

## 論理分割環境における CPU 資源最適化の検討と評価

加藤 礼基 †

IBM zServer の論理分割機能を利用したシステム環境における CPU 資源最適化の考え方は複雑化している。WLC(Workload License Charge)の登場によりエンドユーザーの求めるものが多様化している為である。かつてはシステムの資源を出来るだけ効率よく複数のシステムで共有し「保有する資源を如何に使い切るか」ということが最適化へのアプローチであった。現在に置いてそれが非常に重要なアプローチである事は変わらない。それに加えて WLC の浸透により「業務要件を満たすのに必要な資源のみ与える」事へのアプローチが必要になってきている。ワークロード使用量に応じたソフトウェア料金が請求される為である。適正な資源量で抑止する方法を検討しテストコードによって検証を行った結果、簡単なユーザープログラムである程度の制御が可能である事が確認出来た。

### Evaluation and assessment of CPU resources optimization in the Easy-Sysplex systems

Reiki Katoh †

There are so many systems that are using PR/SM of IBM zServer. In those environments, end users' demands have been changing since WLC (Workload License Charge) has been available. Once "How to use up retained resources" was a proposition. It was important to share system resources with multiple system-images as effectively as possible. The present subject is what "providing just enough resources to fulfill system requirements". The reason is that the fee is charged according to the amount used.

I examined how to supply only required quantity, and I checked that suitable control could be performed by the user program.

#### 1. はじめに

特定の能力を有する処理装置が存在した場合に、如何にその保有能力を使い切るかという事がこれまでの課題であった。複数の OS イメージでシステム資源を共有するような環境においては資源を効率的に割り振る仕組みが重要で、その為の機能として IRD(Intelligent Resource Director)等が提供されている。WLC(Workload License Charge:ワークロード使用量料金)の登場により更にユーザーの要求の幅が広がった。「ハードウェアの保有能力を使い切る事により、ハードウェアに対する投資対効果を引き上げる」という考え方に加えて「本当に業務要件を満たす為に必要な資源のみ消費させて、ソフトウェア費用を抑止する事で IT(Server)全体の投資対効果を高める」というアプローチが選択肢として追加されるからである。当報告書においては前述の新たなアプローチを実現する為に、ユーザーが適用出来る何種かの手法を検証し報告する。

#### 2. 標準機能による資源使用量の制御について もともと zSeries の標準として提供される機能や

OS(z/OS)の標準として提供される機能があり、ユーザーがその機能のみで要件を満たすことが出来ればそれが最も望ましい。本章では標準に提供される機能について簡単に説明する。

##### 2-1. ハードウェア・キャッピングによる制御

この手法は特定の区画の使用量を制限する手法としては非常に古く、ソフトウェア使用量を制御する方式というよりはむしろ開発区画或いは試験用区画が他の区画(例えば本番区画)に影響を与えないようにする機能として用いられている。ハードウェア全体の Weight 値に対して該当区画に割り振られた Weight 値の割合以上は仮にシステム資源に余裕があっても使用することは出来ない。ソフトウェア使用量を制御する仕組みの部品として利用することは可能であるが、この機能だけでユーザーの要求を満たすことは困難である。

##### 2-2. ソフトウェア・キャッピングによる制御

システム環境によってはソフトウェア・キャッピングによる使用量制御が最も適切であり、多くの場合にはこの機能によってユーザーの要求を満たすことが出来る。「特定の区画におけるソフトウェア使用量を制御する」という要求だけであればソフトウェア・キャ

† 日本アイ・ビー・エム(株)  
IBM Japan ,Ltd

ッピングが完全にそれを満たしてくれる。もともとソフトウェア・キャッシングはユーザーのソフトウェア使用量を一定の値で抑えたいという要望を実現するために提供される機能である。従って後述(3章)のような複数システムイメージ全体での制御が要求されないケースにおいては最も適切な方法であると言える。尚、当報告書で使用する「ソフトウェア使用量」という言葉は瞬間的な使用量という意味ではなくワークロード使用量料金制度で料金計算の基準となる使用量、すなわち 4 時間平均の使用量を示している。

### 2-3. ワークロード・マネジャーによる制御

ワークロード・マネジャーは資源グループという概念を持っており、特定の資源グループに対して資源の割り振りを制御する事が出来る。例えば「業務処理Aの為のバッチジョブ群」というまとまりに対する資源の割り振りを特定の値でキャッシングする事が出来る。またキャッシングの有無やその値は動的に変更可能である。ワークロード・マネジャーの持つキャッシング機能によってシステム全体のソフトウェア使用量を制御する事は困難であるが、より細かい制御の為の仕組みとして利用する事が考えられる。

