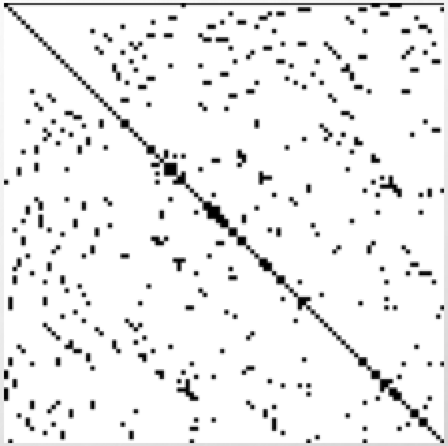
COMBINATORIAL SCIENTIFIC COMPUTING

EFFICIENCY OF SPARSE MATRIX VECTOR MULTIPLICATION

XINGSI ZHONG

SPARSE MATRIX-VECTOR MULTIPLICATION

- SPARSE MATRIX:
  - A matrix populated primarily with zeros.
  - Huge sparse matrices often appear in science or engineering problems
    - Solving partial differential problems.
    - Power and Control Systems Simulation.
    - VLSI design.
    - computation of eigenfrequencies of mechanical constructions.
    - Social network problems.
  - A sparse matrix or system is one in which advantage can be taken of the percentage and/or distribution of zero elements.



A sparse matrix obtained when solving a finite element problem in two dimensions. The non-zero elements are shown in black.

http://en.wikipedia.org/wiki/File:Finite_element_sparse_matrix.png

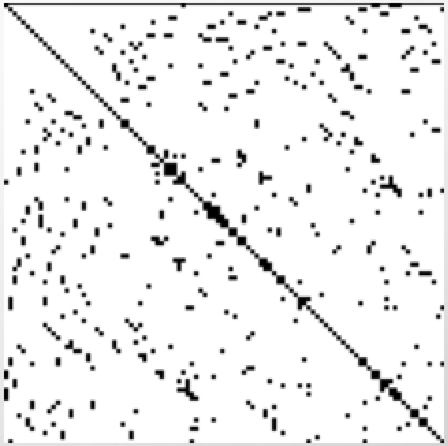
SPARSE MATRIX-VECTOR MULTIPLICATION

- SEVERAL ADVANTAGES THAT CAN BE TAKEN:
  - The information storage and retrieval.

    When only storing nonzero data, much larger and accurate models can be considered.
  - Sparse matrix manipulation and algorithms should be much more efficient than dense matrix manipulation.

    Computation should be organized to avoid operations which involve zeros.

    e.g. matrix-vector multiplication.



A sparse matrix obtained when solving a finite element problem in two dimensions. The non-zero elements are shown in black.

http://en.wikipedia.org/wiki/File:Finite_element_sparse_matrix.png

SPARSE MATRIX STORAGE SCHEMES

- LIST OF LISTS (LIL)

  LIL stores one list per row, where each entry stores a column index and value.

- COORDINATE LIST (COO)

  COO stores a list of `(row, column, value)` tuples. Ideally, the entries are sorted (by row index, then column index) to improve random access times .

- YALE FORMAT
- COMPRESSED SPARSE ROW (CSR OR CRS)
- COMPRESSED SPARSE COLUMN (CSC OR CCS)

SPARSE MATRIX STORAGE SCHEMES

- COMPRESSED SPARSE COLUMN FORMAT (CSC OR CCS)

0	3	0	0	0
22	0	0	0	17
7	5	0	1	0
0	0	0	0	0
0	0	14	0	8

val:	22	7	3	5	14	1	17	8
row_ind:	1	2	0	2	4	2	1	4
col_ptr:	0	2	4	5	6			

SPARSE MATRIX STORAGE SCHEMES

- COMPRESSED SPARSE ROW FORMAT (CSR OR CRS)

