あたり 4 サンプルでなければならない。
- 精度 — 測定値は、ベンチマーク実行中に測定される範囲に対し、摂氏 ±0.5 度またはそれ以上の総合精度を有するセンサーにより、報告されなければならない。

### 6.3 消費電力測定器の要件

消費電力測定値の比較可能性および再現性を確保するために、消費電力測定装置の以下の特性が、SERT に必要とされる。消費電力測定器の精度と波高率の動的特性の理由から、消費電力測定器は、ある消費電力範囲において使用されるときにはこれら要件を満たすが、他の範囲においては満たさない可能性があることに留意すること。消費電力測定器の自動範囲決定機能の使用は認められない。

- 測定値 — 測定器は、有効実効 (RMS) 消費電力 (ワット) と、電圧、アンペア数、および力率のうち少なくとも 2 つの測定単位について、報告しなければならない。
- 精度 — 測定値は、ベンチマーク実行中に測定される範囲に対し、1% またはそれ以上の総合不確実性を有する測定器により、報告されなければならない。総合不確実性は、ベンチマーク実行中に得られた測定値に対する、すべての特定測定器の不確実性の合計を意味する。
- 校正 — 測定器は、NIST (米国) (<http://nist.gov>) あるいは他国における同等の国立計測機関に由来する規格により、校正が可能でなければならない。測定器は、過去 1 年の間に校正されていなければならない。
- 波高率 — 測定器は、最小値が 3 の電流波高率を提供しなければならない。波高率を規定しない測定器の場合、測定器は、ベンチマーク実行中におけるあらゆる 1 秒サンプルにおいて測定された最大アンペア数の、少なくとも 3 倍のアンペアスパイクを測定可能でなければならない。
- ログイング — 測定器は、SPEC Daemon によって測定値を読み取れるようにするインターフェースを有していなければならない。測定器が対応する読み取り速度は、少なくとも 1 秒あたり測定値一式 (この一式はワットと、電圧、アンペア数、力率のうち少なくとも 2 つと定義される) でなければならない。測定器のデータ平均化インターバルは、1 回 (推奨) または 2 回のいずれかの読み

取り間隔でなければならない。「データ平均化インターバル」は、測定一式を提供するために、測定器の高速サンプリング電子装置により捕捉されたすべてのサンプルが平均化される時間として定義される。

例：

最大消費電力値が 200W の試験において使用される、±4 桁の読み取り値に対する±0.5%のベンダー固有精度を有する測定器は、 $((0.5\% \times 200W) + 0.4W) = 1.4W / 200W$ すなわち 200W において 0.7%の「総合」精度を有する。

最大消費電力値が 200W の試験において使用される、±4 桁の範囲に対する±0.25%のベンダー固有精度を有する、ワット範囲が 20~400W の測定器は、 $((0.25\% \times 400W) + 0.4W) = 1.4W / 200W$ すなわち 200W において 0.7%の「総合」精度を有する。

## 6.4 SPEC PTDaemon

SUT 以外のシステムに、測定インターバルにおける消費電力計測器または温度センサーを制御する業務を移すために、SERT では、SPEC PTDaemon (消費電力/温度デーモン、PTD、または ptd としても知られる) が使用される。SPEC PTDaemon は、様々な消費電力計測器インターフェースプロトコルの詳細や、SERT ソフトウェアからの行動を隠し、多様なベンチマークハーネスに容易に統合することができる、一般的な TCP-IP を基本とするインターフェースを表示する。

SERT ハーネスは、TCP ポートを開き、本書の API 章に説明されている単純なコマンドを使用することにより、PTDaemon に接続する。大型構成については、複数の IP/ポートの組み合わせが、複数の装置を制御するために使用される。

PTDaemon は、各装置の種類に特化したプロトコルおよびインターフェースを使用して、多様な種類の測定器およびセンサーに接続することができる。装置の種類は、daemon の初期起動に関するコマンドラインにローカルに送られる変数 (パラメータ) により特定される。

SUT と PTDaemon 間の通信プロトコルは、装置の種類に関係なく変化しない。これにより、対応する装置の種類に関係無く、SERT を開発することができる。

## 6.5 対応および準拠装置

SERTは、SPECの承認測定装置の一覧とSPEC PTDaemon更新方法を利用する。現在 (SPEC PTDaemon が) 対応しており、(仕様において) 準拠している消費電力計測器と温度センサーの一覧については、装置リスト ([http://www.spec.org/power\\_ssj2008/docs/device-list.html](http://www.spec.org/power_ssj2008/docs/device-list.html)) を参照する。

## 6.6 消費電力測定器の設定

消費電力測定器は、交流線間電圧源と SUT の間に位置していなければならない。その他の有効構成装置は、交流線間電圧源と SUT の間において認められない。

SPEC PTDaemon により規定される消費電力計測器の構成設定はを、手動で無効してはならない。

## 6.7 直流線間電圧

SPEC PTDaemon は、現時点において直流負荷への対応あるいはそれをを用いた試験のいずれも行っておらず、現在この対応の追加に充当される資源が無い。本件について策定および対応する機会を得るために、SPECpower 委員会が直流演算に重点的に取り組む企業からの新たな資源を利用できる場合、我々は直流対応の追加を支持する。

また、交流・直流変換損失が直流給電サーバーには含まれていないことから、交流給電されるサーバーと直流給電されるサーバーを比較することは公正ではない。そのため、我々は、直流給電サーバーの単独区分を策定することを推奨する。

## 7 指標(metric)／成績(score)、報告、ロギング

### 7.1 指標／成績

SERT はベンチマークになることを目的としていないが、それでもなお評価ツールとして、SERT は、被試験サーバーの効率を示す指標または成績を提供しなければならない。この指標は、SERT の対象となるすべてのシステム間で比較可能な方法で、SUT の性能と消費電力の両方を組み合わせなければならない。この比較の望ましい結果は、システムの相対的消費電力-性能効率の定量測定である。高い指標を生成するシステムは、低い指標を生成するシステムよりも、大きな消費電力-性能効率を有する。

異なるアーキテクチャは異なる作業負荷を異なる方法で実行することから、SERT は、アーキテクチャ中立性を確保するため、いくつかの独立した小作業負荷で構成されている。各小作業負荷は、SUT が達成した性能を示す測定値を生成し、その後これら測定値は、本ツールに使用されるすべての小作業負荷における SUT の総合性能を示す指標を得るために、他の小作業負荷により生成される測定値と統合されなければならない。SPEC は、この方法で得られた複数の性能測定値を、個別測定値の相乗平均として単一指標に統合することを提言する。

個別小作業負荷が順次または同時に実行されるかに依らず、個別の小作業負荷性能の相乗平均が使用される可能性がある。測定値を相乗平均である単一指標に統合する前に、我々は、選択された小作業負荷とその各測定値の大きさに基づき、基準成績に対して測定値を指数化する予定である。これらの技術には、単位の無い単一指標を提供し、類似範囲における個別測定値の評価基準を維持する利点があるために、大きな個別測定値を伴う小作業負荷が、小さい測定値を伴う作業負荷による結果を埋没させることはない。

総合性能は、確定後、システムの消費電力-性能効率を示す方法により、SUT の消費電力測定値と統合される。これは、異なる利用水準における SUT の消費電力-性能効率を考慮するために、SERT が自動的に実行する複雑な計算である。

SERT により生成された指標は、ENERGY STAR 評価と区別される。EPA は、SERT 成績がその一部分を構成する可能性のある ENERGY STAR 承認用の基準を決定する。試験された機器の上位 25%が ENERGY STAR 適合を達成することが見込まれている。「ゴールドレベル」の ENERGY STAR 適合は、結果の上位 5%を達成する機器に適用される可能性がある。また EU は、消費電力-性能効率に関して、EU が総合基準値を決める A~F レベルの段階的達成制度を提案している。

被試験サーバーは、EPA のものとは別の区分に分類される可能性がある。EPA は、これら区分の適用方法と、ある特定区分の機器を別区分の機器と比較してよいかを決める。

### 7.2 報告と出力のファイル

SERT は、2つの報告書とログファイル一式を生成する。この報告書は、希望情報の表示および／または保存に関する EPA とそのパートナー両方の負担を低減するために、XML 形式で作成される予定である。我々は、報告書の信頼性を確保するための措置（例：暗号化）をとる予定である。

#### 7.2.1 報告 1: 「概要報告」

この報告には、EPA により提供される被試験プラットフォームの「合否」通知のための表記欄が含まれている。試験が技術的エラー伴って完了している場合、試験実行の欄には非準拠と表示される。このような場合、エラーメッセージおよび／または警告が自動的に報告書に記載される。この報告書における情報は公開され、販促用途に使用することができる。

本報告書に含まれる項目は以下のとおり。

- EPA パートナー名および EPA パートナーID
- 試験されたプラットフォームの EPA の ENERGY STAR 区分
- 試験日と試験場所（および試験者名）
- 試験されたプラットフォームの製造事業者名およびモデル番号
- 「合否」表記欄