### 3. ソフトウェア・キャッシングによる制御と課題

ソフトウェア・キャッシングによる制御は多くの場合有効であるが、複数区画のピークがずれるようなケースではソフトウェア・キャッシングによる制御だけでは不十分な場合もあり得る。

具体的な例を示す。それぞれの区画にソフトキャッシングを設定すれば全体としてのソフトウェア使用量は各区画に対する設定処理能力の合計を超える事は無い。「ソフトウェア使用量を特定の設定値を越えないようにする」という観点ではソフトウェア・キャッシングの機能により実現出来る。但し「設定値の範囲内で最大限にシステム資源を活用する」という観点では必ずしも充分とは言えない。場合によってはシステム資源に空きがあり且つ請求対象基準に達していないような状況においても資源割り振りが行われない可能性を残す事になる。具体的には Figure3-1、Figure3-2 のような資源使用状況の 2 つの区画のケースである。ユーザーの要求はシステム全体の使用量を 90MSU (MSU: ワークロード使用量料金において料金請求の基準となる単位) で抑える事だと仮定する。その為に LPAR1、LPAR2 のそれぞれに対して 30MSU、60MSU のソフトウェア・キャッシング設定を施す。全体の使用量はそれぞれの設定値の合計を越えない為、この設定で全体の使用量が 90MSU を越える事は無い。一方 90MSU という

全体への設定量を使い切っているか否かという点を 2 区画合計の使用状況グラフ (Figure3-3) で確認してみる。

#### LPAR1 : Defined Capacity = 30MSU

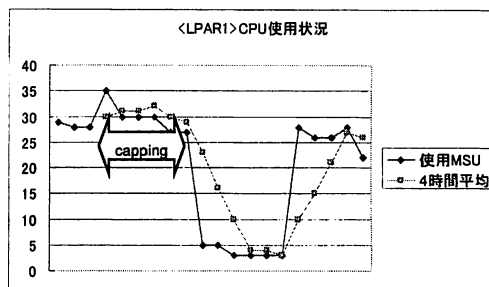


Figure3-1. LPAR1 の使用状況

#### LPAR2 : Defined Capacity = 60MSU

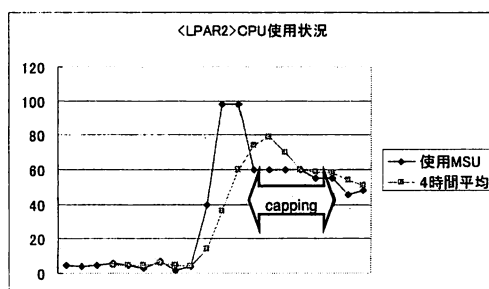


Figure3-2. LPAR2 の使用状況

Figure3-1 をみると LPAR1 は早い時間帯に設定値に達してしまっただけにキャッシングされている事が分かる。また Figure3-2 をみると LPAR2 は LPAR1 とは違う時間帯でキャッシングされている事が確認出来る。

#### LPAR1 + LPAR2 : MSU ≤ 90

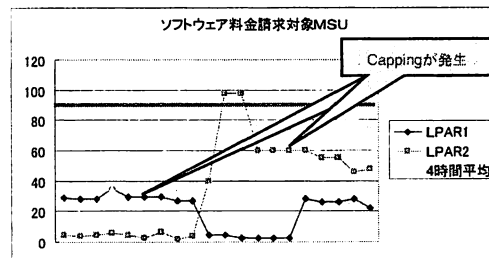


Figure3-3. LPAR1・LPAR2 合計の使用状況

LPAR1、LPAR2 を個別に見るとそれぞれが違う時間帯でキャッシングされている。全体の使用量としては Figure3-3 のようになる。90MSU までを使わせたいというユーザーの要望に対して 90MSU を使い切っていない状況において LPAR2 がキャッシングされてしまっている。ソフトウェア・キャッシングはそれ

それぞれの区画に対して機能する為である。ピークが重なってそれぞれのキャッピングの時間帯が重なった場合には全体としても90MSUという使用量になる為、ユーザーは90MSU分の使用量に見合うソフトウェア料金を払う必要がある。90MSU分の料金を支払いながら、前述の通り90MSUを使い切れないケースが存在する。このように複数システムの合計の使用量を完全に制御出来ない事がソフトウェア・キャッピングの課題と言える。

