


[ホーム](#) | [製品](#) | [サービス & ソリューション](#) | [サポート & ダウンロード](#) | [マイアカウント](#)

[DB2 Developer Domain](#) > [製品別技術情報](#) > [DB2いろはがるた](#) >

DB2いろはがるた

DB2いろはがるた



第45回

「も」・モニターするなら、スナップショット

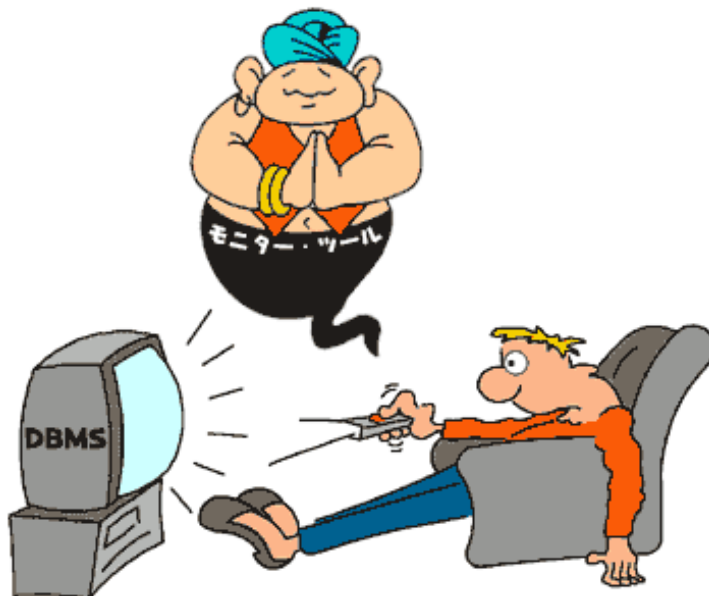


執筆者

春野 さくら

「DB2いろはがるた」を執筆するために参上した、なぞの女性。日本の古典文学を愛する。

DBMSを使っていて、あれ、ちょっと今日は遅いな？とか、なかなか返事がかえってこないけど、どうなっているんだろう？と思う時がありますよね。そういう時、頭の中は？？？で混乱してしまいます。そういう時のためのモニターツールをDB2では提供しています。DBMSの中でなにが起きているのかを探るツール、それがDB2のスナップショット・モニターとイベント・モニターなのです。この2つのツールは通常オフになっていて、起動されていません。なぜなら、このツールを起動すると、何がしかのオーバーヘッドがかかってクライアント・アプリケーションのパフォーマンスに影響を与えるので、通常はオフにしておいて、何か問題が起こった時に、オンにした方がいいからです。問題判別の目的に応じて、この2種類のモニターツールをお使いになることをお勧めします。



スナップショット・モニターはDBMSの内部情報を累積した数で、提供します。例えば、スナップショット・モニターをオンにした以降、そのデータベースに接続した総ユーザー数の数や、データの読み取りや、書き込みが何回起こったかのような数字です。下の表はスナップショット・モニターで取得できる情報を表しています。

グループ	提供される情報	モニター・スイッチ	DBMパラメーター
分類	使用されたヒープ数、オーバーフロー、分類パフォーマンス	SORT	DFT_MON_SORT
ロック	保持されているロックの数、デッドロックの数	LOCK	DFT_MON_LOCK

表	測定活動(読み取られた行数、書き込まれた行数)	TABLE	DFT_MON_TABLE
バッファ・プール	読み取りおよび書き込みの回数、所要時間	BUFFERPOOL	DFT_MON_BUFPOOL
作業単位	開始時刻、終了時刻、完了時刻	UOW	DFT_MON_UOW
SQL 文	開始時刻、停止時刻、文の識別	STATEMENT	DFT_MON_STMT

モニター・スイッチのオン/オフはDBM 構成パラメーターを下のようなコマンドを実行して変更するか、またはセッションごとに変更することができます。

```
update dbm cfg using dft_mon_stmt on
```

上の例ではインスタンス内の全てのデータベースにアクセスするアプリケーションの SQL 文情報の収集を行います。

```
update monitor switches using statement on
```

上の例は、スイッチを起動したアプリケーションの SQL 文情報のみの収集を行います。次の図はスナップショット・モニター・データの表示例です。この出力を表示するには、以下のようなコマンドを実行します。

```
get snapshot for database on sample
```

Database Snapshot

```
Database name                = SAMPLE
Database path                = D:¥DB2¥NODE0000¥SQL00001¥
Input database alias        = SAMPLE
Database status              = Active
Catalog node number         = 0
Catalog network node name    =
Operating system running at database server= NT
Location of the database     = ローカル
First database connect timestamp = 04/23/1999 17:40:16.629075
Last reset timestamp         =
Last backup timestamp        = 04/22/1999 13:07:18.815861
Snapshot timestamp           = 04/23/1999 18:44:45.407204

High water mark for connections = 2
Application connects          = 2
Secondary connects total     = 0
Applications connected currently = 2
Appls. executing in db manager currently = 0
Agents associated with applications = 2
Maximum agents associated with applications= 2
Maximum coordinating agents   = 2

Locks held currently          = 0
Lock waits                    = 収集されませんでした
Time database waited on locks (ms) = 収集されませんでした
Lock list memory in use (Bytes) = 864
```

一方、イベント・モニターの方はDBMSで、イベント・モニターをオンにして起動した以降、オフにするまでの間、イベントやトランザクションが発生すると、それらが起きた時点で記録するツールです。次のようなデータベース・オブジェクトに関するイベントを記録できます。

```
database
tables
deadlocks
tablespaces
bufferpools
connections
statements
transaction
```

イベント・モニターは初めて使う前に、データベースごとに、ユーザーが作成しなければなりません。次のようなコマンドで作成します。

```
create event monitor ev1 for deadlocks write to file 'eventmonitors/deadlock/evmon1'
```

イベント・モニターの開始と停止は次のようなコマンドで行います。

```
set event monitor ev1 state=1
set event monitor ev1 state=0
```

実はイベント・モニターの出力はバイナリーコードなので、そのままでは意味がわかりません。その内容を解析するためには、次のようなdb2evmon コマンドを実行して、バイナリーコードをテキスト・モードに変更する必要があります。

```
db2evmon -db db2cert -evm evmon1
```

次の図はそのイベント・モニターの出力例です。

```
event log header
-----

Event Monitor name:evmon1
Server Product ID:SQL0500
Version of event monitor data:3
Byte order: BIG ENDIAN
Number of nodes in db2 instance:1
Codepage of database:819
Country code of database:1
Server instance name: db2cert
-----

Database Name: DB2CERT
Database Path:¥db2¥db2cert¥node000¥sql00001¥
First connection timestamp: 08-14-1997 17:41:35.711003
REvent Monitor start time: 08-14-1997 20:40:18:955300
-----

6)Deadlock Event...
Number of applications deadlocked: 2
.....
```

スナップショット・モニターやイベント・モニターの開始/終了などは、コントロール・センター経由で簡単に行うことができますので、ぜひお試しください。この使い方さえ知っていれば、ある日突然応答時間が遅くなっても、何の理由かわからず、いらいらすることが少なくなることでしょ。たぶん。

[↑ 上に戻る](#)

[日本IBMについて](#) | [プライバシー](#) | [お問い合わせ](#)