- 該当する場合において、警告またはエラーの通知
- システム構成情報（冗長構成装置は適切に表記される）
  - フォームファクタ
  - プロセッサの数と種類
  - 利用可能なプロセッサソケット数
  - メモリサイズ、種類、メモリ DIMM 数、DIMM スロット数、最大メモリ容量
  - 利用可能な拡張スロット数
  - 電源装置の数と型式・モデル、出力定格、最小／最大
  - 入力電力
  - 対応 OS／試験に使用した OS
  - ストレージ制御装置の数と型式・モデル
  - マスストレージ装置の数と型式・モデル
  - ネットワークインターフェースカード（NIC）の数と型式・モデル
  - 管理制御装置またはサービスプロセッサの搭載有無 [YES/NO]
  - 他のハードウェア特性／付属品（アクセサリ）

### 7.2.2 報告 2: 「消費電力と性能のデータシート」

この報告書には EPA が必要とするすべての情報が含まれており、これら情報は、SPEC が必要と考えるものである。消費電力と性能のデータシートは公開されるが、販促目的の使用は、SERT の公正利用規範（Fair Usage Rule）により禁止される。本情報は、EPA の審査を最も妨げる形式で EPA に提供される予定である。

この報告書は、以下の追加詳細と「概要報告」におけるすべてのデータが含まれる予定である。

- 総合結果／成績
- すべての目標負荷水準における結果
- ハードウェアおよびソフトウェアの構成
- 消費電力測定の概要
- 環境情報

### 7.3 妥当性確認(Validation)／検証(Verification)

SERT ソフトウェア構成要素は、精度の向上とユーザーエラーの低減を目的に、情報精度を改善し、ユーザー入力を検証し、実行時間データの収集を監視し、さらに結果の妥当性を確認するために、可能な限りソフトウェアの検査を実施することによって、無効な報告書が EPA に提供されるのを防止する。

条件または結果が特定の基準を満たしていないときには警告が表示され、SERT 報告書にエラーメッセージが表示される。

これら特性は、EPA パートナーが準拠した結果を生成し、EPA に対する誤った報告書の提出を回避するのを容易にさせる。

準拠検査内容の例は以下のとおり。

- 入力プロパティ（パラメータ）と負荷水準の実行時間を検証する。
- 範囲外温度が報告されている。
- 消費電力と温度の読み取り誤差が選択された境界値を下回っている。

すべての SERT ソフトウェア構成要素は、その機能領域において妥当性検査（例：接続問題の警告、ログ測定エラーと範囲外の状況、ユーザーに対する不明および不完全情報の警告、入力データの妥当性の検査）を実行する。

他の新たな妥当性確認方法は、SERT ソフトウェアの設計と実施の進展に合わせて検討される予定である。



## 7.4 ロギング

ログファイル一式が、各試験実行について生成される。

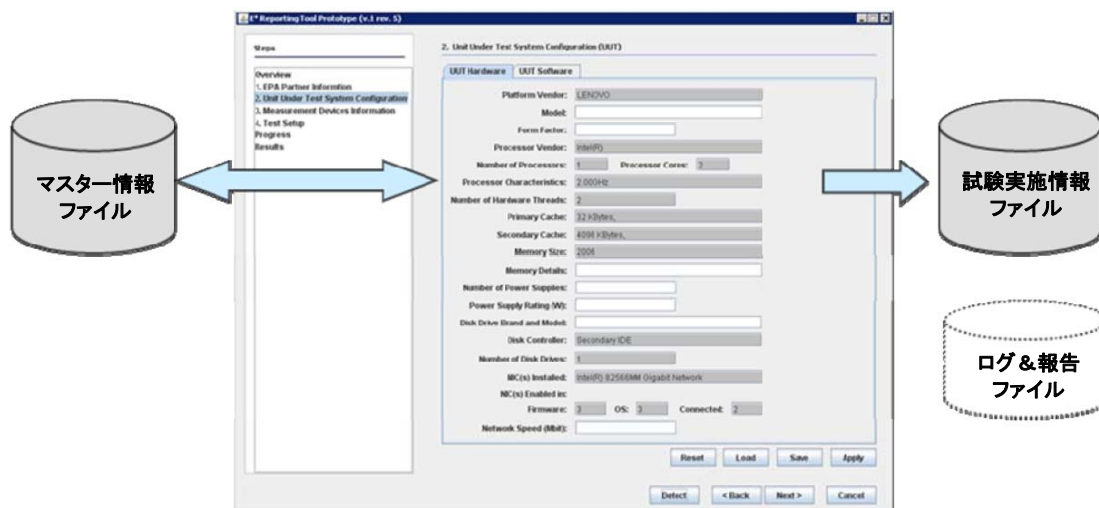
- ログファイルにおける情報は、「非公開」とされる予定である。
- 複数の連続した試験実行により複数のログファイルが作成されることから、これらファイルは、実行連番によって識別される。
- 各ログファイルとは、エラーや警告を含めて、試験の様々な段階におけるソフトウェア動作の記録である。
- ログファイルは、検査と支援を目的としている。
  - 問題または不具合は、この低水準詳細記録によって比較的簡単に解決することができる。パートナーの報告書の精度または正確性に関して問題が生じた場合、これらのログファイル（暗号化される可能性あり）は、多くの問題を解決するのに適していると考えられる。
- ログファイル内容の例は以下のとおり。
  - 様々な構成要素間における応答確認（handshake）妥当性確認メッセージ
  - エラーまたは警告メッセージ
  - 状態変更メッセージ／通知
  - 「トランザクション」即時／定期的概要情報
  - 「トランザクション」応答時間

EPA は、ENERGY STAR 適合の前に、上記の出力情報の一部またはすべてを提供するように求める可能性がある。いずれの場合においても、パートナーは、EPA に届出するあらゆる結果の全出力情報を保管することに、責任を持って取り組まなければならない。

## 8 将来的な強化／拡大目標

### 8.1 グラフィカルユーザーインターフェース(GUI)

拡大目標の 1 つは、試験実行の構成および設定を容易にし、試験実行の同時監視を可能にし、さらに結果を検討するために、グラフィカルユーザーインターフェース (GUI) を組み入れることである。SERT GUI は、ハードウェアおよびソフトウェア構成の検出または入力、予備試験または妥当性確認試験の手配、結果報告と試験環境に共通する他の機能の表示の手順を、使用者に説明する。



最低限の研修で SERT 試験を実行できるようにするとともに、結果の精度を向上させるために、SERT GUI には幾つかの機能が含まれている。

- タブ付き画面による簡単ナビゲーション
- 使用方法 (インライン利用ガイドとヘルプ)
- 構成発見 (検出機能) は、SUT と制御装置のハードウェアおよびソフトウェアに関する情報欄の大部分に、自動的に情報を表示する。
- GUI は、試験環境に関する所要情報を表示し、入力を可能にし、保存する。
  - 報告書用：企業情報、プラットフォーム構成、実施時間設定値等
  - マスターおよび試験実施情報ファイルは、保存され、再利用可能とされ、複数のプラットフォームについて時間を節約する。
- 試験の設定、実行および進行状況の表示
  - 測定の開始；実行の種類を選択 (予備または最終)
  - 進行状況、警告、およびエラーの表示
- 結果を表示し、報告書の印刷と記録を可能にする。
- 冗長構成装置および、消費電力と性能の変更因子 (modifier) に対する規定

### 8.2 試験ソフトウェア

SERT の「拡大目標」は、ある環境に関して、ツールの設定に「Live CD」方法を実行可能にすることであり、基本オペレーティングシステムを伴った本ツール一式は、SUT に他のオペレーティングシステムを設定することなく、単独で起動可能な CD または DVD から完全に実行できるようになる。これにより、設定の容易性が向上し、本ツールの普及率が増加すると考えられる。

この方法において考えられる課題には、新装置のための特定ハードウェアドライバの欠如、ベンダー独自の電力管理機能が無い可能性、一部のオペレーティングシステムに関する使用許可および利用可能度に関する問題等がある。代替方法には、セットアップ中に追加ドライバを設定できるようにする、あるいはベンダー独自の出荷時の OS 設定に使用するための試験インストーラーをバイナリと共に提供することが含まれる。

## 9 SERT および EPA の ENERGY STAR サーバー基準バージョン 2.0

SERT が目的の状況において使用されるようにするために、我々は、ENERGY STAR サーバー基準に以下の項目を含めることを推奨する。

### 9.1 測定

提供された SERT 試験キットは、SERT を実行し、SERT 測定結果を生成するために使用されなければならない。SERT 指標は、SERT 作業負荷の結果である（第 0 章参照）。SERT 結果は、他のアプリケーションから得られた消費電力と性能の指標とは比較できない。

