

## PR/SM 環境における CPU 資源監視方法の比較検証報告

加藤 礼基 †

Computer の性能向上に伴い「Server 統合」に対する approach が積極的に行われている。かつて分散化された System が運用 Cost や Software License の観点から見直され、System 統合される case も多い。1 つの資源を複数の論理 Server で共有し有効活用する為の技術も既に長年に渡って提供されており、今後は 1 台の Hardware 筐体上に複数の論理的な Server 環境を実装する構成が増えると予想される。

1 台の Server で稼働していた Application を PR/SM ( Processor Resource System Manager ) の 1 区画で稼働させるような場合「見かけ上の CPU 使用率」が大きく変わってしまう case がある。統合前の Hardware 能力以上の CPU 資源を消費しているにもかかわらず、算出の元となる情報源によっては CPU 使用率として 100% に満たない数値が報告され得る。使用者が従来の資源消費量との比較を求める場合には報告される数値に対して適切な補正を行う必要が生じる。

本稿では PR/SM 環境において数種の Data を CPU 使用率算出の情報源とし、それらの比較検証結果をまとめる。またそれぞれの情報源について適用業務 ( 運用 Tool ) への実装容易性についても評価する。

### A comparative verification report of the CPU resources surveillance method in PR/SM environment

Reiki Katoh †

"Server consolidation" is being addressed positively, partly because of the improvement in computer performance. Systems once distributed are now reviewed from the viewpoints of operational cost and software license. As a result, system consolidation is carried out in many case.

A technology by which two or more servers can share resources has already been offered over many years. One set of hardware will more likely to offer two or more logical server environments from now on.

When shifting to LPAR mode from the BASIC mode, "CPU utilization" may look decreased in some cases. In spite of the increased consumption of CPU resources beyond the hardware can do before integration, a numerical value of lower than 100% may be reported. Depending on the demand of a user, numerical compensation may be required.

This paper evaluates several sorts of data as sources of information in calculating the CPU utilization rate in PR/SM environment. Moreover, it also evaluates each information source's easiness to mount for application business.

#### 1. はじめに

1990 年代に分散化・並列化されたシステムが近年になって一台の Server に集約されるケースを多く見かける。これは運用コスト等の観点における見直しが要因となるケースも多いが、処理装置の能力向上が Server 統合の促進に一役買っているのは間違いない。処理装置の飛躍的な性能向上はメインフレームの世界を見ても明らかである。1990 年代初めに登場した CMOS 型 CPU は 10MIPS 程度の能力しかなかった。それまで主流だった Bipolar 型 CPU に比べて価格性能比の点では大きく向上したものの、適用業務に必要な能力を 1 個の CPU で提供する事が出来ないという側面も持っていた。その為、複数の CPU を効率良く利用する仕組みが研究・開発さ

れ、提供された。LCMP ( Loosely coupled Multi-processing ) の構成を取る並列 Sysplex がその代表である。1CPU が 450MIPS 程度の能力を提供できる現代において並列 Sysplex System は保有資源の有効活用とシステムの高可用性を実現する仕組みへとその役割を変えつつある。

メインフレーム環境では 1 台の Hardware 筐体で非常に多くの Server 実行環境を構築することが可能である。PR/SM ( Processor Resource System Manager ) による区画分割や VM ( Virtual Machine ) による複数稼働環境の実現により 1 つの Hardware 筐体で 100 を超える Server 環境を提供するような構成も増えている。システム統合前には Uni-Processor の構成で稼働していた Server 環境が PR/SM の 1 区画で稼働するように構成変更されるケースは実例として少なくない。この類の Server 統合を行った場合 CPU 使用率を統合の前後で比較