0	3	0	0	0
22	0	0	0	17
7	5	0	1	0
0	0	0	0	0
0	0	14	0	8

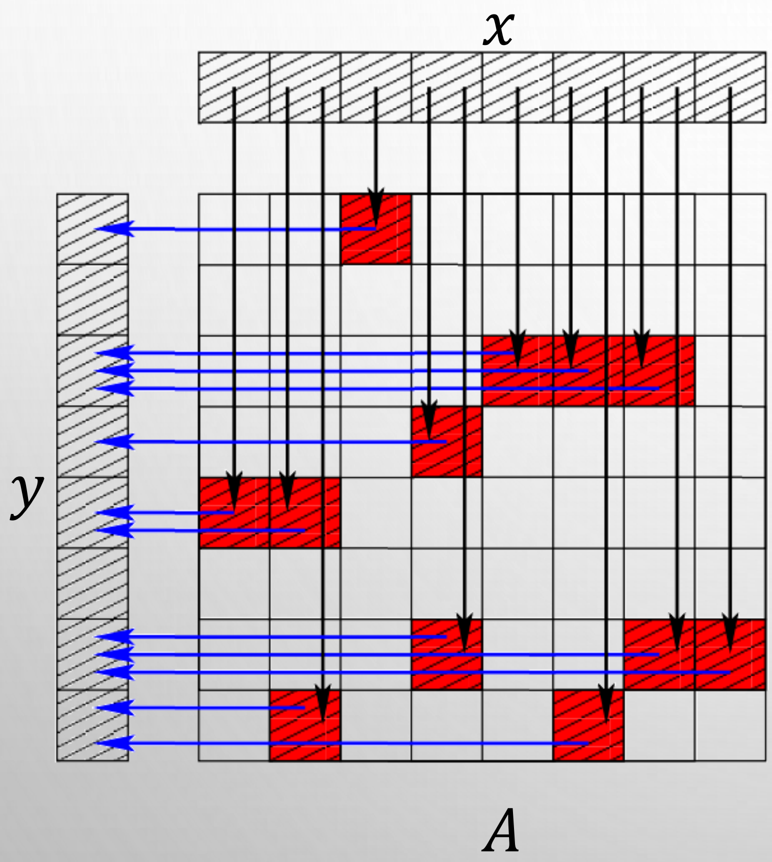
→

val:	3	22	17	7	5	1	14	8
col_ind:	1	0	4	0	1	3	2	4
row_ptr:	0	1	3	_	6			

Pseudo code for a n by m sparse matrix-vector multiplication y=Ax for CRS storage scheme

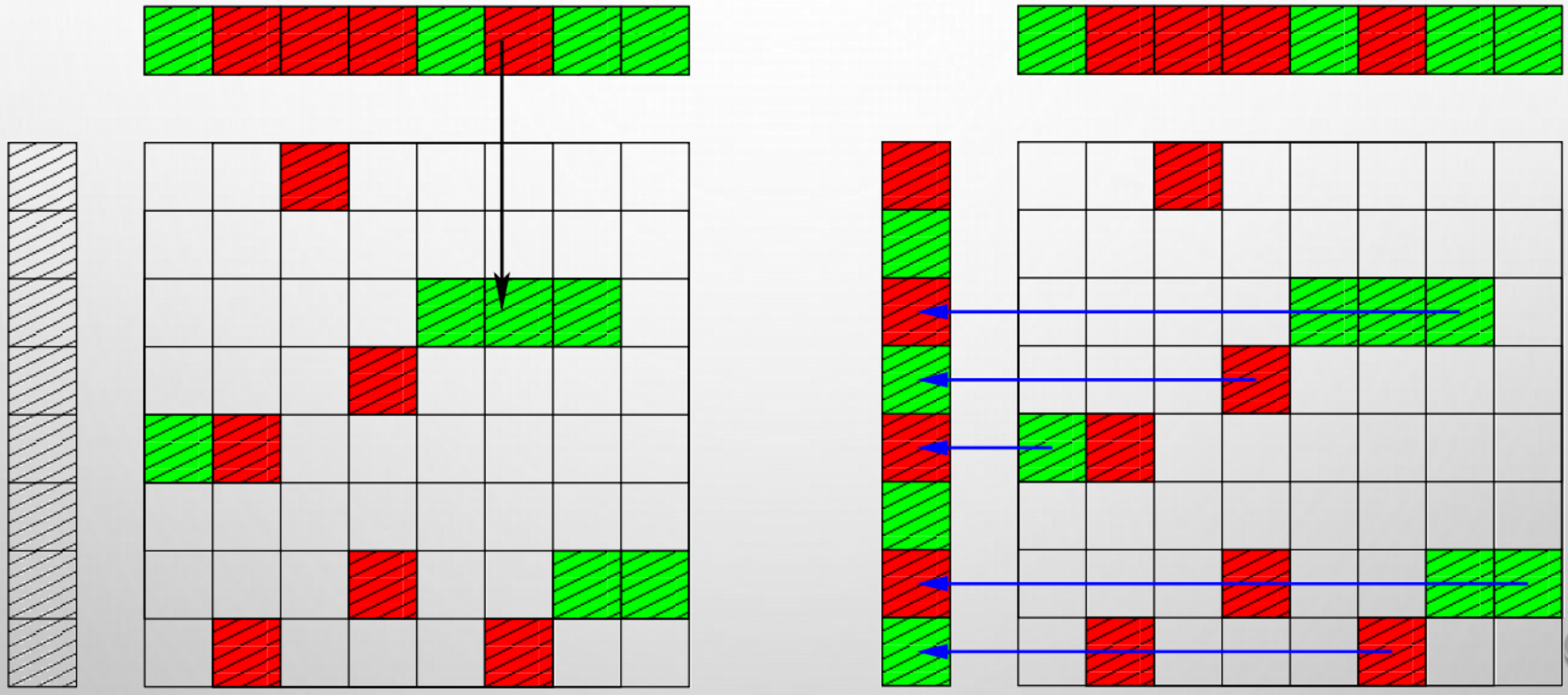
```
for i = 1, n
    y(i)  = 0
    for j = row_ptr(i), row_ptr(i+1) - 1
        y(i) = y(i) + val(j) * x(col_ind(j))
    end;
end;
```