### 9.2 SERT バイナリおよび再コンパイル

有効な実行には提供されたバイナリファイルを使用しなければならず、これらファイルは、いかなる方法においても更新または修正されてはならない。

### 9.3 手動介入

初期設定の完了後、SUT またはその内的小さいおよび外的環境に対する手動介入あるいは最適化は、試験測定中において認められない。

### 9.4 SERT 情報の公正な使用

ENERGY STAR プログラムの明確な目的は、可能な限り幅広くベンダーからの参加を得ることである。コンピュータ業界の性能ベンチマーク関係者は、性能の詳細情報が販促用に利用できるようになると、（公表時において）優れた製品を有するベンダーのみが結果の公表に意欲的になることを経験から認識している。業界に広く参加を促すため、本ツールで得られるあらゆる詳細情報の販促を目的とする使用を制限する、厳しい規定が実施されなければならない。事実に基づいている ENERGY STAR 適合を除き、いかなるデータも、EPA パートナーの販促用資料に使用されるべきではない。これら規定は、本ツールの使用許可と EPA パートナー合意の両方に明記される予定である。

これら規定は、厳密には本ツールの「設計」の一部ではないが、これら規定の存在は、設計の柔軟性や、求められる詳細な消費者向け情報の配信を認めるために必要であることに留意してほしい。

#### 9.4.1 公正使用規則

- 本ツールにより提供され販促用資料に使用することができる唯一の情報は、サーバー構成またはサーバー製品群（ファミリー）の ENERGY STAR 適合である。
- 本ツールにより提供され公開比較に使用することができる唯一の情報は、サーバー構成またはサーバー製品群（ファミリー）の ENERGY STAR 適合である。本ツールで得られるその他公開情報は、試験が正しく実行されていることを確認し、試験された構成が特定の要求をどの程度満たしているのかを消費者がより良く理解できるようにすることを支援するために利用される。
- ENERGY STAR プログラム以外の情報を得ることを目的とした調査に本ツールを使用する場合、得られた情報は、ENERGY STAR プログラムの結果と比較できない可能性があり、生成されたデータを用いた競争力の比較はできない可能性がある。
- EPA の ENERGY STAR 適合は、EPA の規定により管理されている。

### 9.5 認定独立試験所

認定された独立試験所を使用するという要件は、EPA の ENERGY STAR パートナー、特に小規模企業にとって、大きな負担をかける可能性がある。我々は、独立試験所の使用を選択肢として推奨するが、要件としての実施は推奨しない。

## 9.6 供給電圧公差

1%の変化範囲内において電圧を使用するためには、追加の電圧源が必要である。これは、パートナー、特に小規模企業にとって、不必要な費用の増加をもたらす。よって我々は、公差を±5%に設定することを推奨する。

## 10 小作業負荷候補

以下の表は、現時点における小作業負荷の候補と、様々な SERT 試験段階における採用予想を示している。早期公表段階に含まれている小作業負荷の候補は、以降の公表段階において変更される可能性がある。早期公表試験の結果は、その後の公表における小作業負荷の追加に影響を与える可能性がある。

作業負荷	小作業負荷の候補	アルファ	ベータ1	ベータ2	RC1
CPU	CPU_Compress	含む	未定	未定	未定
CPU	CPU_CryptoAES	含む	未定	未定	未定
CPU	CPU_SOR	含む	未定	未定	未定
CPU	CPU_FFT	含む	未定	未定	未定
CPU	CPU_LU	含む	未定	未定	未定
CPU	CPU_XMLvalidate	含む	未定	未定	未定
メモリ	Mem_Flood	含む	未定	未定	未定
メモリ	Mem_XMLvalidate1	含む	未定	未定	未定
メモリ	Mem_XMLvalidate2	含む	未定	未定	未定
ストレージ	Storage_Random	含む	未定	未定	未定
ストレージ	Storage_Sequential	—	未定	未定	未定
ストレージ	Storage_Mixed	—	未定	未定	未定
システム	System_CSSJ	含む	未定	未定	未定

## 10.1 CPU 小作業負荷: Compress

### 10.1.1 概要

Compress 作業負荷は、調整された Lempel-Ziv 方法 (LZW) を使用して、データを圧縮および解凍するトランザクションを実行する。基本的に、本作業負荷は、共通サブストリングを発見し、それらを可変サイズのコードと置き換える。これは決定性であり、かつオンザフライで行われる。このように、解凍手順は、入力テーブルを必要としないが、そのテーブルが構築された方法を監視する。このアルゴリズムは、IEEE Computer Vol17, No 6 (1984年6月) の8~19ページに掲載された、Terry A. Welch による「A Technique for High Performance Data Compression」に基づいている。

### 10.1.2 配列実行方法

段階的測定配列

### 10.1.3 指標

1秒あたりのトランザクション数

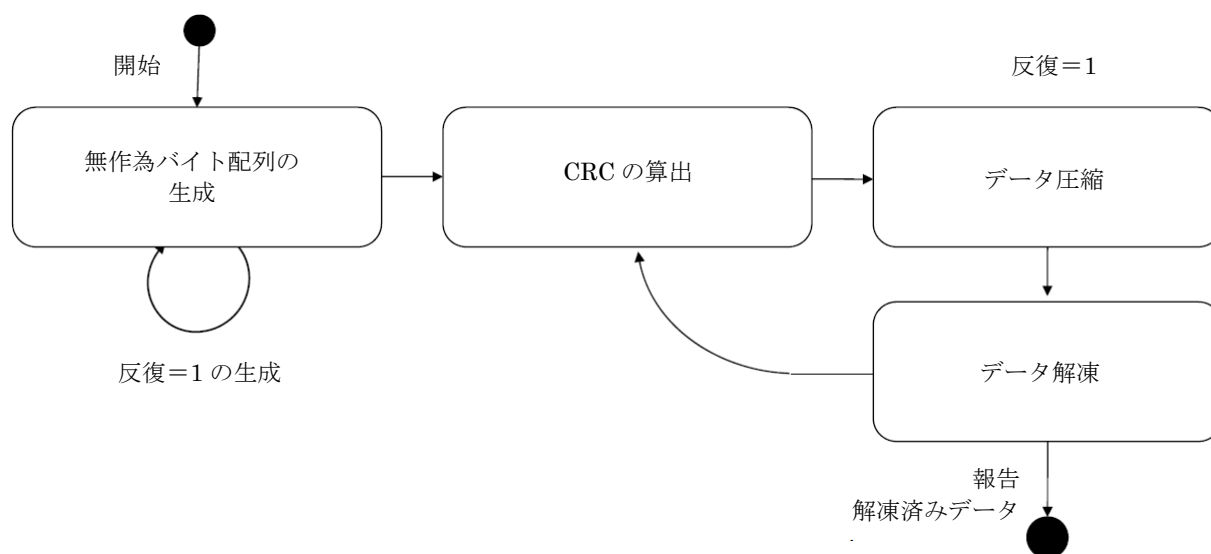
### 10.1.4 所要初期化

一定サイズのバイト配列は、各トランザクションを実行する前に、オンザフライで生成される。バイト配列の内容は無作為に作成される。

### 10.1.5 構成設定値

size	各トランザクション実行用の入力バイト配列の大きさ
enable-idx	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	無作為バイト配列課題の反復回数

### 10.1.6 トランザクションコード



## 10.2 CPU 小作業負荷: CryptoAES

### 10.2.1 概要

CryptoAES 作業負荷は、AES（または DES）ブロック暗号アルゴリズムを使用して、データを暗号化および復号化するトランザクションを実行する。そのアルゴリズムは設定可能なパラメータであるが、現在候補のバージョンは、CBC を有する AES を使用し、PKCS5 パディングを行わない。暗号化および復号化は、Java Cryptographic Extension (JCE) フレームワーク、特に Cipher クラスを用いて実施される。

### 10.2.2 シーケンス実行方法

段階的測定シーケンス

### 10.2.3 指標

1 秒あたりのトランザクション数

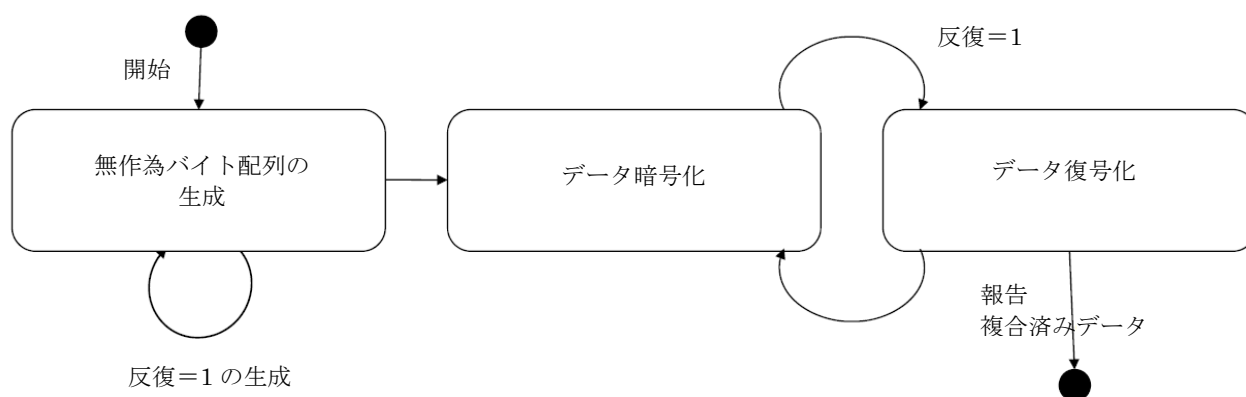
### 10.2.4 所要初期化

一定サイズのバイト配列は、各トランザクションを実行する前に、オンザフライで生成される。バイト配列の内容は無作為に作成される。

### 10.2.5 構成設定値

size	各トランザクション実行用の入力バイト配列の大きさ。
key-generator	キージェネレーターアルゴリズム。(AES または DESede)
key-size	キーの大きさ。(AES については 128。DES については 168)
algorithm	暗号化アルゴリズム。(例: AES/CBC/パディング無し、AES/CBC/PKCS5 パディング、DESede/CBC/パディング無し、DES/CBC/PKCS5 パディング)
level	暗号化の実行回数
enable-idc	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	無作為バイト配列課題の反復回数