---

† 日本アイ・ビー・エム(株)  
IBM Japan Co.,Ltd

してみると数字上大きな違いを確認する事ができる。元来『CPU 使用率』という言葉の定義自体に曖昧な部分があり、明確な定義が存在している訳ではない。使用者の目的によって求めるものに違いが出てくる。資源管理の立場から見ると該当の Server 環境において想定している CPU 資源の使用量を分母と考えて CPU 使用率を監視したいという要望が多い。HMC ( Hardware Management console )上で表示する SAD ( System Activity Display )上の数値が最も求めるものに近い数値であり、その数値を定期的に操作員に通知する機能を望む声も多い。本稿では Operating System 又は標準的な Sub System で入手できる数種の情報を比較し、定期的に状況を報告する為の情報源としてそれぞれの情報の適応性を評価した。

## 2. CPU の使用状況を監視する為の情報

Operating System から CPU 使用率を算出する為に必要な情報を入手する方法は何通りか考えられるが、今回の試験では比較的一般的な方法で情報を入手し比較検証を行った。情報源として採用したのは以下の 3 種である。

- ・ RMF (Resource Measurement Facility)
- ・ WLM (Work Load Manager)
- ・ LCCA (Logical Configuration Communication Area)

最も代表的な情報源は RMF の持つ数値である。今回の試験においては 30 分間隔の Monitor I の情報と Snapshot 的な Monitor II の情報を入手し比較した。

WLM の持つ数値は元来 CPU 使用率を求める為の数値ではない。システムにおける『重要度』毎の資源使用状況を把握する為の数値である。この数値を CPU 使用率の計算 Source として利用する場合、どのような結果が得られるかを確認する為に、比較

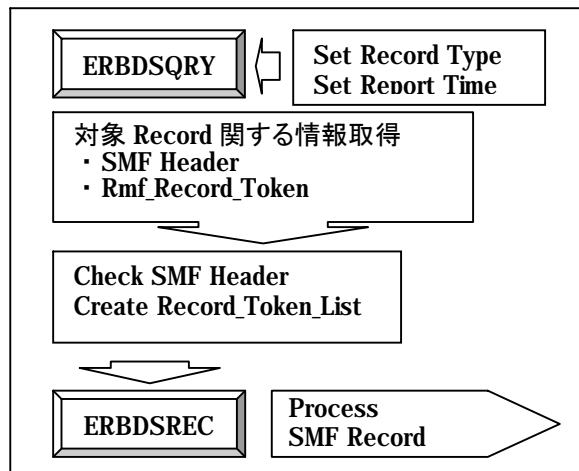


Figure2-1. Sysplex Data Services から情報入手

対象データとして WLM の提供するデータも採用した。MVS 上の Control Block を参照する方法は古くから用いられているオーソドックスな方式である。今回の試験でも LCCA の情報を元に CPU 使用率を算出し他の情報と比較評価した。

### 2.1. RMF 情報の入手方法

RMF の情報は前述の通り Monitor I の情報と Monitor II の情報を収集評価した。具体的な情報の入手方法及び情報源(計算対象 Field)について簡単に説明する。

Monitor I の情報( SMF Type70 )は Sysplex Data Services の機能で Dynamic に RMF から入手出来る。簡単な取得の流れは Figure2-1.を参照して頂きたい。最初に取得対象時間及び対象 Record Type を指定し、Record 情報の照会を行う。次に入手した情報を Check し、必要な Record を選定してから実際に取り出し要求を行う。今回の試験では最初の要求で入手した情報を元に最新の Record の Token のみ選択し、実データとしては 1 Record のみを入手する方式にした。CPU 使用率の算出は入手した Record の SMF70EDT ( SMF70EDT:Logical processor effective dispatch time ) の Field を利用し、

$$\%CPU = \frac{\text{SMF70EDT}}{\text{Interval Time}} \times 100(\%)$$

として計算した。

Monitor I の Record は RMF の Interval 毎に作成される為、Snapshot のデータとしては適切ではないが、今回の試験環境における過去の情報との比較実施も想定し情報源として採用した。