PROBLEMS IN PARALLEL COMPUTING: WORKLOAD, COMMUNICATION LOAD

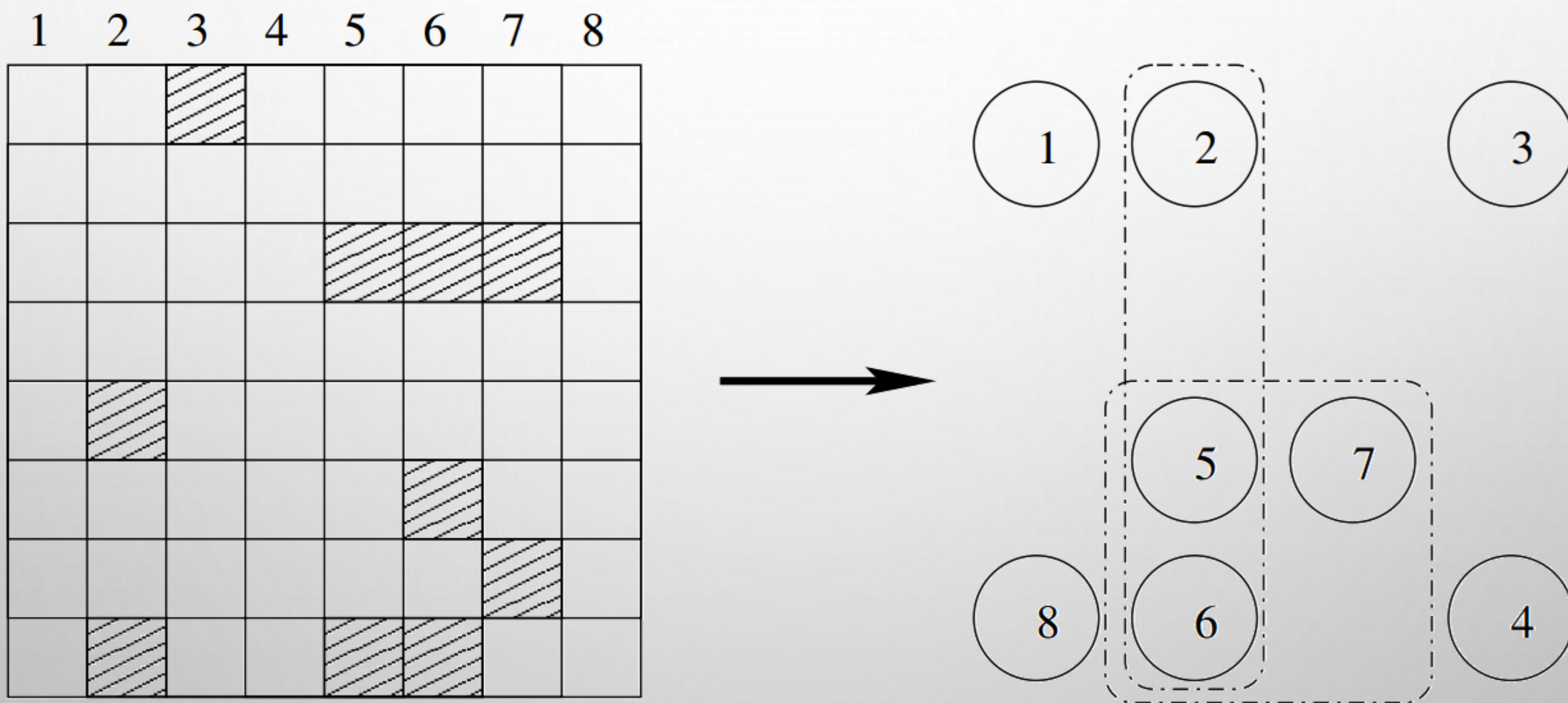


$$y = Ax$$

PROBLEMS IN PARALLEL COMPUTING: COMMUNICATION LOAD