#### 4. CPU 資源最適化を実現する仕組みの検討

資源最適化を実現する為には定期的に情報を収集して全体の使用状況を把握する仕組みと、ユーザーの定義値(閾値)に達した(或いは近づいた)場合に適切なアクションによる対応を行う仕組みが必要になる。前者の使用状況(MSU)を把握する情報元としてはWLMのもつ情報を利用する。WLMのAPIを利用すればシブプレックス各メンバーの資源使用状況を入手する事が出来る。入手できる情報は、それぞれのシステムで消費された資源量(サービス・ユニット数)である。このサービス・ユニットに関しては、重要度毎の使用量や過去の単位時間あたりの使用量等のある程度詳細な情報が収集可能である。何パターンかの時間間隔が提供されていて時系列の使用資源の変化も確認する事が出来る。それらの情報をWLMモニターで収集して利用する。また、機能拡張の為にRMFのデータを収集する仕

組みも実装する。これはあくまでも機能拡張の為のデータであって主とする監視対象の情報はWLMから入手するデータである。このデータにRMF Sysplex Data Serviceは以下の追加情報を提供する事が可能となる。

- ・ Sysplex 外のシステムの使用状況
- ・ 使用しているサービス・クラスに関する情報

これらの拡張された追加情報(RMFの情報)によりサービス・クラス毎の資源使用状況を把握する事が出来る。前述のWLMの資源グループに対する制御(2-3で説明)と組み合わせる事で非常に細かな制御が可能になる。詳細は後述する。

モニター部分の動きとしては3分毎にWLMからの情報を収集し、蓄積されたデータを更新する。RMFに関してはレコードの識別情報を取得して前回取得したタイミング以降の更新があればレコード内容を取得する。これはRMFのレコードの更新タイミングがシステムの設定によって一定でない為である。モニター本体の動きをトレースする事が出来るように基本的に情報の更新毎にSMF Recordを書き出す機能を実装する。

情報収集元	収集情報	情報収集間隔	情報保管先
WLM	WLM資源使用情報	3分	SMF Type nnn Subtype1 (3分)
RMF(Data Server)	SMF Type70 & 72 PR/SM Information WLM Information	3分	SMF Type nnn Subtype2 (3分)