Monitor II の情報については、ERBSMF1 という情報収集の為の module が提供されており、この module を呼び出すことにより情報を入手する事が出来る。ERBSMF1 の呼び出し Image は Figure2-2 の通りである。ERBSMF1 は Type79 の Record 情報を指定した Area にセットすると同時に Parameter で指定した Address に CPU 使用率をセットする。今回の試験では SMF Record の情報ではなくて Parameter 域にセットされた CPU 使用率を比較対象データとする。

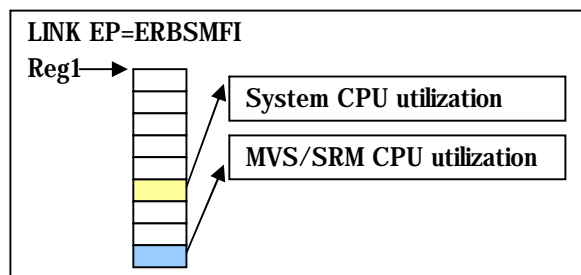


Figure2-2. ERBSMF1 による情報収集

## 2.2.WLM 情報の入手

前述の通り WLMの情報は元来システム全体の CPU 使用率を求める為の数値ではない。WLMから入手出来る情報は WLM が管理している『重要度』毎の CPU 資源使用量である。この情報を利用して CPU 使用率を計算する場合は Uncaptured Time に対する考慮が必要である。但し今回の試験における WLM の数値はあくまで参考データとしての扱いである為 Uncaptured Time に対する補正計算等は特に行わず数値比較を行った。

WLMには IWMWSYSQ という API (MACRO) が提供されており、簡単に WLMの持つ情報の一部を USER が入手する事が出来る。今回の試験では以下の計算式で CPU 使用率を算出した。

$$\%CPU = \frac{\text{SYSI\_SUM60(imp1\&Low)} - \text{Unused}}{\text{SYSI\_SUM60(imp1\&Low)}} \times 100(\%)$$

計算式上の SYSI\_SUM60 の詳細に関しては Figure2-3. を参照して頂きたい。IWMWSYSQ によって得られる情報にはシステム毎に図の 7 個の Entry があり、SYSI\_SU\_ENTRY は該当の重要度及びそれ以下の重要度で消費された資源の情報を持っている。5 段階の重要度に Discretionary 及び unused を加えた合計 7 個の Entry のそれぞれに 1 分間、3 分間、10 分間の消費した CPU サービス量を保持している。

この 7 個の Entries を 1 セットとして Sysplex の全メンバー分の情報を一度の要求でまとめて提供する。今回は Snapshot 的なデータを参照したい為、時間間隔としては 1 分間のデータを利用して計算した。

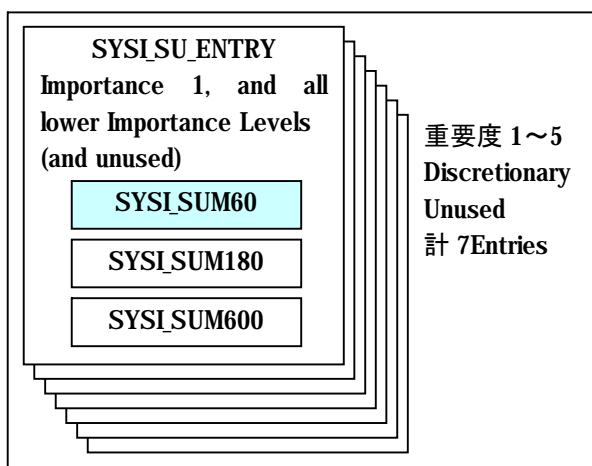


Figure2-3. IWMWSYSQ によって入手できる情報

## 2.3. LCCA の情報を利用した使用率の計算

CPU 毎に Logical Configuration Communication

Area (LCCA) という情報域を持ち、LCCA には累積の CPU wait 時間 (LCCAWTIM) が保持されている。今回の試験では一定時間間隔 (50 秒間) の LCCAWTIM の変化を測定し、CPU 使用率を求めた。計算式は以下の通りである。

$$\%CPU = \frac{\text{Interval} \times \text{CPU 数} - \text{Total wait}}{\text{Interval} \times \text{CPU 数}} \times 100(\%)$$

$$\text{Total wait} = \text{Total}(t②\text{LCCAWTIM} - t①\text{LCCAWTIM})$$

測定時間間隔前後の LCCAWTIM の差を求め、それを Interval の CPU 待ち時間とした。測定の時点で Online 状態であったすべての CPU についてそれらを合算して合計 CPU 待ち時間 (Total wait) とした。

