



OpenTect中的并行计算

第2章的2.1.1节

第6章



2.1.1 分布式计算和并行化

使用多线程的数据处理

Dip-Steering Algorithms	
Compute a Steering Cube with BG Steering	Completely re-designed in Nov. 2015
Compute a Steering Cube with FFT	Completely re-designed in Nov. 2015
Apply Full Steering dip-steering in attributes and filters	Dip-Steering 断层属性计算
Apply Central Steering in attributes and filters	

Attributes 地震属性	
Convolve (all except Wavelet option)	
Curvature	
Dip Angle 曲率和倾角	



Velocity Fan Filter (=DipFilter)	
Energy (all except Gradient option)	
Event	
Frequency Filter	multi- threading implemented in March 2014
Frequency	multi- threading implemented in March 2014
HorizonCube Data	
HorizonCube Density	层位
HorizonCube Layer	
Hilbert	
Instantaneous	multi-threading improved in March 2014
Local Fluid Contact Finder	
Maths (except when expression is recursive)	
Polar Dip	
Position	
PreStack	
Reference	
Scaling (all except scaling type AGC and stats type = detrend)	
Semblance	相干体计算
Similarity	
Spectral Decomposition	multi- threading implemented in March 2014
Texture	
Tutorial	
Volume Statistics	

求真务实

求真务实



尚未使用多线程的数据处理

FaultDip	断层倾角
FingerPrint	
GapDecon	
HorizonCube Curvature	
HorizonCube Dip	
HorizonCube Spacing	层位和层位属性
Horizon	计算
Match Delta	
Maths (recursive expression)	
Perpendicular Dip Extractor	
SampleValue	
Scaling (scaling type AGC and stats type = detrend)	
Shift	

Attributes
Constant Steering
Convolve (Wavelet option)
DeltaResample
Energy (Gradient option)
Event Steering

注：由此可见，很多地震属性计算，主要是断层与层位解释，需要并行计算，提高解释效率，针对大规模地震数据的解释。



6 处理(Processing)

6.6批处理

教学视频

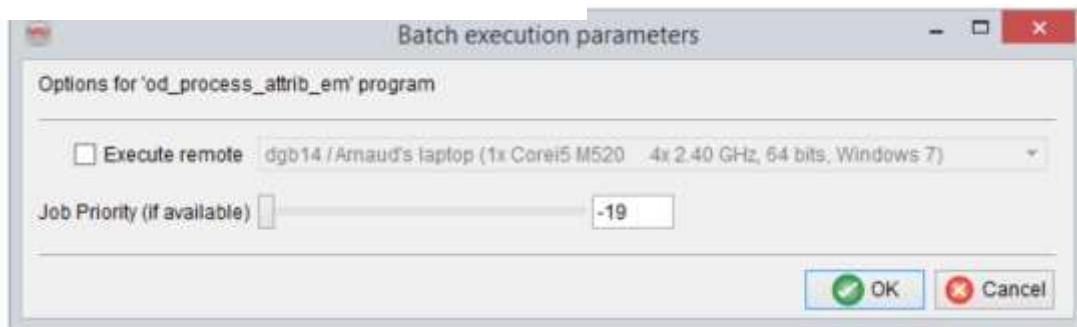
<https://www.youtube.com/watch?v=9bEXxZ0RyxU>

Though *Batch Processing* does not actually appear in the pop-out menu under *Processing*, this is a convenient place in this User Documentation to present the information on these processes.

Batch Processing实际上不出现在Processing的弹出菜单。

6.6.1 Single Machine Batch Processing Window

单机上的批处理窗口



单机模式处理，数据在当地机器或远程主机上处理。将存储计算进程的所有相关信息（见Job信息文件）。

Execute remote（作业优先度：-19最低，19最高）



6.6.2 分布式计算窗口

The distributed computing window controls on which machines a volume output or SteeringCube batch job will be processed. Jobs are distributed over the *Used hosts on a line-by-line basis* (the *number of inlines per job can be specified*). Hosts can be *Added and/or Stopped* at all times. Processed results are stored in a *Temporary storage directory*.

At the end of the processing sequence, OpendText will merge all processed lines and store the data in the output file that was specified in the *Volume output or Create SteeringCube* window, and it will delete the temporary disk files. If for any reason OpendText fails to perform this data transfer, this can also be done manually in the *File - Manage* module. The temporary data store appears with a name starting with *Proc_*. Select this item and *copy it to a new cube*.



6.6.2 分布式计算窗口

It is possible that at the end of a distributed computing batch job not all data was processed successfully. Some jobs may have failed (e.g. because one of the hosts was temporarily not available). OpendText will come back with a warning message stating which jobs (i.e. which inlines) have not been processed yet. It is then advised to re-submit these jobs until all data are processed. The *Auto-fill* option automatically scans and fills gaps in the processed volume.