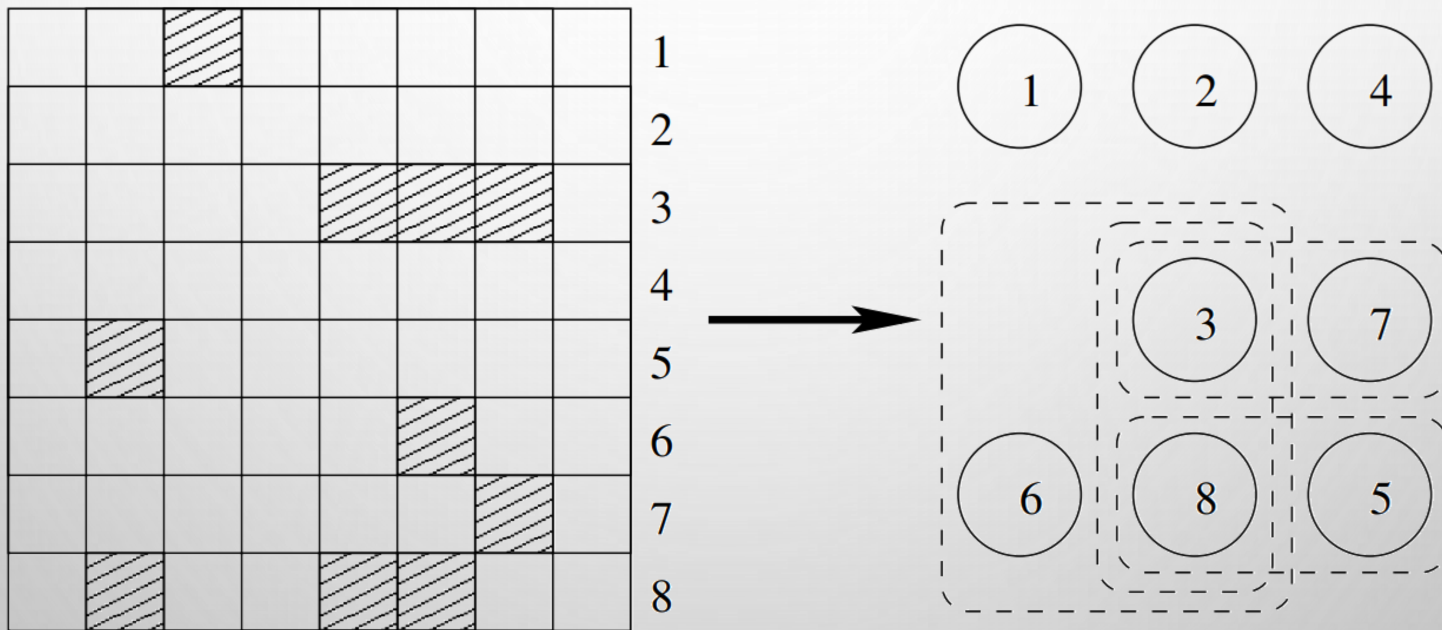
COMMUNICATION LOAD



Nonzeroes on the same row distributed to different processors during read from memory.