## 3. 試験環境

今回の測定は zServer (2064-2C3) を LPAR Mode で稼働させた 8 区画中の 1 区画で行った。Operating System は z/OS V1.4 で主要 Sub System としてデータベース管理システムが稼働している。

該当区画の CPU weight は 10 で、8 区画すべての weight の合計は 165 である。該当区画の想定能力は 50MIPS 強となる計算である。適用業務はかつて 50MIPS 程度の処理装置で CPU 資源を 100% 近く消費していた Application をそのまま稼働させている。試験の目的が Snapshot 的な CPU 使用率としての情報源の評価であるから RMF Monitor I については Interval の 30 分毎に取得したが、それ以外の情報は毎分取得した。測定 Tool は Assembler で Coding され、測定以外の処理は行っていない為 Tool による CPU 資源等への影響は無視できるレベルである。

## 4. RMF Monitor I の測定結果

最初に RMF Monitor I の測定結果について検証を行った。該当 Server が BASIC mode で稼働していた当時に取得したデータを参考値として、今回の測定で入手したデータと比較する方式をとった。稼働していた環境や処理対象のデータ量の違い、CPU 使用率の計算方法が BASIC mode と LPAR mode で違いがある点等を考慮すると純粋に数値を比較する事は出来ない。あくまでも参考値としての比較である。その結果が Table4-1 である。Weight 補正値は今回の測定結果に対して 165/10 つまり (Total weight) / (Partition weight) を乗じた値である。この結果から判断すると約 50MIPS という想定能力に対する使用率が必要な場合には実際の測定結果に対して適切な補正を施す事で期待する数値が得られると考えられる。

	Monitor I	Weight補正	参考値
8:00	2.44	40.26	43.48
8:30	3.97	65.51	80.57
9:00	2.79	46.04	40.51
9:30	2.40	39.60	42.37
10:00	2.69	44.39	43.05
10:30	4.02	66.33	67.66
11:00	2.77	45.71	46.16
11:30	2.94	48.51	49.48
12:00	4.35	71.78	71.06
12:30	5.84	96.36	95.40
13:00	4.83	79.70	81.29
13:30	3.97	65.51	68.78
14:00	5.62	92.73	91.80
14:30	6.29	103.79	93.41
15:00	7.00	115.50	99.33
15:30	6.54	107.91	96.04
16:00	5.00	82.50	86.63
16:30	5.81	95.87	94.91
17:00	3.35	55.28	56.93

Table4-1.RMF Monitor I の測定結果

### 5. RMF Monitor I, II, WLM の比較

RMF Monitor I の測定結果は期待する数値として利用できる事が確認できた。但し Monitor I の情報は RMF の Interval 毎の情報であり、Snapshot の情報として使用する事は出来ない。他の Snapshot 的に取得出来る情報源と RMF Monitor I における測定結果を比較する事で、他の情報源の適応性を評価する。

RMF Monitor II から Parameter 指定 Area に返される値としては 2 種類の CPU 使用率(Figure2-2 参照)が得られるが、計算式は次の通りである。

