A Method for OSEM PET Reconstruction on Parallel Architectures using STIR

Tobias Beisel 1
Stefan Lietsch 2
Kris Thielemans 3

1 tobise@uni-paderborn.de
2 slietsch@uni-paderborn.de
3 Kris.Thielemans@ge.com

STIR: Software for Tomographic Image Reconstruction

Simple Distribution Scheme

1. broadcast image estimate
2. for all views in the subset do
3. distribute viewgrams
4. calculate multiplicative factor on each slave
5. use calculated factor on local image estimate
6. end for
7. reduce reconstructed images at master node

Enhanced Distribution Scheme

1. mark all views as processed
2. for view, count do
3. wait for slave node request
4. if not empty(list of unprocessed view to slave)
5. send list of future processed view to slave
6. mark view as processed
7. else
8. if first iteration then
9. get any unprocessed view
10. else
11. get unprocessed view of latest slave
12. end if
13. send view to be processed on slave
14. mark view as processed
15. add view to list of processed views of slave
16. end if
17. end for

OSEM algorithm:

\[ f_{n+1} = f_n + \frac{1}{2} \left( W f_n \right) \]

\[ f_{n+1} = f_n \]

\[ f_{n+1} = f_n \]

- Workers need to be initialized with
  - Projectors
  - Image estimate
- Workers receive work packages
  - Each processes 4 views

Results

- Used ECAT EXACT HR+ data in 3D mode
  - 144 views, 7 segments, 144/num_subsets work packages
  - 63 axial positions, 288 tangential positions
  - Image dimensions (265, 265, 63), 16 MB per image
  - 36 sub-iterations
- Considerable speedups attained
  - Speedup of up to 3.2 on the Intel Xeon Clovertown
  - Speedup of up to 8 on the cluster system
  - Higher speedups for modern 3D data expected
- Substantial runtime reduction
  - 9 to 3.5 min (2D data, Intel Xeon Clovertown)
  - 29 to 5 min (2D data, cluster system)
  - 29 to 9 min (3D data, Intel Xeon Clovertown)
  - 95 to 12.5 min (3D data, cluster system)

Contributions and Achievements

- Identified shared memory systems as best in cost/performance ratio
- Encapsulation of parallel functionality for future use
- Portability of developed algorithms
- Contributions available as Open Source

Future Work

- Extend parallel STIR implementation
- Sensitivity image calculation, list-mode data
- Port to other algorithms (e.g. OSSPS) and modalities (CT/MRI)
- Port to GPUs, Clearspeed, Cell Broadband Engine

Example parallel architectures

Intel Xeon Clovertown System
- Shared memory (8 GB, FSB connected)
- Short communication distances
- Restricted number of nodes (8)

Xeon DP based ‘ARMINIUS’ cluster
- Distributed memory (900 GB, InfiniBand connected)
- Long communication distances
- Extendable number of nodes (400 available)

Contributions and Achievements

- Identified shared memory systems as best in cost/performance ratio
- Encapsulation of parallel functionality for future use
- Portability of developed algorithms
- Contributions available as Open Source

Future Work

- Extend parallel STIR implementation
- Sensitivity image calculation, list-mode data
- Port to other algorithms (e.g. OSSPS) and modalities (CT/MRI)
- Port to GPUs, Clearspeed, Cell Broadband Engine