Using the Vector Signal and Image Processing Library with OpenMP and MPI

- Exploiting VSIPL and OpenMP for Parallel Image Processing, Jeremy Kepner MIT Lincoln Laboratory

- Interoperability of VSIPL with MPI, Purushotham Bangalore Mississippi State University
**VSIPL**

- C API for processing vector, signal or image data.
- Optimized for use with single processor machines.
- Data storage is abstracted, data is accessible only through VSIPL functions.
Interpreted Data Language

- Software package used for scientific data analysis.
- Uses VSIPL.
- Implements an image convolution algorithm.
Image Convolution

- For every pixel in an image, do something based on the value of neighbor pixels and add the new pixel to the result.
- The value of the new pixel is defined by Point Response Functions (one or more).
- For each PRF, every pixel must be evaluated.
Parallelizing Image Convolution with OpenMP in IDL.

- Kepner modified IDL to use OpenMP.
- Convolve each PRF on a different processor and combine the results on a single processor.
- Works well with VSIPL and OpenMP, you do not need direct access to VSIPL data.
- Simple and effective solution.
SGI Origin 2000

- 64 300MHz MIPS R10000 processors.
- 4 processors/node 1GB/node (16 GB total).
- Non-uniform memory architecture.
- Memory physically distributed over nodes with one system image.
- Effectively a SMP machine.
Results
Interoperability of VSIPL and MPI

- VSIPL data can only be accessed through VSIPL functions.
- Data must be accessible directly to do a MPI send/receive.
- Bangalore gives three possible solutions.
Solution #1

- Copy data from abstract data types to public data arrays using VSIPPL functions.
- Make MPI send and receive calls.
- Copy data into abstract data type using VSIPPL functions.
Solution #1

• Advantages:
  – Simple
  – Easy to implement

• Disadvantages:
  – Overhead
  – Extra copies of data
Solution #2

- Use admit and release functionality provided by VSIPL.
- Admit function – locks data into VSIPL ownership, makes data accessible only to VSIPL API.
- Release function – unlocks data into user ownership, makes data accessible only to user activities.
Solution #2

- Best solution.
- Admit and release functionality was added to VSIPL after David Schwartz suggested it (originally as import and export functions).
Solution #3

• Create a new API.
  – New data types
  – Abstract VSIPL and MPI calls
Solution #3

- Not a quick solution.
- Feasible for long term uses.
Any Questions?