2 種類共に共通の計算式として

$CPU\ Busy\% =$

$\frac{Total(CPU\ Time) * 100}{Total(Interval\ Time) / CP\ 数}$

が用いられる。2 種類の数値の違いは Total(CPU Time)の計算方法の違いにより生じる。

<MVS Busy%>

$Total(CPU\ Time) = Total(Interval - Wait\ Time)$

< LPAR Busy% : Wait Completion=YES >

$Total(CPU\ Time) = Total(Dispatch\ Time - Wait\ Time)$

< LPAR Busy% : Wait Completion=NO >

$Total(CPU\ Time) = Total(Dispatch\ Time)$

この 2 種類のデータはシステムの状況を簡易的に把握する Product である SDSF ( System Display and Search Facility ) 等でも取得され画面上に表示されている情報である。2 種類のデータはそれぞれ別の目的で参照される。今回の CPU 使用率という観点では LPAR Busy percent がターゲットになるが、MVS Busy percent も比較対象データとして採用した。RMF Monitor I の測定結果と比較する為、毎分の取得結果の 30 分毎の平均値を求めた。平均値を利用した点は WLM についても同様である。これらのデータの比較が Figure5-1 のグラフである。

グラフを見ると MVS Busy %が他の測定結果と違う値を示しているが、計算式からも分かる通りこの値は MVS ( Operating System ) の観点からの Busy %を表し、CPU のボトルネックを見る時に使用するもので、今回の目的に使用する情報としては適切ではない。他の 3 つの情報源を使用した測定結果に関しては比較的近い値を示している。