Figure4-2. モニター部分機能概要

後者の制御部分に関しては2種の方法による使用

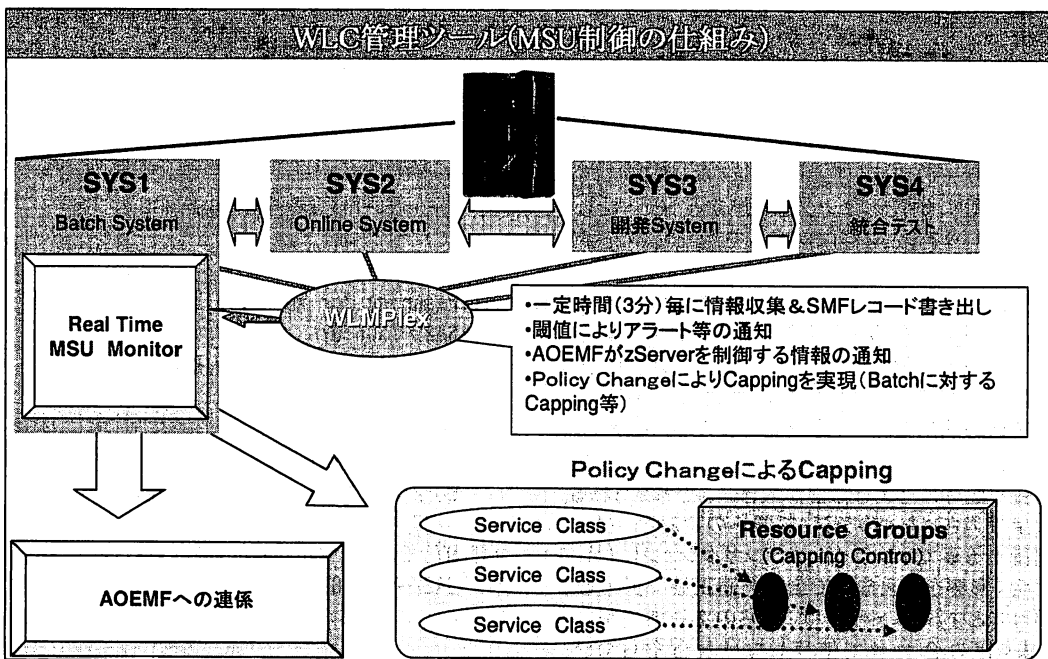


Figure4-1 CPU 資源最適化を実現する仕組みのイメージ

量の制御を検討している。Figure4-1 から Real Time Monitor からの命令が 2 系統ある事が確認出来る。AOEMF(Automated Operation and Environment Monitor Facility)に対する命令と WLM(Work Load Manager)に対する命令の 2 系統である。AOEMF はシステムの自動運転を実現する為のツールであり、常にオペレーティングシステムのコンソールを監視している。Monitor からは OS のコンソールに対して AOEMF が判断出来るメッセージを出力する事で、AOEMF に対してアクションを依頼する事が出来る。AOEMF はハードウェアを直接制御する機能を有している為、自分がアクションすべきメッセージを確認したら Capping の ON/OFF 等の操作を実行する事が出来る。具体的には以下のようなメッセージを判断してハードウェアを制御するようにしている。

```

WLCA001| AOEMF COMMAND PROCESSING TO SYS1 : SET MSU
= 10
WLCA002| AOEMF COMMAND PROCESSING TO SYS1 : SET
WEIGHT = 10
WLCA003| AOEMF COMMAND PROCESSING TO SYS1 : SET H/W
CAP ON
WLCA005| AOEMF COMMAND PROCESSING TO SYS1 : SET S/W
CAP ON
  
```

AOEMF への命令を実現する事で、ハードウェア的に使用量を制限させる事が可能となる。但しいきなりハードウェア的に該当区画に制限をかける為にアプリケーションへの影響は無視出来ない。その為、クッションになるような制御が求められる。

WLM による制御は AOEMF によるハードウェア制御の前段階で資源割り振りの適正化を実現する方法として適している。具体的にはソフトウェアの使用量がユーザーの設定した閾値に近くなった時に該当区画に対するキャッピングを実施する前処理として、特定のバッチジョブ群に対するキャッピングを実施する等の細かな制御を可能にする。多くのサンプルデータでシミュレーションした結果、特定のバッチジョブ群に対してのみ割り振る資源を制御する事で区画(OS Image)単位のキャッピングをする事無く全体の使用量を制御出来る事が確認出来ている。これは実際の適用業務においてはオンライン業務よりもむしろ瞬間的なバッチジョブの大量投入による資源消費が全体に与える影響が大きいためと考えられる。大量投入されたバッチジョブ群への資源割り振りを制御する事でオンライン・トランザクションに対しては全く影響を与えないで全体を適正値に抑える事が可能となる。ユーザーはこれらの 2 系統の制御をうまく組み合わせ使用しなければならない。Figure4-3 は今回検証試験に用いた Program で使

用した制御用のパラメーターである。これらは今後も試験を重ねながら Tuning されていく必要がある。

パラメーター(x:0~5)	説明
L0MSUは標準のPolicyに戻す基準値を指定 L0POLは標準のPolicy	
LxMSU	閾値のMSUを数字で指定
LxACTION	SMF: SMF Only WTO: SMF & WTO POLCHG: SMF & WTO & Policy Change AOEMFCTL: AOEMFへの命令
LxPOL	閾値に達した場合のPolicyを指定

Figure4-3. 制御用パラメーター

## 5. テストプログラムによる検証と評価

今回検討したモニター及び制御の仕組みを簡易プログラムにより検証したところ、機能としては期待する動きが確認出来た。尚ソフトウェア使用量の基準となる MSU はハードウェアの能力とは直接関係が無い為、内部にテーブルを持つ等の対応が必要であった。(Figure5-1 参照) 仕組みとしての実現性は確認出来たので、今後はどのタイミングでどのような制御をするかというユーザー・デザインを支援する観点で改良を加える予定である。

		CPU数	SU/SEC	MSU	SU/MSU
z/900	2064-2C1	1	14692.38	52	282.55
	2064-2C2	2	13961.61	100	279.23
	2064-2C3	3	13377.93	144	278.71
	2064-2C4	4	13082.58	184	284.40
z/990	2084-301	1	21857.92	70	312.26
	2084-302	2	20752.27	132	314.43
	2084-303	3	20075.28	191	315.32
	2084-304	4	19559.90	248	315.48
z9	2094-601	1	23774.15	65	365.76
	2094-602	2	22857.14	127	359.96
	2094-603	3	22191.40	184	361.82
	2094-604	4	21621.62	240	360.36
	2094-701	1	29520.30	81	364.45
	2094-702	2	28368.79	158	359.10
	2094-703	3	27538.73	229	360.77
	2094-704	4	26845.64	298	360.34

Figure5-1. 【参考情報】SUとMSUの関連

## 参考文献

- [1] Planning for Workload License Charges  
SA22-7506-07
- [2] Using the Subcapacity Reporting Tool  
SCRT Version II  
SG24-6522-14
- [3] MVS Programming  
Workload Management Services  
SA22- 7619-05
- [4] MVS System Management Facilities  
SA22-7630-07