EAGE

SSP07

Assessing Full Azimuth RTM

M. Hanzich* (Barcelona Supercomputing Center), M. Araya-Polo (Barcelona Supercomputing Center), J.M. Cela (Barcelona Supercomputing Center) & F. Ortigosa (Repsol)

SUMMARY

The combination of rich azimuth seismic data and RTM enhance the seismic imaging quality. However, many problems arose that have to be tackled when using RTM with rich azimuth data. The aim of this work is to show the problems and propose some solutions.

The following image depicts the illumination results for a regular narrow/single azimuth acquisition. Clearly, under the salt dome we have no illumination information, thus the delineation of the dome base is not reliable.

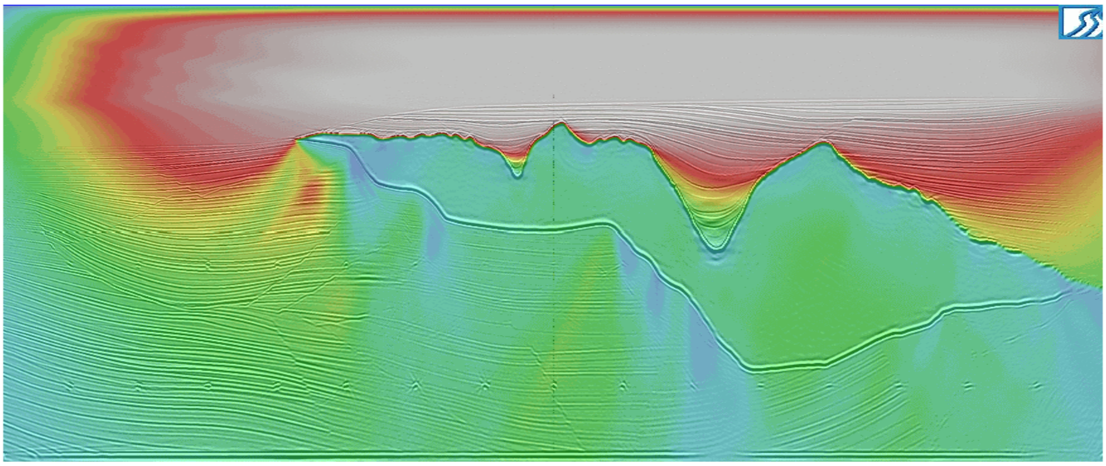

Illustration 1: Courtesy ScreenImaging.

One way to improve these results is to extend the acquisition to rich azimuth, thus increasing the coverage of the target subsalt area.

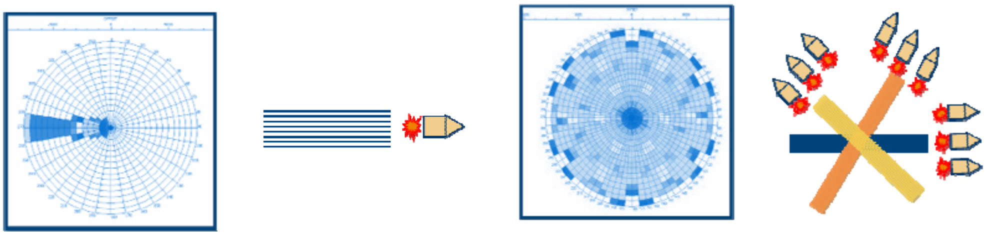

Illustration 2: Left: single azimuth, right: rich azimuth. Kapoor et al. SEG 2006.

The combination of rich azimuth plus Reverse Time Migration (RTM) is what we called high-definition seismic imaging. This is because the quality of the images produced by RTM is the best that the industry has so far, and the increased coverage helps to improve the quality of the structure definition. This combination not only requires a change in the acquisition, but also in the RTM. The main RTM change is the process of the extra information that is incorporated to the migration.

Current RTM implementations for single azimuth require a large amount of computational resources. A extended RTM implementation will be even more demanding on those resources. The main RTM constraints are: memory, I/O and computational.

