



2009 : 75 リリースメモ

TNT 製品情報

2009 年 1 月 7 日

TNTmips ジョブ処理

TNTmips の DV2009 バージョンは、ユーザが地理空間解析の時間を効率的に利用できるように新しいジョブ処理 (JP) のコンセプトを導入しています。新しいジョブ処理によって、対話的な TNTmips のインターフェースを使って、大きなジョブや多くの似たジョブ、SML スクリプトを設定し、分散システムや他の環境の下で時間外に実行したり自動でコアを割り当てて使用することができます。

ジョブ処理の動作

TNTmips の DV2009 を立ち上げると、新しい TNT ジョブ処理機能も自動的に立ち上がります。[ツール (Tools)] メニューから [ジョブマネージャ (Job Manager)] を選択しなければ、表面上は違いはありません。

多くの TNTmips 処理には [実行] や [キャンセル] などの配置ボタンと同じ場所に、新しく [ジョブの保存 (Save Job)] ボタンが出来ます。タスクを実行する準備が整い、TNTmips 2008 では [実行] を選択するところでジョブの保存を選ぶことができます。これを選ぶと、指定されたジョブフォルダ (ディレクトリ) に XML ジョブファイルが作成されます。例えば、インポートするために 100 個のランドサットシーンを選んで、損失なしの JPEG2000 圧縮のラスターオブジェクトを作るとします。[実行] ボタンを使うと、TNT はすぐに 100 回の連続的なインポート (すなわち一度に 1 つのシーンずつ) の実行を開始します。[ジョブの保存] ボタンを選択すると、100 個の XML ジョブファイルがジョブフォルダに書き込まれます。〈ジョブマネージャ〉ウィンドウが自動的に開き、これらのジョブを複数のコアを使って並行的にすぐに実行するか、後で実行するために保存しておくかを選ぶことができます。各ジョブが完了すると、ジョブファイルは [ジョブ完了 (Jobs Done)] フォルダに移動します (ユーザが削除するまで残っています)。

また、SML スクリプトにユーザインターフェースがなかったり、[ジョブの保存] ボタンのインターフェースがない場合、SML スクリプトを実行すると XML ジョブファイルが作成されます。スクリプトから XML ジョブファイルが書き出され、その場ですぐに実行したりジョブ管理することができます。どんなジョブファイルも人が読むことができるので、TNT 処理やスクリプトによって作成された XML ジョブファイルをテンプレートとして使用したり、プログラムを使ってデータのファイル名やパス、URL などを変更することができます。例えば、あなたのプログラムやウェブ処理が、何か別の条件をモニタリングしていて、その結果を使ってジョブテンプレートを作り変え、完全な XML ジョブファイルをジョブフォルダに書き込み、実行することができます。ジョブファイルは TNTmips とは無関係にテキストエディタやユーザのプログラムを使用して作成することもできます。

コアが 1 個のコンピュータの場合

コアが 1 個のコンピュータを使用している場合でも、処理やスクリプトにおける [ジョブの保存] オプションを使って、いつ、どんな順番で TNTmips のタスクを自動実行するかを制御することができます。これにより、CPU 能力を要する TNTmips のタスクを夜間に行うようセットアップして、日中のコンピュータ資源を他のソフトウェアの使用に空けることができます。よくある用途として、100 個の画像のリサンプリングと投影変換を午後 6 時に開始するようにセットアップするなどです。コアが 1 個しか使用できないため、これらのジョブは順番に実行され、完了するのは翌日になるかもしれません。

コアが 2、4、8 個またはそれ以上のコンピュータの場合

2 個以上のコアを持った 1 台のコンピュータで使う場合、いくつのコアを使って保存したジョブを実行するかを指定することができます。デュアルコアのコンピュータがあるとして、日中のジョブ処理を 1 個のコアをバックグラウンドでのファイルのインポートに使い、もう 1 つのコアを他の仕事に残しておくこともできます。このモードでは、結果をできるだけ早くほしいが、その間他の用事があるといった場合に、1 つのコアをリサンプリングや投影変換のジョブを割り当てることができます。保存した同じジョブを夜間に行うのであれば、両方のコアを使用するように設定することもできます。

クアッドコア（コア4個）のデスクトップパソコン（あるいはノートパソコンでも構いませんが）が、現在入手可能です。インテルの新しいCore i7チップを使ったクアッドコアのCPUが、2009年のデスクトップコンピュータでは一般的になるでしょう。Core i7チップを2個使った8コアのCPUは、生産する単位としても経済的です。4個のコアに対してTNTmipsのジョブ処理を設定することができ、1個のコアで他のソフトウェアを使った対話的処理や、TNTmipsのジョブをセットアップし、残りの3個のコアをバックグラウンドで行う別のTNTmipsジョブ処理に割り当てることができます。4個全てを夜間のジョブ処理に設定することもできます。コアが8個の場合、日中は対話処理に2個、ジョブ処理に6個割り当て、夜間は8個全てを使って処理を行うことも考えられます。以上がマイクロイメージでテストおよび実際に使用している方法の典型例です。