http://people.cs.kuleuven.be/~albert-jan.yzelman/slides/basel10.pdf

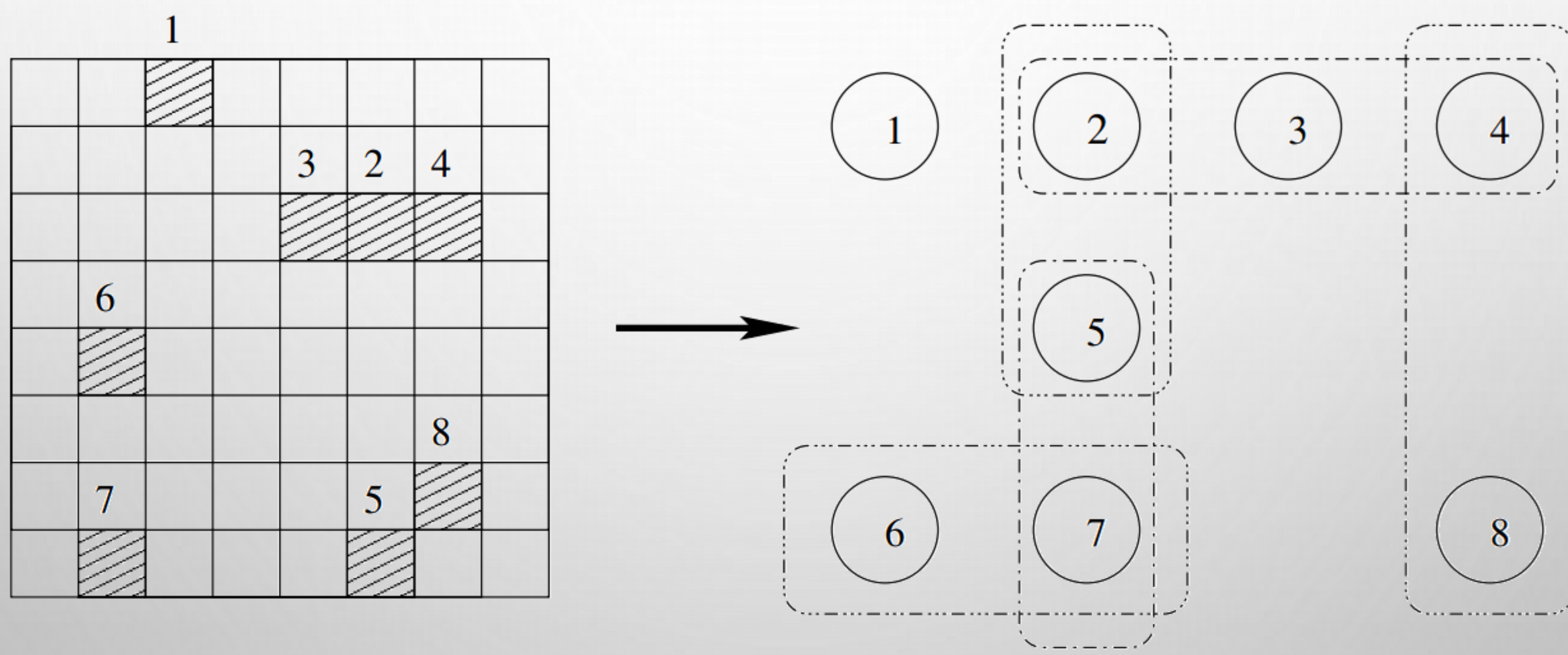
COMMUNICATION LOAD



Nonzeroes on the same column distributed to different processors during write to memory.

http://people.cs.kuleuven.be/~albert-jan.yzelman/slides/basel10.pdf

COMMUNICATION LOAD



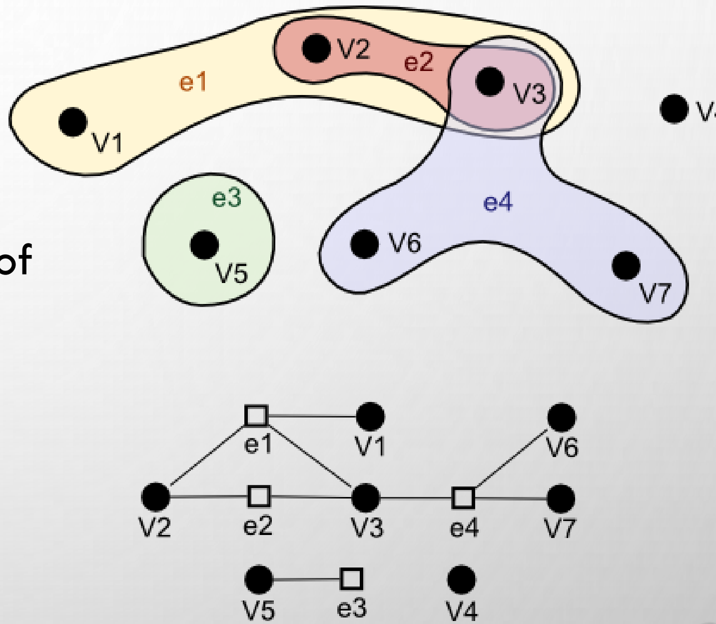
Fine-grain model: consider the communication of both read and write.

http://people.cs.kuleuven.be/~albert-jan.yzelman/slides/basel10.pdf

HYPERGRAPH PARTITION PROBLEM

- HYPEREDGE (NET):

  A pair of sets H = (V, E). V is the set of vertices of the hypergraph and E is the set of hyperedges of the hypergraph. Each hyperedge in a hypergraph is a non-empty subset of V, the size of this subset is called the hyperedge's degree. A weighted hypergraph has non-negative numeric weights associated with each vertex, each hyperedge, or both.



http://sharpen.engr.colostate.edu/mediawiki/index.php/File:Hypergraph.png

HYPERGRAPH PARTITION PROBLEM

- The k-way Hypergraph Partitioning Problem:

  Given a hypergraph $H = (V, E)$ find a k-way partitionment $\delta : V \rightarrow P$ that maps the vertices of H to one of k disjoint partitions such that some cost function $c : \delta \rightarrow R$ is minimized.

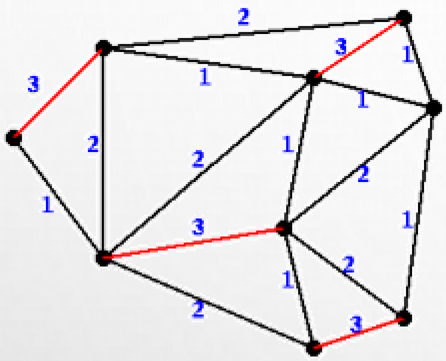
  Balance Constraint:

  A pair of values (l, u). A partition P with a balance constraint must obey $l \leq \sum_{v \in \mathrm{P}} W(v) \leq u$, where $W(v)$ is the weight of vertex v or 1 if the vertex has no weight.