### 10.2.6 トランザクションコード



## 10.3 CPU 小作業負荷: FFT

### 10.3.1 概要

Fast Fourier Transform (FFT) 作業負荷は、複素数の一次元フォワード変換を行うトランザクションを実行する。その浮動小数点計算は、複素数演算、シャッフリング、非定常メモリ参照、および三角関数を実行する。第1セクションでは、ビット逆転部分（浮動小数点演算なし）を実行し、第2セクションでは、実際の  $N \log(N)$  計算手順を実行する。（NIST 策定の Scimark ベンチマークからの出典。）

### 10.3.2 シーケンス実行方法

段階的測定シーケンス

### 10.3.3 指標

1秒あたりのトランザクション数

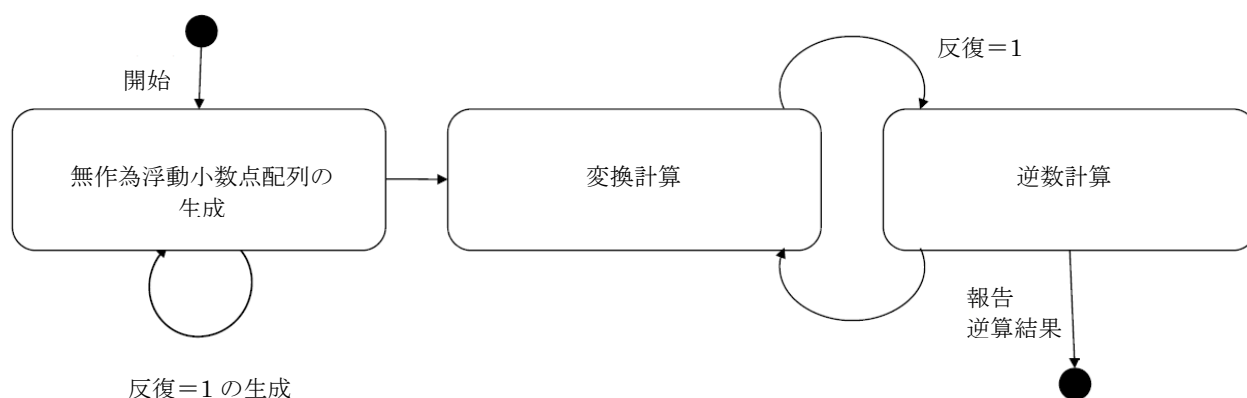
### 10.3.4 所要初期化

一定サイズの浮動小数点配列は、各トランザクションを実行する前に、オンザフライで生成される。この配列の内容は無作為に作成される。

### 10.3.5 構成設定値

array-length	各トランザクション実行用の入力浮動小数点配列の大きさ。
enable-idc	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	無作為配列課題の反復回数

### 10.3.6 トランザクションコード





## 10.4 CPU 作業負荷: LU

### 10.4.1 概要

LU 作業負荷は、部分軸回転を使用して、密行列の LU 因数分解を計算するトランザクションを実行する。この作業負荷では、一次代数カーネル (BLAS) および密行列演算を実行する。アルゴリズムは、1 階の更新を伴う right-looking 版の LU である。(NIST 策定の Scimark ベンチマークからの出典。)

### 10.4.2 シーケンス実行方法

段階的測定シーケンス

### 10.4.3 指標

1 秒あたりのトランザクション数

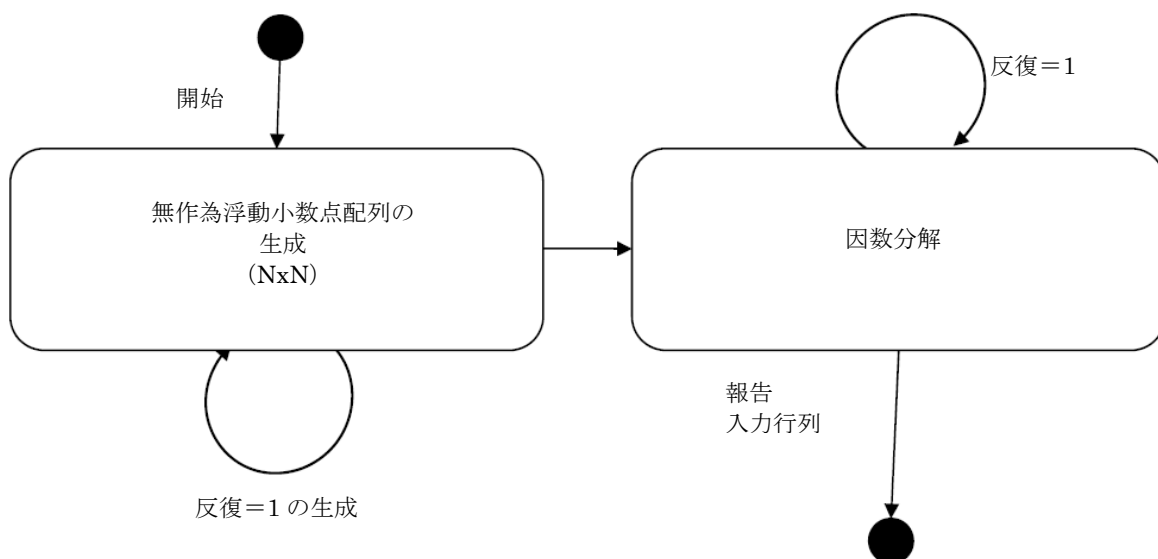
### 10.4.4 所要初期化

一定サイズの浮動小数点行列は、各トランザクションを実行する前に、オンザフライで生成される。この行列の内容は無作為に作成される。

### 10.4.5 構成設定値

matrix-dimen	各トランザクション実行用の入力浮動小数点行列の大きさ。(NxN)
enable-idc	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	無作為行列課題の反復回数

### 10.4.6 トランザクションコード



## 10.5 CPU 作業負荷: SOR

### 10.5.1 概要

Jacobi Successive Over-relaxation (SOR) 作業負荷は、例えばディリクレ境界条件を有する二次元のラプラス方程式を解くというような、有限の異なる用途において、標準的なアクセスパターンを実行する。このアルゴリズムは、各  $A(i,j)$  に4つの最近傍値の平均重みを割り当てる、基本「グリッド平均化」メモリパターンを実行する。一部の手動による最適化は、最新の数式における配列アクセスを合理化するために、G[]行をエイリアス処理することにより行われる。(NIST 策定の Scimark ベンチマークからの出典。)

### 10.5.2 シーケンス実行方法

段階的測定シーケンス

### 10.5.3 指標

1秒あたりのトランザクション数

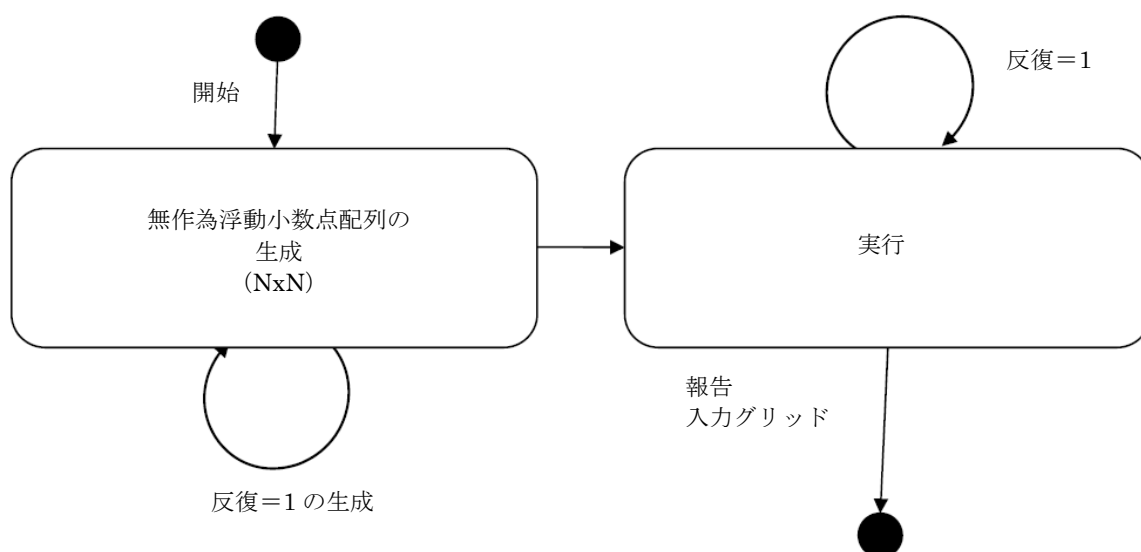
### 10.5.4 所要初期化

一定サイズの浮動小数点グリッドは、各トランザクションを実行する前に、オンザフライで生成される。このグリッドの内容は無作為に作成される。

### 10.5.5 構成設定値

grid-dimen	各トランザクション実行用の入力浮動小数点グリッドの大きさ。(NxN)
enable-idc	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	無作為グリッド課題の反復回数