SML スクリプトの使用

TNTmipsのメニューでの処理と同様に、SMLスクリプトを使ってTNTmipsのジョブ処理をセットアップすることができます。スクリプトはいつものようにユーザと対話しながら入力を受け付け、ユーザの目的に合ったジョブをセットアップするよう設計されています。TNTmipsの処理の中の「ジョブの保存」ボタンによって作成されたように、XMLジョブファイルが保存され、実行されます。

分散システムに関して

TNTmips処理やスクリプトによるジョブセットアップは、1つまたは同じTNTmipsのコンピュータ上で実行される必要はありません。コンピュータが複数台あって、複数のTNTmipsのキーや複数ユーザで使えるフローティングライセンスがある場合、数日にわたる生産作業を全部のコンピュータを使って夜間に走らせることもできます。

GIS 処理提供サービス

マイクロイメージ社のサンプルのGIS処理提供サイト geospatialgateway.com では、ここで紹介したジョブ処理やSMLスクリプトを使っています。エンドユーザは異なる目的のウェブアプリケーションから選択することができます。サービスはそれぞれ異なるインターフェースを持ち、SMLスクリプトに対する必要な入力を受け付けます。入力情報はユーザのウェブサーバ上のPHPスクリプトに送られ、XMLジョブファイルに構造化され、TNTジョブフォルダに書き込まれます。PHPの代わりに、ASPやPerl、Pythonなどユーザが得意なサーバサイドのスクリプト言語を使用することができます。

各ジョブファイルには、SMLスクリプトと同じ内容が含まれています。TNTmipsのジョブファイルのディレクトリはLAN上どこでも置くことができます。1つまたは複数のTNTmipsシステムが同じネットワークコンピュータ上でたえず動いており、ジョブフォルダを監視してジョブファイルがあるかチェックしています。処理に必要な地理データは、ネットワークドライブやファイルサーバ、TNTmipsを1つだけ使用している場合はローカルドライブからアクセスされます。各TNTmipsはこの共通の待ち行列から、①上で説明したような使用するコア数や、②スクリプト中にセットされたジョブタイプ毎の処理の優先順位および③各TNTmipsステーションごとのジョブマネージャの設定に従って、1つまたは複数のジョブを引き出します。TNTmipsがジョブを完了すると、スクリプトが結果のファイルをFTPサイトに書き込み、ユーザにGIS処理が完了したことを知らせ、ファイルダウンロードのリンクを教える電子メールを送信します。他にも様々な方法を使ってユーザや他のプログラムにジョブが完了、使用可能になったことが通知できます。

TNTmipsをベースとしたGISデータ処理提供サービスの操作方法について、詳しくは

www.geospatialgateway.com/geoprovisioningWorks.htm

で説明しており、www.geospatialgateway.com で試すことができます。

非同期マルチコア処理とは

TNTmipsでは、非同期マルチコア処理（AMP）を使って、保存したジョブをバックグラウンドまたはユーザが指定したスケジュールで実行することができます。AMPの考え方は単純です。ユーザは今までも複数の似たようなあるいは違うTNT処理を手動で開始して、1つまたは複数のコアを使って同時に実行してきました。AMPはこれと同じ考え方を保存した全てのジョブの処理に自動的に適用します。TNTの処理をバックグラウンドで処理するのに3個のコアが使えるとしましょう。TNTのジョブ処理は、同一または異なる処理の3つの作業をそれぞれが1個のコアを使って開始します（マルチコア処理）。これらのジョブのどれかが完了してコアが空くと、新しいジョブがそのコアにあてがわれます（非同期タスク処理）。

AMP とマルチスレッド処理の関係

最近の文献は、最新のコンパイラや言語、デバッグツールを使ってマルチスレッドのアプリケーションを作るのがどれだけ大変かを議論する記事であふれています。マルチスレッドのメリットを最大限に利用するためのプログラムの変換作業は、ツールの改良や交替も相俟って、ゆっくりと慎重に進められています。マルチスレッド化は非常にゆっくり