  Cut:

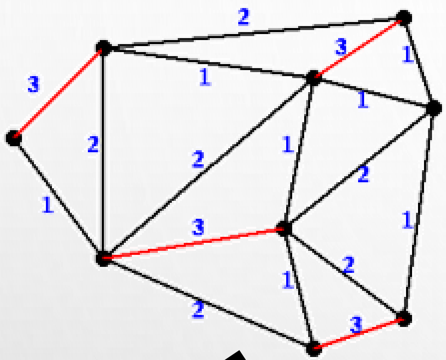
  A hyperedge e of hypergraph is cut if, with respect to a particular partitionment δ, its vertices are mapped to more than one partition. The net cut of a partitionment is the total number of hyperedges that are cut. The weighted net cut of a partitionment is the sum of the weights of hyperedges that are cut.

MULTILEVEL ALGORITHM

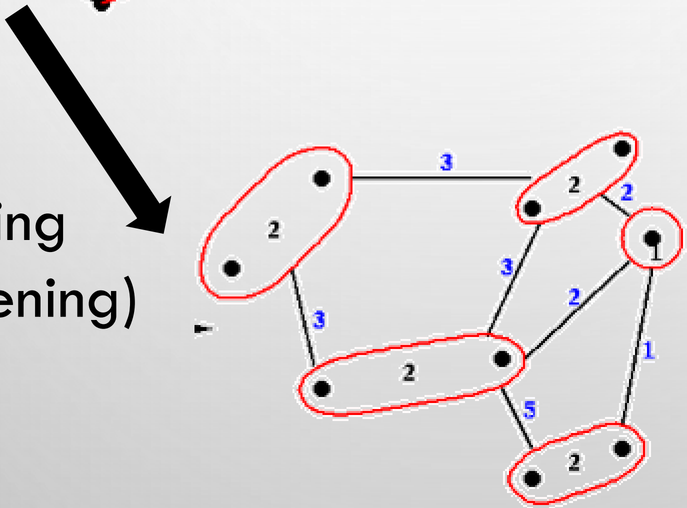


http://www.cs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html

MULTILEVEL ALGORITHM

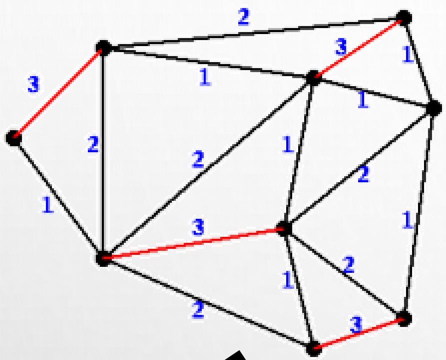


Clustering
(Coarsening)

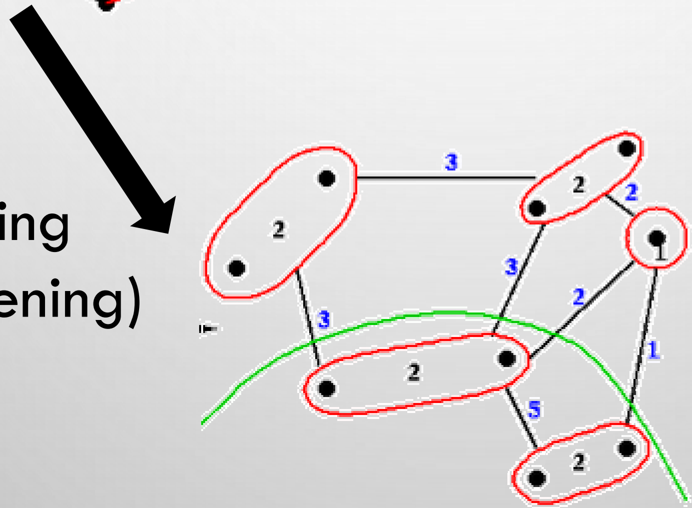


http://www.cs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html

MULTILEVEL ALGORITHM



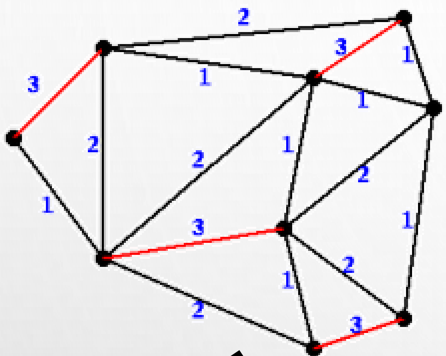
Clustering
(Coarsening)