The *Nice* level sets the priority the process gets. With the nice level set on 19 for exemple the process has very low priority and other processes that run on the same computer get priority. If the nice level is set to 1 the process gets the highest priority.

The *Processes* menu allows to set the *Run, Pause, or Go - Only between* options. The *Go - Only between* option, pauses and runs the processes at user-defined times.

OpendText calls the system utilities of the *'hostent'* (sethostent, gethostent, etc.) type to get a table of hosts that can be selected. How the Operating System builds the lists is dependent on the particular system setup; most likely */etc/hosts* and/or the *NIS* tables are consulted. OpendText supports *multi-threading* which means that all processors of multi-processor machines will be used.



并不是所有计算都支持多线程，例如涉及递归计算的算法。见下表：

Multi-threaded	Not multi-threaded
BG Steering*	Constant Steering
Central Steering	Convolve (Wavelet option)
Convolve (all except Wavelet option)	DeltaResample
Curvature	Energy (Gradient option)
Dip Angle	Event Steering
Velocity Fan Filter (=DipFilter)	Fault Dip
Energy (all except Gradient option)	FingerPrint
Event	GapDecon
FreqFilter*	HorizonCube Curvature
Frequency*	HorizonCube Dip
Full Steering	HorizonCube Spacing
HorizonCube Data	Horizon
HorizonCube Density	Match Delta
HorizonCube Layer	Maths (recursive expression)
Hilbert	Perpendicular Dip Extractor
Instantaneous**	SampleValue
Local Fluid Contact Finder	Scaling (scaling type AGC and stats type = detrend)
Maths (except when expression is recursive)	Shift
Polar Dip	
Position	
PreStack	
Reference	
Scaling (all except scaling type AGC and stats type = detrend)	
Semblance	
Similarity	
Spectral Decomposition*	
SpectrogramDip (create SteeringCube, FFT steering)	
Texture	
Tutorial	
Volume Statistics	



Distributed Computing on Windows OS

The new system works with a **Daemon Service** running in background on every remote machine to be used for processing. The communication works **with TCP/IP** and requires some configurations to actually make things working.

Mapping of Survey folder: We assume a DataRoot folder (\$DTECT_DATA) to be in a centralised server. This folder has to be mapped on the same drive in every PC to be used as hosts. For example, we have 4 PCs (A,B,C,D). A is a server and the ODDData is in drive D. Then map D:ODDData on e.g. S: on B, C and D. PCs B, C, and D can then be used as your processing nodes.

OpendTect installation: You need to have OpendTect installed on all hosts, and make sure they all use the same survey at the same time (have access to). For example if B is using F3_Demo and want to process something in F3_Demo. then it has to be made sure that the rest of the two PCs also use the same folder as long as the processing is needed.



Distributed Computing on Windows OS

BatchHosts file: Add the IP addresses of C and D in the BatchHosts file inside the application data folder.

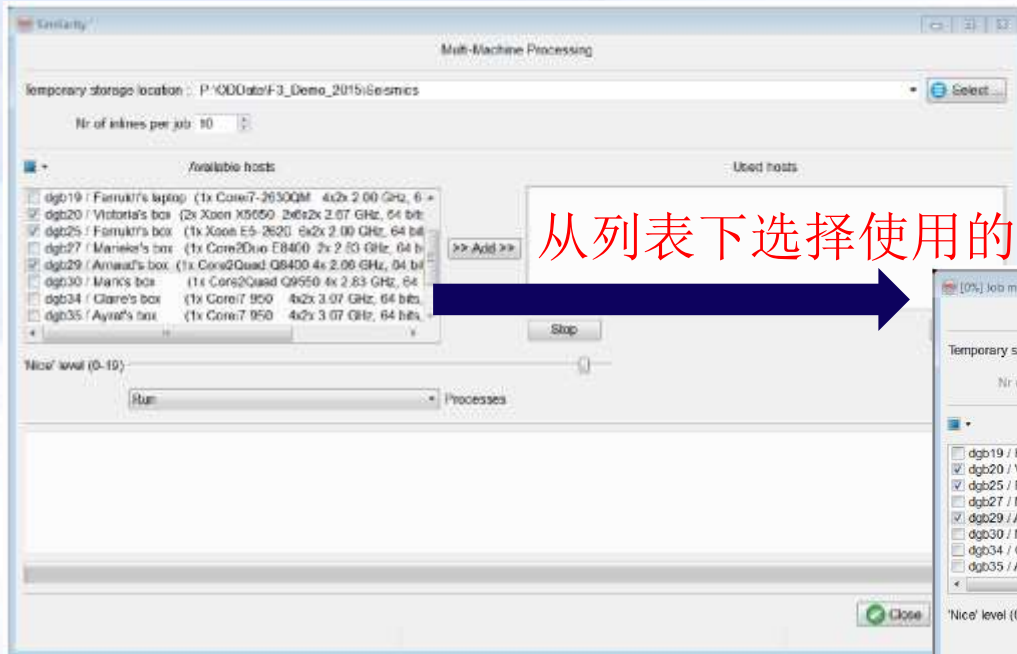
Start the daemon: If launching process from B to the other two, then B is his local machine, C and D are remote machine. In this case the Daemon service (odremoteservice) application has to be launched from bin\win folder (win64) only in the remote machines and not in the local machine (B). Please note the odremoteservice.exe not to be run directly instead a launching tool will be found in win32mmod_remote_service_manager. Use od_remote_service_manager to launch the daemon which will also add a notification icon to the system tray. Once the service starts, the remote machines are ready.

