Sara Dadizadeh
CS521 Fall 2009

Sparse Matrix-Vector Multiplication

Motivation

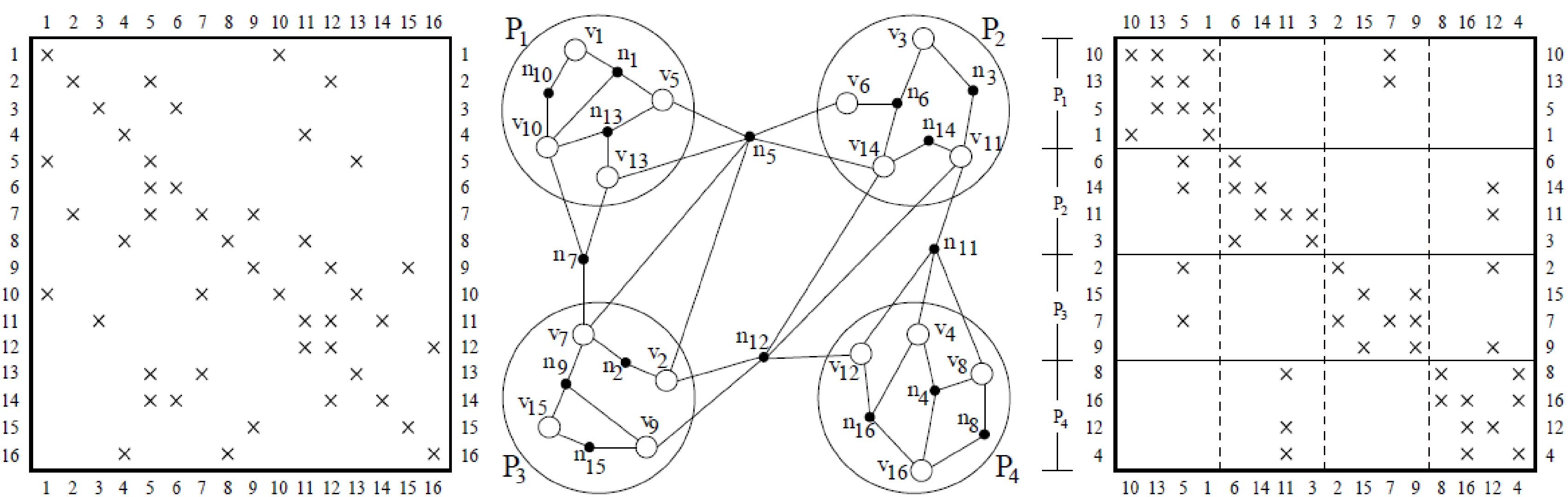
- Sparse matrix multiplication used heavily in scientific computing to solve numerical PDE's
- Goal: product of the form $A\mathbf{x} = \mathbf{y}$
- Possible algorithms
  - Cholesky decomposition (direct)
  - Conjugate gradient method (iterative)
  - Graph/hypergraph decomposition

Hypergraph Representation

- Typical solutions represent matrix as a graph
- We represent matrix as a hypergraph, in column-net model
  - Rows represent vertices
  - Columns represent hyperedges
- Representing matrix as a hypergraph has several advantages
  - Can represent unsymmetric matrices as well as symmetric
  - Hypergraphs describe communication better than graphs (helps partitioning algorithms make better partitions)

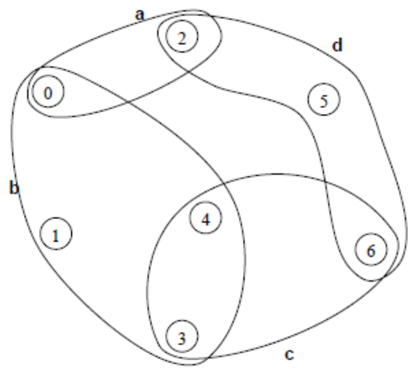
An Example

- Each processor is assigned a subset of the rows of the matrix
- Partition vertices of graph as to minimize the number of edges that are cut
  - Cut edges represent communication



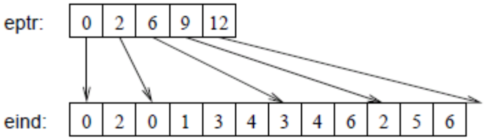
Implementation Details

- Build hypergraph from matrix



Hyperedges

```
0, 2
0, 1, 3, 4
3, 4, 6
2, 5, 6
```

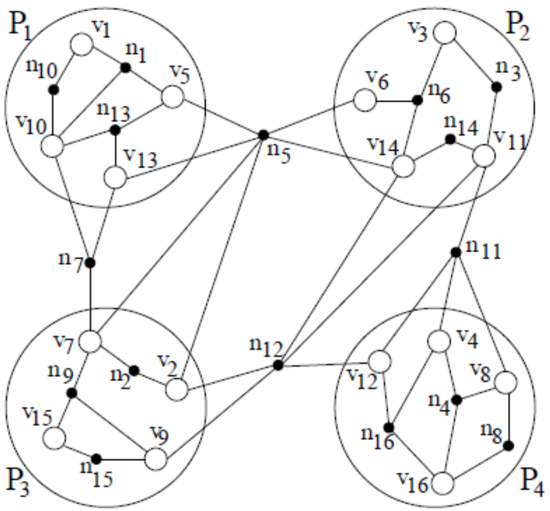


- Call a hypergraph partitioning package called hMetis which returns a partitioning

HMETIS_PartRecursive (int nvtxs, int nhedges, int *vwgts, int *eptr, int *eind, int *hewgts, int nparts, int ubfactor, int *options, int *part, int *edgecut)

Implementation Details (2)

- Assign row i, y_i and x_i to a processor P according to the generated partition
- Create a communication group for each cut net n_j
  - The processor that contains v_j broadcasts its local x_j to all the processors in the connectivity set of n_j (i.e. the group)
  - Each net n_j incurs the computation $y_i = y_i + a_{ij}x_j$ for each vertex v_i in n_j
- In parallel, each processor computes y_i for the vertices it has been assigned
- Finally, combine results to form the result vector **y**



Expected Speedup

- Sparse matrix multiplication does not typically scale well
- As we throw more processors at the problem, the number of cut nets increases, and communication increases faster than computation
- Require certain conditions to be met for speedup
  - Matrix must have the right sparsity index
  - Number of processors must not be too large

Progress

- Currently working on communication groups
- Still to do
  - Gather results into result vector
  - Write a program to perform sequential sparse matrix multiplication
  - Take benchmarks

References

- Figures on slide 5 taken from hMetis reference manual
  - http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download
- Figure on slide 4 taken from
  - Hypergraph-Partitioning Based Decomposition for Parallel Sparse-Matrix Vector Multiplication, Umit V. Catalyurek , Cevdet Aykanat, IEEE Trans. on Parallel and Distributed Computing
  - http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.9288