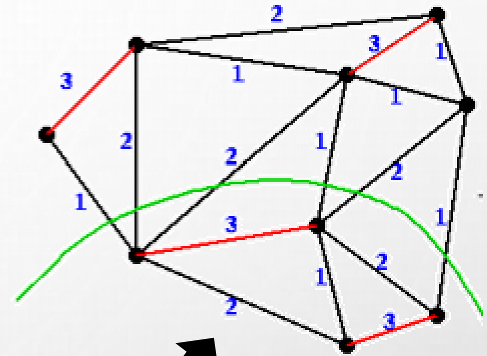


Top-level partitioning

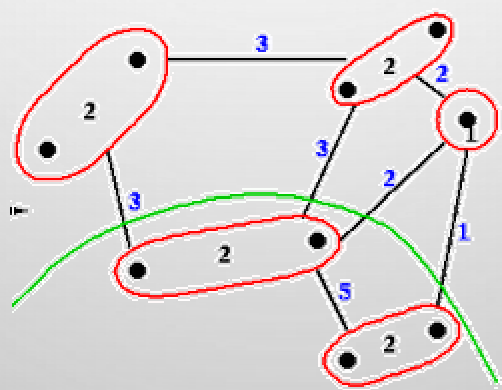
http://www.cs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html

MULTILEVEL ALGORITHM





Clustering
(Coarsening)

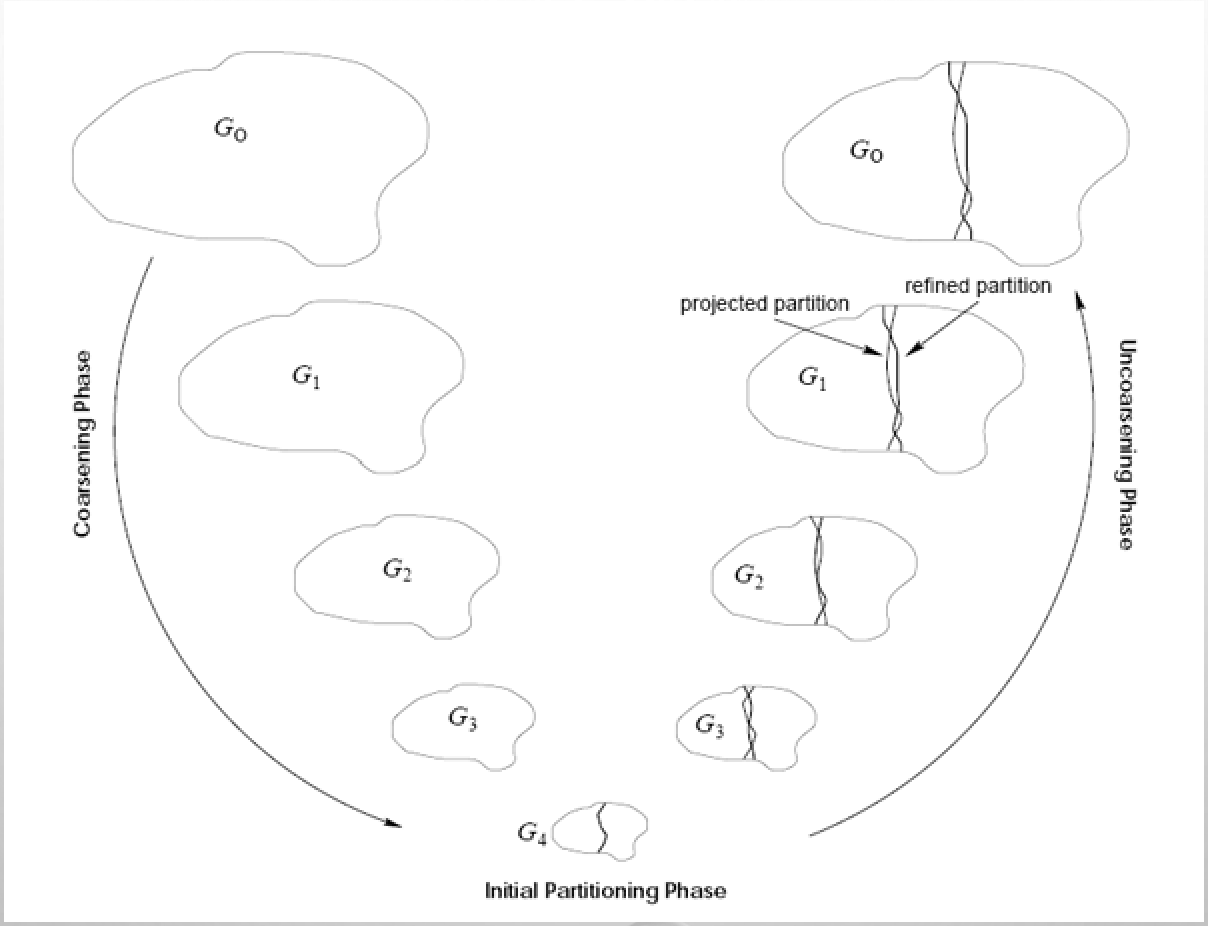


Renement
(Uncoarsening)