Memory: RTM memory consumption is mainly related to the frequency at which the migration should be done. High frequencies imply the usage of several GB of memory for single shot migration.
Even considering that the current HPC platforms normally have a lot of memory per node, it may be not enough to migrate one shot for a given frequency. This force RTM to run across many nodes to gather enough memory, then increasing the execution time due to the communications overhead.
Moreover, accessing several GB of memory data in a short period of time overwhelms the memory hierarchy. Therefore, severely punishing the computation time.

Finally, some modern HPC environments (e.g. Cell/B.E or SGI Altix) have a memory organization that need a different access time depending on the data layout. That organization should be considered by the implementation, in order to avoid memory contention, unbalancing and enhance scalability at core level.

<u>Input/Output</u>: several problems related to I/O affects RTM efficiency: data size, storage limitations and concurrency.

- As mentioned in memory section we will handle huge computational domains, moreover the time discretization and image conditions may turn in a large data volume to be stored in disk. The two main constraints of the storage systems are: storage capacity and transfer speed. RTM implementations store the whole computational domain regarding the number of timesteps, which may overwhelm the storage capacity.
- At the same time, the required time to store a computational domain must be equivalent to at least the bandwidth of the system. Otherwise, the computational performance is negatively affected.
- Distributed systems allow several nodes access the same file concurrently. This feature adds new constraints to be taken into account: storage network bandwidth and the maximum number of concurrent petitions that can be served.

<u>Computing</u>: the RTM main computational constraint is the stencil computation. Its memory access pattern is a main concern when designing the RTM kernel code. The pattern strongly depends on the memory hierarchy structure of the target architecture. Among the details we need to consider for each architecture we have: the number of cache levels and the size and access cost for each one of them.

Within Kaleidoscope (www.kaleidoscopeproject.info) project we have developed a high-performance RTM based on Cell/B.E. capable of processing rich azimuth campaigns. Our RTM solves all the mentioned problems delivering a high quality product.

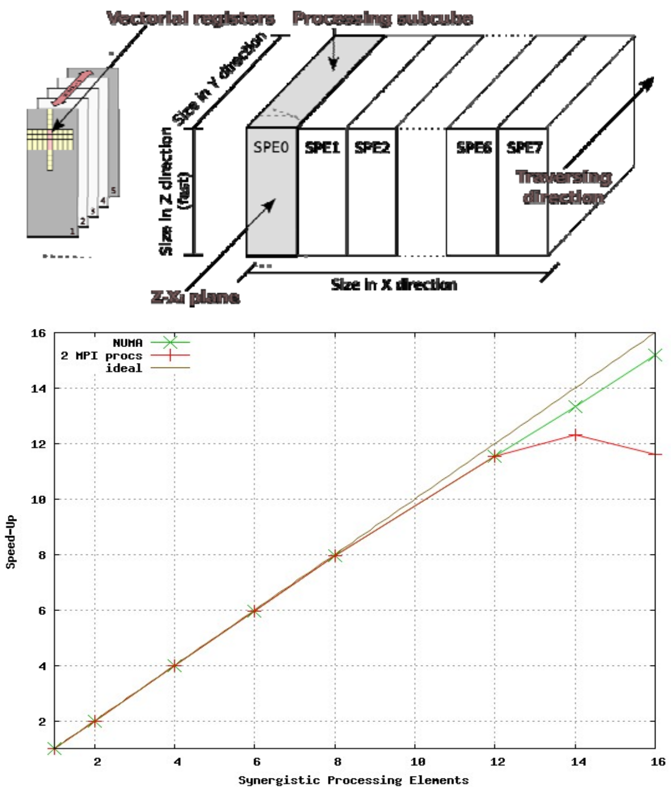

Illustration 3: The figure on top correspond to an optimal data organization for the Cell/B.E. architecture. This organization is optimal because distribute data and computation evenly among accelerators. The bottom figure depicts the scalability of our RTM in terms of the number of accelerators, as we can see the most efficient variant of our implementation reaches up to 15.3 of 16 scalability, which means that almost all the accelerators work at full performance.