### 10.5.6 トランザクションコード



## 10.6 CPU 作業負荷: XmlValidate

### 10.6.1 概要

XML 妥当性確認作業負荷は、Java の XML 妥当性確認パッケージ (javax.xml.validation) を実行する。SAX および DOM API の両方を使用し、XML ファイル (.xml) は、XML スキーマファイル (.xsd) に対して、妥当性が確認される。入力データを無作為化するため、XML 入力データ内において注釈付き領域を交換するアルゴリズムが適用される。

### 10.6.2 シーケンス実行方法

段階的測定シーケンス

### 10.6.3 指標

1 秒あたりのトランザクション数

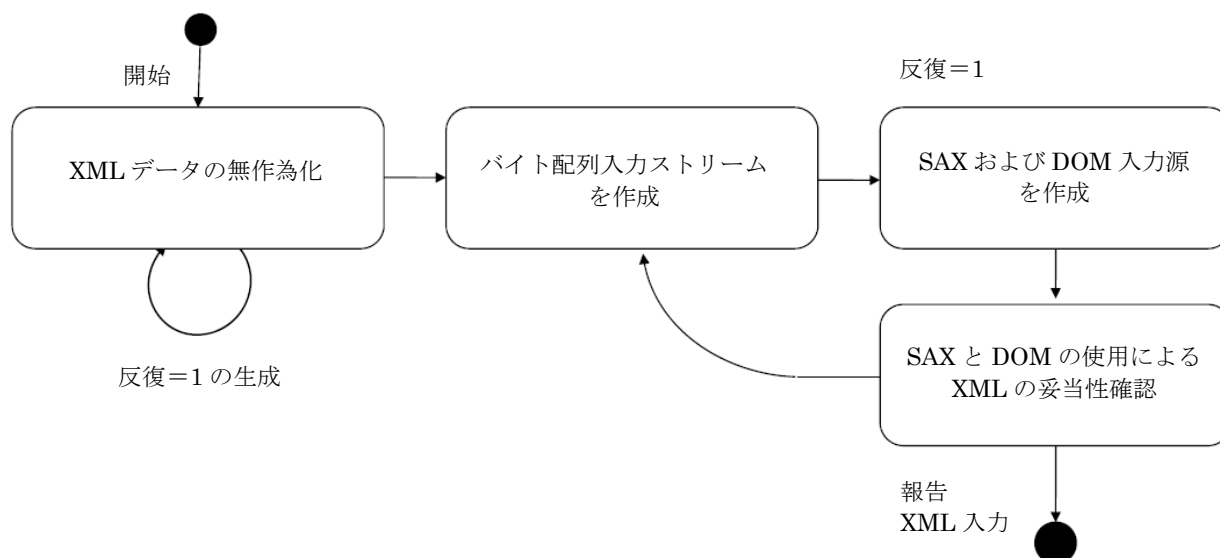
### 10.6.4 所要初期化

XML および XML スキーマの両ファイルがディスクから読み込まれ、その後の使用のためにバッファに保存される、初期化時間。(この初期化の完了後、さらなるディスク IO は無い。) 各トランザクションの実行について、ファイルの大きさあるいは複雑性を変更することなく構文解析に多様性を持たせるために、無作為化アルゴリズムは、事前にオンザフライでオリジナルの XML データに適用される。

### 10.6.5 構成設定値

xml-schema-dir	XML スキーマファイルのディレクトリを特定する。
xml-schema-file	XML スキーマファイルの名称を特定する。
xml-dir	XML ファイルのディレクトリを特定する。
xml-file	XML ファイルの名称を特定する。
enable-idc	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	XML ファイル無作為化の反復回数

### 10.6.6 トランザクションコード



## 10.7 メモリ小作業負荷: Flood

### 10.7.1 概要

Flood 作業負荷は、4 つの共通する重要な配列演算にわたるメモリ帯域幅を測定する一般的なベンチマークであり、STREAM に基づいている。Flood に使用される長い (64-bit) 整数配列については、課題 (assignment) につき以下のメモリ量が関係している。

1. **COPY** :  $a(i) = b(i)$   
-- 課題あたりの 8 バイト読み込み + 8 バイト書き込み = 16 バイト/課題
2. **SCALE** :  $a(i) = k * b(i)$   
-- 課題あたりの 8 バイト読み込み + 8 バイト書き込み = 16 バイト/課題
3. **ADD** :  $a(i) = b(i) + c(i)$   
-- 課題あたりの 16 バイト読み込み + 8 バイト書き込み = 24 バイト/課題
4. **TRIAD** :  $a(i) = b(i) + k * c(i)$   
-- 課題あたりの 16 バイト読み込み + 8 バイト書き込み = 24 バイト/課題

Flood 成績は、これら 4 つの試験の平均値に SUT に搭載されている物理的メモリ量を乗算することにより算出される、総システムメモリ帯域幅に基づいている。Flood は STREAM に基づいているが、Flood 成績は STREAM コードを使用せず、完全に Java で実行される。

Flood は、様々な重要な方法で STREAM を強化する。

1. Flood は、物理的メモリサイズに基づいて結果を増減することにより、大型メモリ構成のシステムに対応する。
2. Flood は、最新のマルチコアサーバーのメモリ帯域幅能力を、十分に引き出すように設計されている。Flood はマルチスレッド化されており、スレッドは帯域幅測定中に同時に動作するように予定され、最大スループットを確保し、結果のばらつきを最小化する。
3. Flood は、ユーザーによる設定をほとんどあるいは全く必要とすることなく、利用可能なメモリを十分活用するように、試験用データを自動的に拡大する。

多数のコアおよび複数のメモリ制御装置を有する大型サーバーの総システムメモリ帯域幅を測定することは、非常に困難である。特に実施に関するばらつきは、既存のメモリ帯域幅ベンチマークでは管理不可能な場合が多い。Flood は、3 つのメモリ帯域幅試験を連続して実施し、最初と最後の試験を放棄することにより、実施のばらつきを最小限にする。この方法により、Flood 成績の算出に使用される中間の測定において、すべてのスレッドを完全な同時状態で実行していることが確保される。

Flood 成績は、SUT の総メモリ帯域幅および SUT の物理的メモリ構成に応じて、直線的に増減する。CPU、ストレージ、およびネットワークの性能は、Flood 成績にほとんどあるいは全く影響しない。

Flood 作業負荷は常に固定の反復回数を利用し、また試験用のメモリ容量は搭載 DRAM を十分活用することから、実行時間はシステム設定に基づき変化する。24 スレッドおよび 48GB の物理的メモリを有する、2.2GHz、24 コア SUT において、Flood は完了するまでに約 20 分を要する。実施時間は、SUT に搭載されている物理的メモリ容量に応じて、比例的に変化する。また実行時間は、総スレッド数に影響される。

### 10.7.2 シーケンス実行方法

固定反復ディレクタシーケンス — Flood は、config.xml 内に規定されている任意の反復回数実行される。

### 10.7.3 指標

成績 = 総システムメモリ帯域幅 (GB/s) \* 物理的メモリサイズ (GB)