Start processing: Select the PC's B, C and D from the list of machines in the distributed computing launch window and start processing:

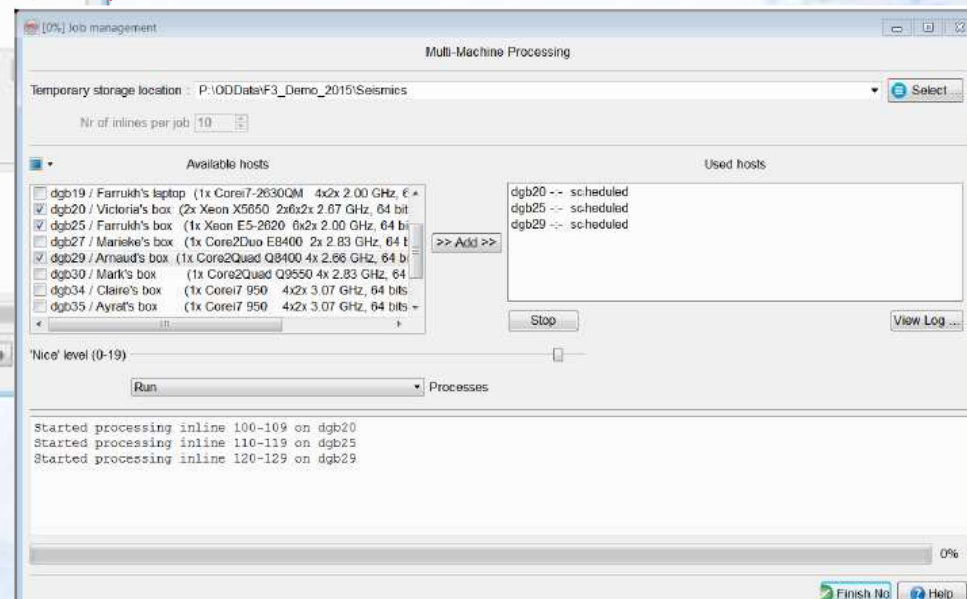


Distributed Computing on Windows OS

LINUX系统上的操作也类似



从列表下选择使用的机器



Distributed Computing progress window

For more information, please refer to the **OpendTect YouTube Channel** for the webinar on: **Distributed Computing Set-Up**



6.6.3批处理日志文件

程序每处理N个震道，输出1个符号，有5种符号，表示：

- a period (.) means 1 trace processed
- a colon (:) means 10 traces processed
- an equal sign (=) means 100 traces processed
- a pipe marker (|) means 1000 traces processed
- an asterisk (*) means 10000 traces processed

```
Progress
Starting program: od_process_attrib P:\ODData\F3_Demo_2015\Proc\Similarity.par
Processing on: dgb16
Process ID: 7524
Preparing processing of 'Similarity'.
Estimated number of positions to be processed (assuming regular input): 94900
.....:.....: 0% (21/s) (1h:15m:17s)
.....=:.....=: 1% (608.273/s) (2m:35s)
=====|=====|=====|=====|=====| 6% (622.123/s) (2m:24s)
=====|=====|=====|=====|=====| 11% (551.633/s) (2m:33s)
=====|=====|=====|=====|=====| 16% (595.806/s) (2m:13s)

===
```

有如下选项：



Stops the process.



Saves the log file.



Brings you to the help menu.



6.6.4 集群处理

批处理作业可以从OpendTect发送到集群管理工具下运行。目前已成功测试了SLURM。

启动集群处理需要以下前提条件：

- (1) 设置环境变量DTECT_CLUSTER_PROC=yes
- (2) 集群管理工具必须运行安装，即主机上运行slurmctld，计算节点上运行了mlurmd
- (3) PATH变量中包括了SLURM的bin路径

在数据体处理窗口的Submit选项菜单下，选择Cluster，启动集群处理。目前仅3D属性使用集群计算。

会弹出新的窗口列出使用临时存储文件的一系列路径。使用用户自定义的inlines数分解作业。文件命名为'Cluster Processing command'，代表从集群管理工具运行进程的二进制名称。

你可以命令行运行"Main script file" (默认是~/your survery/Proc/clusterprocscrip)，将使用以上命令依次运行各个作业，也可以启动UI显示进度，执行合并临时数据的后处理。