やって来ます。しかし、マルチスレッド化は TNTmips のような製品において以下の点で有効です：

- 1) パンニングの際のタイル単位の表示（バッファリング）のような改良されたユーザインターフェース操作。
- 2) ローカルのレイヤを操作しながらインターネットから画像レイヤをロードするといった、バックグラウンドでネットワークを使用処理。
- 3) 多数の画像の投影変換やエクスポートなどの際の作業の生産性の向上。

TNTmips のバージョン 2008 では 1) と 2) の問題について改善しています。TNTmips DV2009 におけるジョブ処理と AMP は多くの処理において、マルチスレッド化すべきバッチ志向処理を書き換えることなく、3) の問題を解決しようとしています。

AMP は、マルチスレッド用に修正された TNTmips の処理を使うことができます。AMP はただ単に TNT ジョブ処理を組み合わせ、自動的にマルチタスク処理を遂行します。コンピュータの全てのコアを最大限使用することができるならば、処理が AMP で行われたかあるいはマルチスレッドだけか、または両方の組み合わせによって遂行されたかはほとんど区別できません。AMP を使った TNT のジョブ処理は、複雑なマルチスレッド化のコーディングを回避し、全てのコアを最大限使用することができます。

JPEG 2000 圧縮や解凍に使用される Kakadu ライブラリはマルチスレッド化されています。大きな JP2 ファイルにリンクして新しい JP2 ファイルをエクスポートし、その際圧縮比を変えることができます（例えば損失なし圧縮から 1/15 圧縮）。この処理は画像をいくつかのセグメントに解凍し、マルチコアを使って再圧縮します。この間 CPU は連結して動いており、OS のタスクマネージャを使えばコアの使用状況を見ることができます。このようなシングルタスクに対して TNTmips がコアを 4 つ使用する場合、4 個の CPU または 4 個のコアが約 25% のパワーでマルチスレッドを使いながら短い読み書きのための中断を入れながら動くのを観測することができます。TNTmips 2008 を使うと、4 つ以上の大きなファイルのエクスポート処理を手動で順番に開始した場合、4 個全てのコアの CPU 時間が 100% に達するのが分かります。TNTmips の DV2009 では、AMP を使って簡単に CPU の使用率を自動化することができます。同じ 4 つ以上のファイルを選択し、[ジョブの保存] ボタンを使ってそれら全てを 1 つのジョブとしてエクスポートします。ジョブ処理での AMP を使った方法は、4 つ全ての CPU をほぼ 100% レベルで使用します。この場合、結果が同じなのでマルチスレッドが働いてコアの使用が最大化しているのか、AMP が働いて最大化しているのか分かりませんし、気にする必要もありません。

ジョブマネージャ処理

たくさんの異なる種類のジョブやスクリプトをジョブフォルダを介して処理待ちの行列に入れることができます。それらは TNTmips の処理や SML スクリプト、手動で準備した XML ファイル、HTML 形式のウェブページ、ウェブアプリケーション、ユーザのプログラム、GIS 処理提供サイトなどから来ます。DV2009 のジョブマネージャプロセッサは TNT の [ツール (Tools)] メニューから開くことができ、また処理で [ジョブの保存] ボタンが使われた場合にも自動的に開きます。プロセッサは、各 XML ジョブファイルの処理がいつ、どのように行われるかを管理・モニターするために必要なインターフェースを提供します。このウィンドウで、ジョブを実行するときに TNTmips が使用するコア数を設定します。また、トグルボタンがあって、新しいジョブが待ち行列に追加されたときにすぐに実行するか、あるいは実行しないでそれらのステータスや優先度を設定することができます。以下で説明するタブパネルのように既に実装された機能の他にも、より多くの機能が計画されています。

[ジョブリスト (Job List)] タブパネル

ジョブマネージャのインターフェースでは、[Queue (待ち行列)] や [Done (完了)]、[Failed (失敗)]、[Settings (設定)] タブパネルを提供します。[待ち行列] パネルには、処理 ID や名前など各ジョブに関するメタデータの他に、現在待ち行列の中にある各ジョブの状況や優先度、ID、名前、作成日時を一覧表示します。各ジョブのジョブステータスには実行や実行中、停止、削除などがあります。ジョブのプライオリティは 1、2、3... という番号で示されます。ジョブやジョブグループのステータスと優先順位は、このパネルで設定したり変更することができます。[完了] パネルには、終了したジョブに関するメタデータがジョブが完了した順に表示されます。ジョブがこのパネルに一覧表示されると、XML ジョブファイルは [ジョブ完了] フォルダに移動します。[失敗] パネルには、正常に実行されず [ジョブ] フォルダに残り、スキップされたジョブに関するメタデータが含まれます。