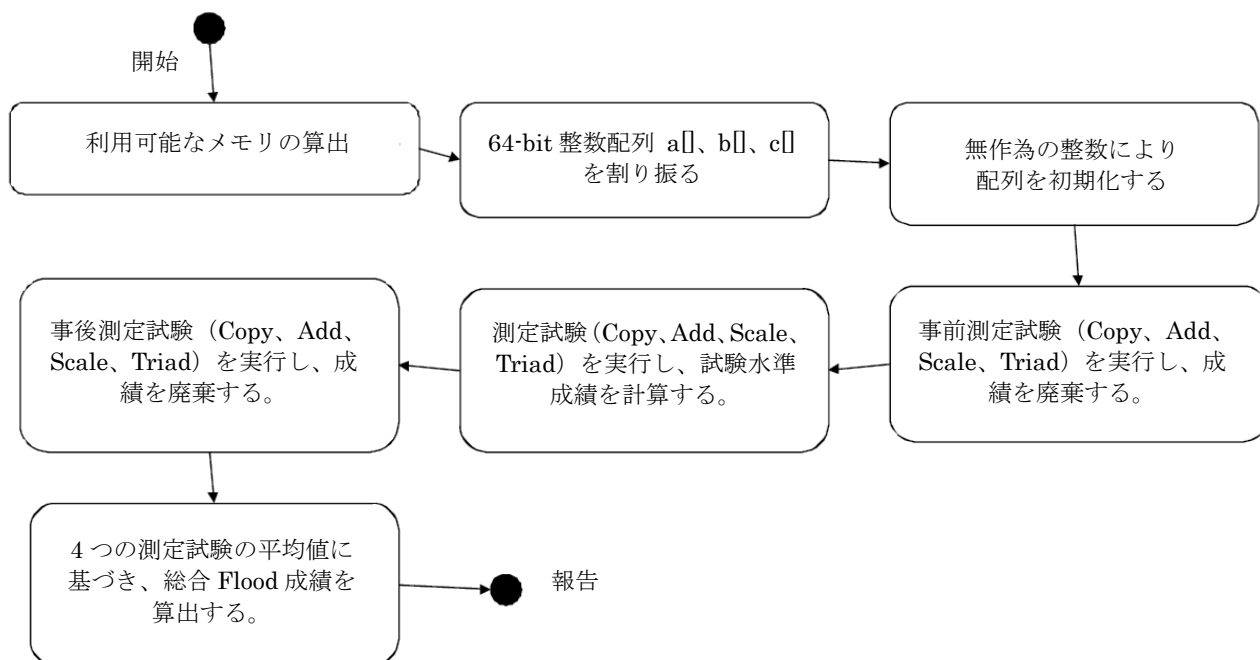
### 10.7.4 所要初期化

Flood は、スレッドに利用可能なメモリ容量を計算し、すべての空き容量を完全に利用する 3 つの 64-bit (長い) 整数配列である `a[]`、`b[]`、`c[]` を作成する。これら配列は、無作為データにより初期化される。帯域幅測定中におけるすべての負荷の同時並行性を確保するために、完全な事前測定試験が、同様の測定期間の前に行われ、測定期間の後には同様の事後測定試験が続く。測定期間の試験結果のみが、Flood 成績の作成に使用される。

### 10.7.5 構成設定値

memory-under-test	「-1 MB」の初期値が、データサイズの自動設定を作動させる。しかし、ユーザーはこの動作を無効にして、JVM に基づいて試験するためのメモリ容量を明確に規定することができる。有効数値は (san 引用符)、「200 MB」、「1.1 GB」、「10000000 B」である。
iterations	Flood は、反復設定値に基づき、メモリ帯域幅試験の回数を内部的に繰り返す。初期値は 100。
debug-level	詳細な診断情報は、デバッグ設定により有効にすることができる。有効数値は、0 = 追加デバッグ情報なし (初期値)、1 = デバッグ情報起動、2 = 詳細なデバッグ情報。
return-bandwidth	Flood が算出した未処理の総システムメモリ帯域幅は、return-bandwidth 設定値を「true」に設定することにより入手可能である。この場合、Flood は、成績の代わりに、メモリ帯域幅測定値を報告する。初期値は「false」。

### 10.7.6 トランザクションコード



## 10.8 メモリ作業負荷: XmlValidate

### 10.8.1 概要

XML 妥当性確認作業負荷は、Java の XML 妥当性確認パッケージ (javax.xml.validation) を実行する。SAX および DOM API の両方を使用し、XML ファイル (.xml) は、XML スキーマファイル (.xsd) に対して、妥当性が確認される。入力データを無作為化するため、XML 入力データ内において注釈付き領域を交換するアルゴリズムが適用される。

XMLValidate におけるメモリ増減は、入力データキャッシング (IDC) として知られるスキームを通じて行われる。IDC において、見込まれる入力データ (ここでは無作為化 XML ファイルデータ) の領域が事前に算出され、その後作業負荷の開始前にメモリ内にキャッシュされる。作業負荷の実行中、特定のトランザクションインスタンス用の入力データは無作為に選択され、オンザフライで算出されるよりもこのキャッシュから取り出される。

### 10.8.2 シーケンス実行方法

段階的測定シーケンス

### 10.8.3 指標

1 秒あたりのトランザクション数 \* キャッシュサイズ \* キャッシュサイズ倍率

### 10.8.4 所要初期化

XML および XML スキーマの両ファイルがディスクから読み込まれ、その後の使用のためにバッファに保存される、初期化時間。(この初期化の完了後、さらなるディスク IO は無い。) IDC 初期化は、すべての見込まれる入力データが事前に計算されメモリにキャッシュされた後に続いて行われる。各入力データについて、ファイルの大きさあるいは複雑性を変更することなく構文解析に多様性を持たせるために、無作為化アルゴリズムは、オリジナル XML データに適用される。

### 10.8.5 構成設定値

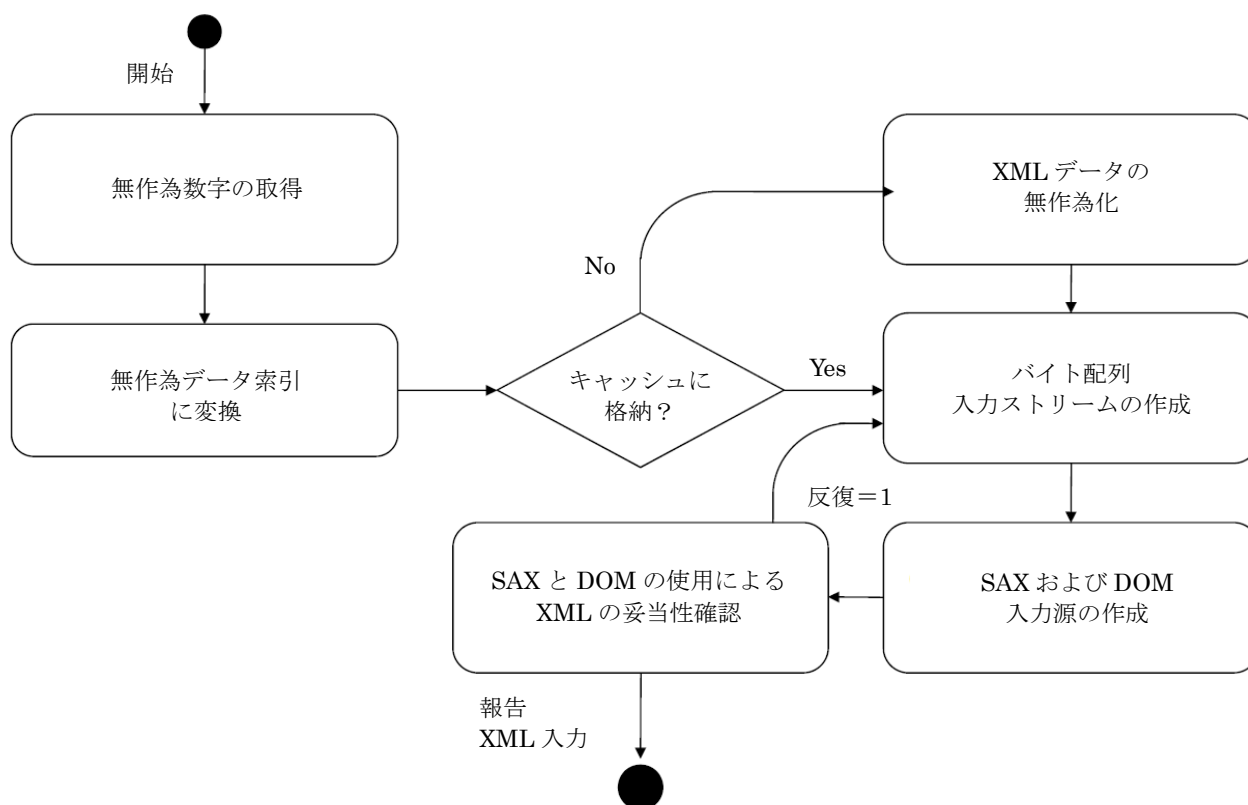
XmlValidate 設定値：

xml-schema-dir	XML スキーマファイルのディレクトリを特定する。
xml-schema-file	XML スキーマファイルの名称を特定する。
xml-dir	XML ファイルのディレクトリを特定する。
xml-file	XML ファイルの名称を特定する。
enable-idc	入力データキャッシング (IDC) を使用したメモリ増減機能の有効化/無効化。False に設定されていなければならない。
iterations	トランザクションあたりの実行数
debug-level	実行中に印刷されるデバッグメッセージ量を管理する数値。
input-generate-iterations	XML ファイル無作為化の反復回数

