Zoltan: Load Balancing, Partitioning and Ordering

Zoltan provides parallel combinatorial algorithms important to scientific computing.

- **Dynamic load balancing** maintains balance and locality in adaptive or evolving applications.
- **Data ordering**

Zoltan supports a variety of applications with differing data structures and computational requirements.

- **Geometric methods** use physical coordinates to maintain locality of data.
- **Topology-based methods** use data dependencies to reduce communication and data movement.

Using Zoltan in new or existing applications is easy.

- Native Zoltan interface uses callback functions for applications to describe their data.
- The iZoltan service accesses Zoltan using ITAPS-compliant mesh interfaces (iMesh, iMeshP).
- Common interface to all methods allows experimentation within applications to find the most effective strategy.

ParMA: Partition improvement using Mesh Adjacencies

Mesh adjacencies represent application data more completely than standard graph-partitioning models.

- **All mesh entities can be considered, while graph-partitioning models use only a subset of mesh adjacency information.**
- **Any adjacency can be obtained in O(1) time (assuming use of a complete mesh adjacency structure).**
- **Partition model supports efficient partition adjacency queries.** (See FASTMath’s Unstructured Mesh poster for more details.)

**Partition Improvement with Multiple Criteria**

- Diffusive procedure driven by application-defined priority list of mesh entity types to balance. Iteration over three stages:
  - **Schedule** load transfer from heavy parts to light parts.
  - **Select** mesh elements on the part boundaries that will smooth the partition boundary
  - **Migrate** selected mesh elements according to schedule

With ParMA Vtx>Rgn, PHASTA CFD strong scaling improves from 0.88 to 0.95 on 288k cores, JuGene BG/P, with a 18-region mesh.

Hierarchical Partitioning for Multicore Computers

- Zoltan’s hierarchical partitioning distributes data across levels of computers’ hardware hierarchy (nodes, sockets, dies, cores).
- **Improve data locality within hardware components**
- **Experiments on Magny-Cours (NERSC’s Hopper) showed greatest benefit from partitioning first with respect to nodes and then with respect to cores (Hier.4.24).**

Predictive Load Balancing Research

- **Parallel unstructured mesh adaptation can generate parts with up to 400% imbalance on non-trivial geometries due to local coarsening & refinement.**
- **Refining** before repartitioning can exceed available memory in some processes, even if the system’s total memory is sufficient.
- **Solution:** Redistribute mesh before adapting.
  - **Merge parts that will be coarsened to create some empty parts.**
  - **Split parts with heavy refinement into the empty parts to remove imbalance spikes.**
  - **Refine/coarsen the mesh.**
  - **Apply ParMA’s diffusive partition improvement.**

Current Research and Development

- **Integration of Zoltan and ParMA in iZoltan**
- **MPI+threads implementation of partitioner**
- **Communication reduction via asynchronous unstructured mesh migration procedures**
- **Scalable, effective partitioning for high core counts**
- **Hypergraph partitioning and ordering for PDSLin**

More Information: [http://www.fastmath-scidac.org](http://www.fastmath-scidac.org) or contact any of the team members identified above.