WLMの情報に関しては Uncaptured Time が存在す

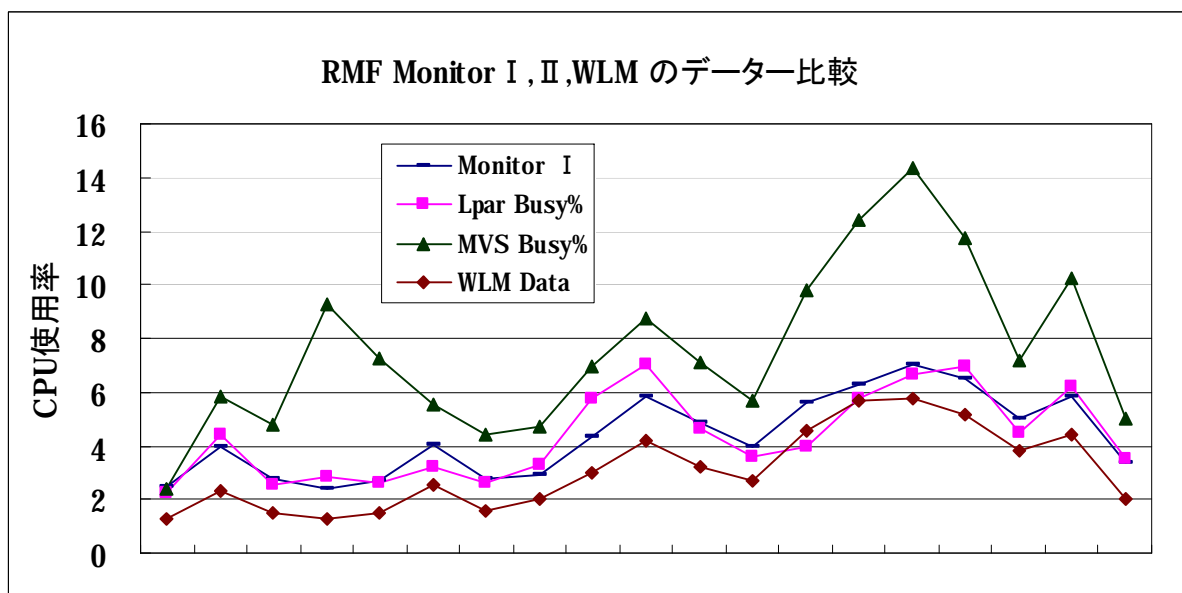


Figure5-1.RMF Monitor I, II, WLM の比較



る為、全体的に他の測定結果と比較して低めの数値が出ている。元々Uncaptured Time の影響は想定していたので予想通りの測定結果であると言える。Snapshot の CPU 使用率自体、システム使用状況の目安として参考にする数値であるから厳密な正確さを要求する事自体に無理が有り、且つあまり意味の無い事である。しかし最終的に Weight 値による補正を施す事を考えると Uncaptured Time の影響は数値的に無視するには大きすぎる感がある。一方 RMF Monitor II のデータにも多少の誤差は含まれている。ERBSMF1 によって指定域に返される値は整数値であるので、整数化する段階で誤差が含まれる事になる。ただ 30 分間の数値を平均した値は RMF Monitor I の測定値と非常に近い値を示しているので整数化する段階で発生する誤差は Snapshot の資源使用状況のみをみる数値として利用するという前提では無視出来る。

## 6. LCCA の情報

LCCA の情報を元に CPU 使用率を計算する事の評価は RMF Monitor II の測定結果と比較する事により行った。測定時間のうち任意の 30 分を数ショット取り出し数値の比較を行った。Figure 6-1 はその内の 1 ショットをグラフにしたものである。

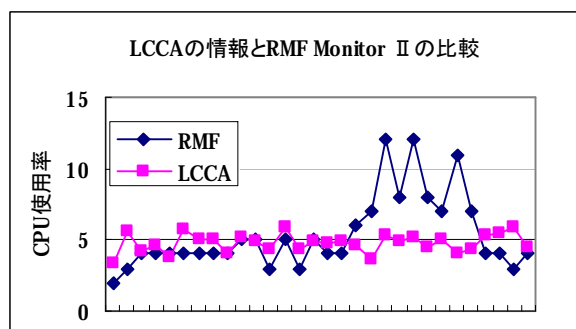


Figure6-1.LCCA の情報とRMF Monitor II の比較

LCCA の情報を利用する上で前述の通り測定の時間間隔を 50 秒間とした。Snapshot としては比較的長い時間であるが、その影響が RMF のデータよりも数値のばらつきが少ない。数値のばらつき度合いには違いがあるものの、得られる値自体を RMF Monitor II のデータと比較してみると大きな違いは認められない。各ショットにおいて 30 分間の平均値を計算しても 2 種のデータに殆ど違いは見られなかった。この比較結果から判断すると LCCA の情報も数値としては利用可能であると判断できる。

## 7. 情報源の評価

3 種の情報源について測定結果から数値的な評価を行ってきた。システム運用 Tool に利用する情報源に対しては数値の信頼性が求められるのは当然であるが、それ以外に実装容易性や拡張性も求められる。ここでは、これまでに述べてきた数値の信頼性とは別の観点でそれぞれの情報源を評価する。

### 7.1. WLM 情報の価値

CPU 使用率を算出する為の情報源としては WLM の情報は適切とは言い難いが、Snapshot で情報を取得するデーターとしては十分に利用価値がある。WLM の持っている数字は『使用量』である。またそれぞれの重要度毎の数字を持っているので重要度毎に稼働する業務が分かっているような環境においてはいろいろな利用方法が考えられる。例えば WLM の Policy でオンライン業務を重要度 2、バッチ業務を重要度 3 というように設定してある環境においては WLM の情報を利用して Snapshot でオンライン業務とバッチ業務の使用状況を大まかに把握できる。また報告される情報は Sysplex 全体の情報であるから、Sysplex 環境においてどのメンバーがどの程度の資源使用状況かを把握するには最適の情報となる。RMF は「Sysplex Data Services」によって Sysplex Wide の情報を収集出来る。但し、これは RMF Interval 毎の情報であり Snapshot の情報ではない。他の情報に関しては Sysplex 全体の情報を 1 メンバーから集める事は出来ない。Sysplex 全体で Snapshot の使用状況を把握したいような場合には WLM の情報が唯一の選択肢となる。実装という面でも提供されている API は非常に使いやすい。他に従量制のソフトウェア課金等を検討する場合にも有用なデーターになると思われる。WLM が提供する情報は「CPU サービス量」という値で報告されるので CPU 使用量を Monitor する情報源としては最も活用しやすいデーターである。