追加 IDC 構成の設定値：

store-type	キャッシュミスが発生したときに生成されるデータに使用するためのアルゴリズムを特定する。
locality-distribution	入力データの無作為選択に索引をつけるときに使用するための確率分布を規定する。
data-store-size	見込まれる入力データの領域の大きさを特定する。
data-cache-size	入力データキャッシュの大きさを特定する。
data-cache-report-interval	キャッシュヒット／ミス比率に関する出力メッセージの頻度を管理する。
custom-score-policy	キャッシュサイズ設定を反映する特注成績の計算において使用するためのアルゴリズムを特定する。
data-cache-size-scalefactor	DataCacheSizeMultiplierGB 特注成績アルゴリズムにおいて使用するための倍率を特定する。
data-cache-to-heap-ratio	自動キャッシュサイズ設定に使用される、JVM ヒープサイズに対するキャッシュサイズの比率。

### 10.8.6 トランザクションコード



## 10.9 ストレージ IO 作業負荷

### 10.9.1 概要

ストレージ作業負荷には、2つの無作為トランザクション一組と2つ順次トランザクション一組から成る、4つの異なるトランザクションがある。各組は、書き込みと読み込みのトランザクションを有する。

### 10.9.2 シーケンス実行方法

[段階的測定シーケンス] または [固定反復測定シーケンス]

### 10.9.3 指標

成績の名称と成績値が示す意味の定義

### 10.9.4 所要初期化

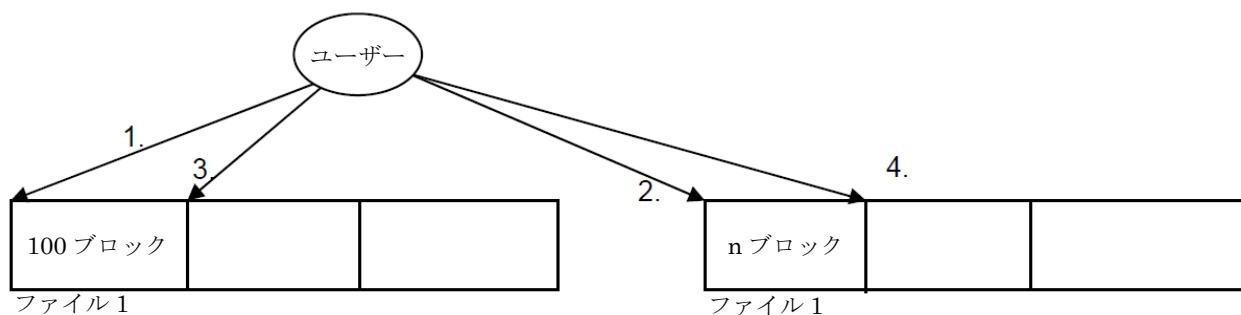
ファイル一式が、トランザクションの実行前に作成される。

### 10.9.5 構成設定値

file-size-byte	ファイルの大きさ
file-per-user	各ユーザーが開くファイル数。
file-path	ファイル保管場所。本例の場合、パスは「D:\data\」であり、ファイルは常に「data」と呼ばれるサブフォルダに存在する。
max-count	次のファイルに対応する前に、1つのファイルにおいて順次トランザクションが利用するブロック量。

例：

```
<file-size-bytes>1000000</file-size-bytes>
<file-path>D:\</file-path>
<file-per-user>2</file-per-user>
<max-count>100</max-count>
```



[図 7: ファイル例(ユーザーあたり 2 ファイル、および max-count=100)]



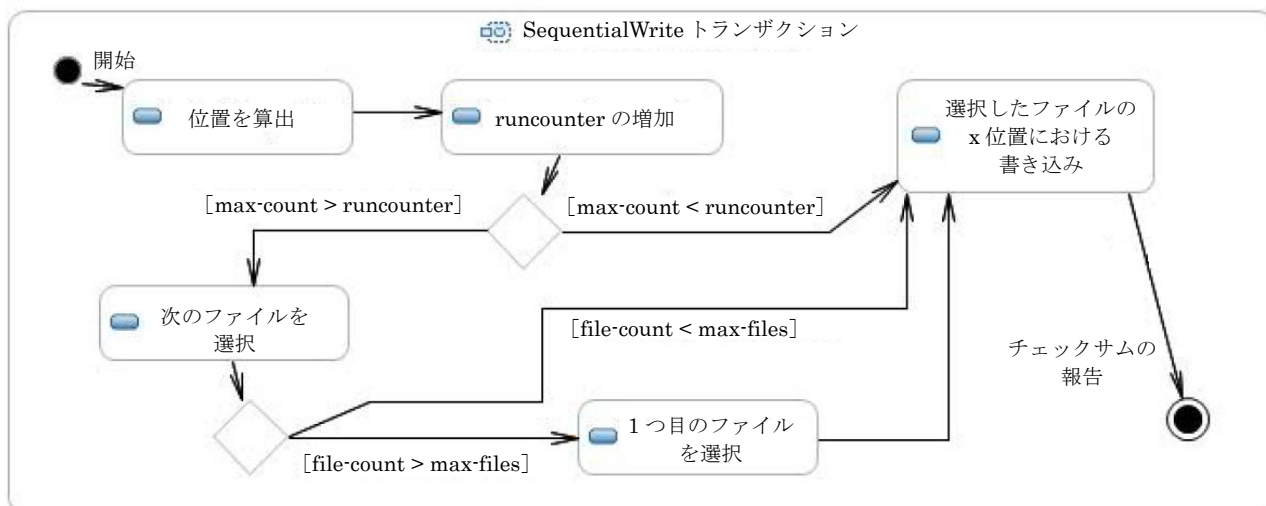
10.9.6 トランザクションコード1-RandomRead



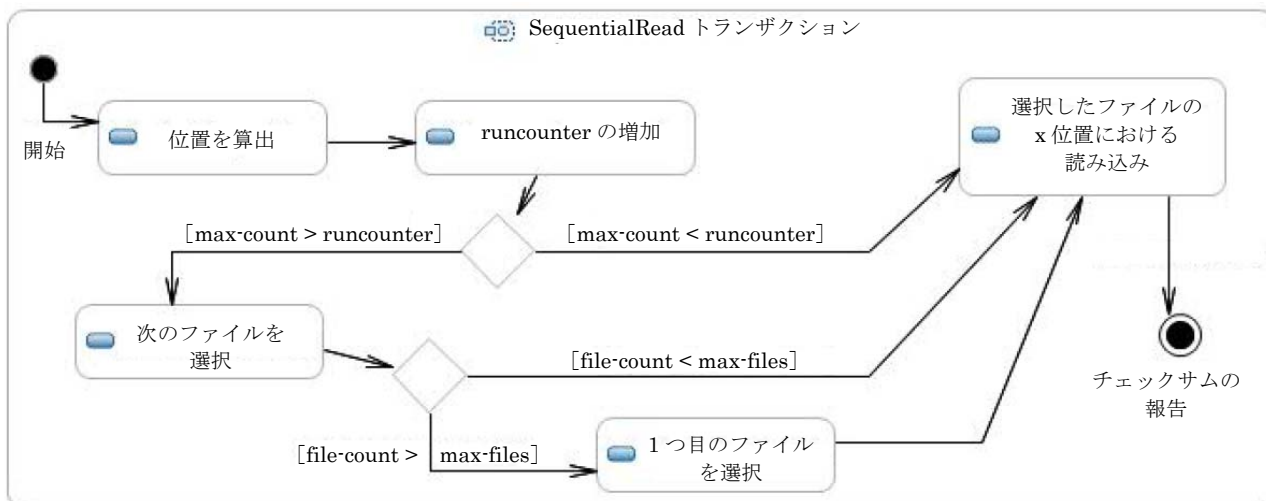
10.9.7 トランザクションコード1-RandomWrite



10.9.8 トランザクションコード2-SequentialRead



10.9.9 トランザクションコード2-SequentialWrite



## 10.10 システム小作業負荷: CSSJ

### 10.10.1 概要

CSSJ は、オンライントランザクション処理 (OLTP) 作業負荷であり、サーバー側 Java アプリケーションを表している。TPC-C 仕様から影響を受けた SPECjbb2005 に基づく、SPECpower\_ss2008 の SSJ 作業負荷に基づいているが、これら作業負荷のすべてに幾つかの違いがあり、CSSJ の結果は、これら他のベンチマークのいずれとも比較することができない。

システム小作業負荷は、UUT の CPU、キャッシュ、およびメモリを働かせる。ピークスループット水準は、システムが 1 秒あたりに実行することができる、上記種類の最大トランザクション数により判断される。任意のシステムについてトランザクションのピーク値が決定されると、小作業負荷は、ピーク (100%) からシステムアイドルまで段階的に低減して実行される。