[設定 (Settings)] タブパネル

このパネルでは、[ジョブ] フォルダや [ジョブ完了] フォルダを置く場所のパスを設定することができます。場所はユーザのネットワーク上のどこにでも設定することができます。いくつかの TNTmips システムで同じファイルや場所に設定することができます。このパネルから、全てのジョブの実行に対しての遅延時間や他のコントロールパラメータを設定することができます。

ジョブ処理が使用可能な処理

以下は、[ジョブの保存] ボタンが使えるように修正された TNTmips の処理です。処理に対話処理がある場合、[ジョブの保存] ボタンはありません。例えば、領域の定義にポリゴンを使う場合、[ジョブの保存] ボタンは現れず、タスクはその場ですぐ実行されます。領域を描いたり、地物を選択するなどといった対話処理を必要とする作業は、通常1度に1つずつ実行されます。現在、ジョブマネージャの改良に焦点を当ててコーディング作業が行われています。しかし、下に記載していない TNT 処理で [ジョブの保存] ボタンの追加要望があれば、実装が可能かどうかマイクロイメージに確認して下さい。

ジョブ処理が使えるようにすでに修正された処理

SML

地形特性 (Topographic Properties)
ラスタのリサンプル (Raster Resample)
ベクタの投影 (Vector Reproject)
ピラミッド (Pyramid)
無相関 (Decorrelate)
色の変換 (Convert Color)
主成分 (Principle Components)

ジョブ処理が使えるよう修正中の処理

バッファゾーンの作成 (Buffer Zones)
抜き出し (Extract)
自動分類 (Autoclassify)

ジョブ処理が使えるインポート / エクスポート形式

ArcBIL/BIP (.bil, .bip)
Arc-Grid (.adf)
Arc-Shapefile (.shp)
Arc-Text
ASTER-HDF (.hdf, .dat, .l1r, .l1g)
BMP (.bmp)
CADRG
CCRS (.tm)
CGM (.cgm)
DCR (.dcr)
DEM (.dem)
DEM-GTOPO30 (.dem)
DGN (.dgn)
DIMAP
DOQ (.doq)
DTE (.dte)
DWG (.dwg)
DXF (.dxf)
ECW (.ecw)
ER-Mapper (.ers)
ERDAS-IMG (.img)
Geosoft-XYZ (.xyz)
GIF (.gif)
GIRAS
Google-Local (.jpg, .jpeg, .png)
GRASS-Raster
HDF4-Generic (.hdf, .dat, .l1r, .l1g)
IRS
JP2, GeoJP2 (.jp2)
JPEG (.jpg, .jpeg)

Landsat-NLAPS (.hd, .hi, .dh, .h1)
Landsat7-HDF (.hdf, .dat, .l1r, l1g)
LAS, LAS-DB (.las)
MODIS-HDF (.hdf, .dat, .l1r, .l1g)
MrSID (.sid)
NetCDF (.nc, .cdf)
NetCDF_GMT (.grd, .gmt, .nc, .cdf)
NITF, NITF2.1 (.ntf)
NTAD (.geo)
NTF-vect (.ntf)
PCI (.pix)
PCX (.pcx)
PNG (.png)
PRISM
RADARSAT
SDF (.arg, .pnt, .pgn)
SDTS-DEM (.dbf)
SRTM (.hgt)
SunRast (.ras)
Surfer (.grd)
TIFF, GeoTIFF, BigTIFF (.tif, .tiff)
TIGER (.rt*)
TM-Fast-L7A

ジョブ処理をサポートしていないインポート / エクスポート形式

Access (.mdb)
Access (.mdb)
ADRG (.thf)
Arc-Coverage
Arc-E00-Grid
Arc-E00-Vector (.e00)
Arc-Generate
AVIRIS (.img, .rfi)
CARIS (.txt)
DB-Text (.txt)
DTED (.dto, .dt1, .dt2)
ENVI (.hdr)
ERDAS-GIS/LAN (.lan, .gis)
ERS-SAR
Excel (.xls)
Geosoft-GRD (.grd)
Geosoft-GXF (.gxf)
GML (.gml)
IDRISI (.rdc, .img)
ILWISR (.mpr)
JERS1
KML (.kml, .kmz)
Landsat-CCRS
MapInfo, MapInfo-Attrib (.tab)
MIF (.mif)
MOSS (.mos)
My_SQL
MySQL_Spatial
ODBC (.dbf, .ndx, .mdx, .xls, .mdb, .dbc)
Oracle
PostGIS

PostgreSQL
Simple-Array
SPOT (.dat)
SPOTView (.hdr)
SQLServer
Text-Raster (.txt)
Text-XYZ (.txt, .csv, .xyz)
TM-Fast
User-Defined