### 7.2. RMF Monitor II の評価

RMF Monitor II のデーターは該当システムの Snapshot における CPU 使用率を入手する情報源としては最も使いやすく且つ拡張性がある。使用方法は単純な Module の Call であるから Assembler だけではなく、C 言語や PL/I から簡単に利用できる。また今回は CPU 使用率のみを評価対象としたが、他の情報も容易に入手出来る点で拡張性においても優れているといえる。例えば Paging に関する情報は CPU と同様に Parameter で指定する領域に数値が返される。ユーザーは大規模な Program の変更をする必要なく新しい情報を利用できる。更に SMF

Type79 のデーターを入手できるので、Device の状況や各 Address Space の状況も必要に応じて入手出来る。それにより CPU 以外の資源の状況を適時把握したいというような要求にも簡単に対応することが出来る。RMF が情報を提供するのでユーザーは Operating System 等の内部変更を意識する必要が無い。この事から保守容易性の観点でも優れている。

### 7.3. LCCA 情報の評価

CPU 使用率を算出する情報として数値の信頼性という面では全く問題ない。但し実装容易性や拡張性という面では他の方式に比べて劣っている感がある。

実装容易性という面で見ると、他の方式が MACRO やLINKによる module 呼び出しで簡単に利用できるのに対して LCCA を利用する場合は利用者が各 CPU の状況を確認しながら有効な情報を収集するという作業を行う必要があり、Program Coding も他の方式よりは遙かに複雑である。

拡張性の面では評価が難しい。LCCA のみを参照するという前提であれば「殆ど拡張性がない」と言えるが「MVS の Control Block 参照する方式」という風に考えると、新たに情報を追加する事自体は多くの場合可能である。

例えばRMF Monitor II のcaseでも例に挙げた Pagingの情報を入手しようと思えばRCE ( Real Storage Manager Control and Enumeration Area )を参照すればRCETOTPI, RCETOTPOというFieldから Pagingに関する情報を取得する事が出来る。

「CPU使用率以外の情報を入手可能か。」という質問に対しては「Yes」であるが、「簡単に入手可能か。」という質問であれば答えは「No」である。

Operating SystemのControl Blockを参照する方式は古くから非常に良く利用されている方式だと思われるが将来的な拡張性や使いやすさという点では他の方式よりは劣っている。

### 8. CPU 使用率について

CPU 使用率は一意に決められる数字ではない。RMF が提供する CPU 使用率にも LPAR view 使用率と MVS view 使用率があるように、利用する目的によって複数の数値が存在し得る。今回の測定に利用した区画の特定の測定時間(Figure4-1 参照)における平均のCPU使用率は H/W 全体の能力を分母とするなら 4%であるが、該当区画における 50MIPS という割り振り想定資源を分母とするなら 70%である。Server 統合の環境において資源全体の有効活用に対してアプローチするなら 4%という数字

を使用率ととらえるのが適切である。一方、該当区画に対して資源見積もりを行っており、見積もり資源量に対する使用率を把握しておきたいなら 70%という数値の方がわかりやすい。

目的によって要求される数値が異なるが、殆どの場合において Operating System から得られる情報に対し適切な補正を施す事で CPU 使用率として期待する値を得られる事が予想出来る。ユーザーが必要とする観点での CPU 使用率を定期的に通知するような仕組みは今回調査した情報源を利用することで構築可能である。今回の測定は zServer 上での MVS ( z/OS )における環境で実施した。将来的には Linux 等の区画においても同レベルの情報が取得できる仕組みが望まれる。今後の zServer の機能拡張に期待したい。

### 参考文献

- [1] MVS Data Areas, Volume 3 (ITV - RCWK)  
GA22-7583-05
- [2] Resource Measurement Facility  
Programmer's Guide  
SC33-7994-03
- [3] MVS Programming  
Workload Management Services  
SA22-7619-05
- [4] MVS System Management Facilities  
SA22-7630-07