システム小作業負荷の実行は、そのシステムのプロセッサの種類、プロセッサの数、プロセッサの動作速度、およびメモリサブシステムの帯域幅の組み合わせに依存している。

CSSJ には、以下に示される概算頻度を伴う、6 つのトランザクションが含まれている。

- 新規注文 (New Order) (30.3%) — 新規注文は、システムに挿入される。
- 支払い (Payment) (30.3%) — 顧客に対する支払いを記録する。
- 注文状況 (Order Status) (3.0%) — 現行の注文状況を要求する。
- 配達 (Delivery) (3.0%) — 配達用に注文を処理する。
- 在庫水準 (Stock Level) (3.0%) — 在庫水準の低い最近注文された品目を発見する。
- 顧客報告 (Customer Report) (30.3%) — 顧客に関する最近の活動報告書を作成する。

### 10.10.2 シーケンス実行方法

段階的測定シーケンス

### 10.10.3 指標

1 秒あたりのトランザクション数

### 10.10.4 所要初期化

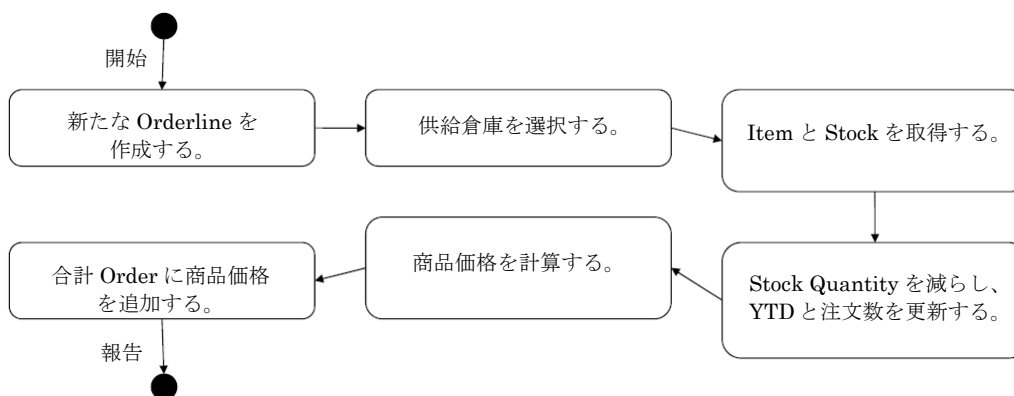
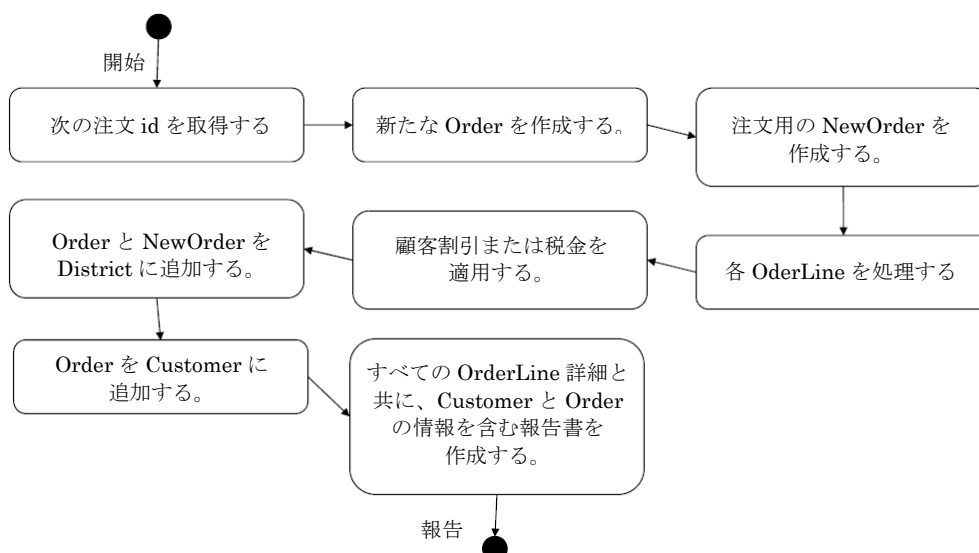
各ユーザーは倉庫 (warehouse) を表している。初期化において、各倉庫には、顧客、初注文、および注文履歴を含む基本データが投入されている。

### 10.10.5 構成設定値

CSSJ 作業負荷には、支持される構成設定値は無い。

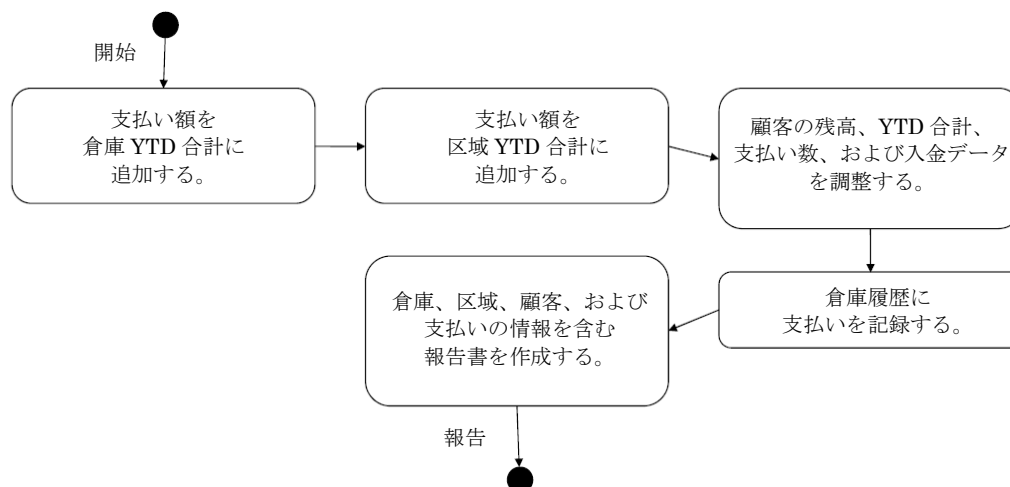
### 10.10.6 新規注文 (New Order) トランザクション

新規注文トランザクション用の入力、ユーザーの倉庫 (warehouse) における無作為の区域 (district) および顧客 id と、5 から 15 の無作為の注文明細数 (orderlines) で構成されている。



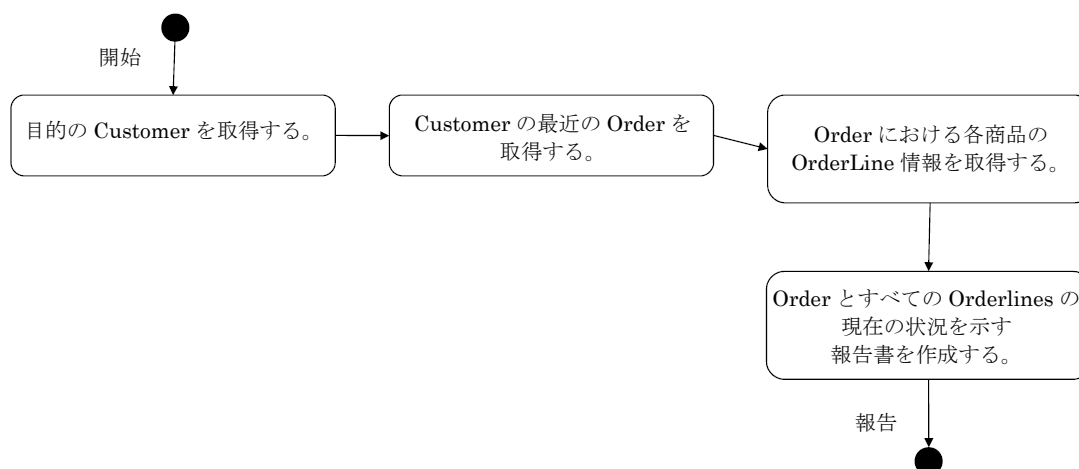
### 10.10.7 支払 (Payment) トランザクション

支払トランザクション用の入力は、ユーザーの倉庫 (warehouse) からの無作為区域 (district)、(ユーザーの倉庫または遠隔ウェアハウスからの) 無作為顧客 id または名字、および無作為支払額で構成される。



### 10.10.8 注文状況(Order Status)トランザクション

注文状況トランザクション用の入力は、ユーザーの倉庫 (warehouse) からの無作為区域 (district)、および無作為顧客 id または名字のいずれかで構成される。



### 10.10.9 配達(Delivery)トランザクション

配達トランザクション用の入力は、任意の運送事業者 id である。

[動作図は作成中]

### 10.10.10 在庫水準(Stock Level)トランザクション

在庫水準トランザクション用の入力は、ユーザーの倉庫 (warehouse) からの無作為区域 (district)、および10~20の間の無作為「低水準」境界値で構成される。



### 10.10.11 顧客報告(Customer Report)トランザクション

顧客情報トランザクション用の入力は、ユーザーの倉庫 (warehouse) からの無作為区域 (district)、および (ユーザーの倉庫または遠隔倉庫からの) 無作為顧客 id または名字で構成される。