Top-level partitioning

http://www.cs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html

MULTILEVEL ALGORITHM



http://masters.donntu.edu.ua/2006/fvti/shepel/diss/indexe.htm

MONDRIAAN

PARTITIONING SOFTWARE FOR SPARSE MATRIX COMPUTATIONS
DEPARTMENT OF MATHEMATICS
UTRECHT UNIVERSITY
HTTP://WWW.STAFF.SCIENCE.UU.NL/~BISSE101/MONDRIAAN/

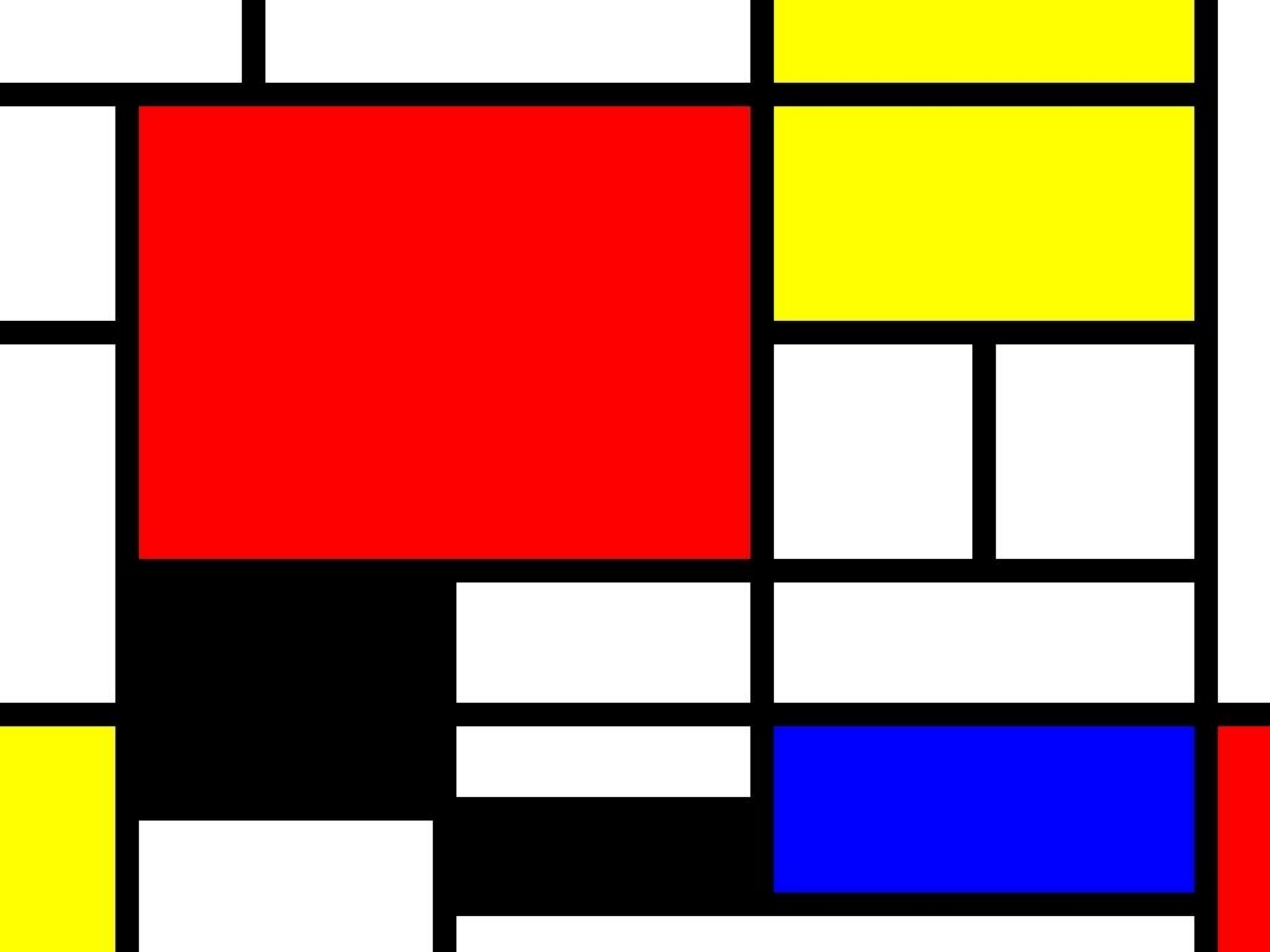
- CURRENT VERSION:

  Version 4.0 August 29, 2013

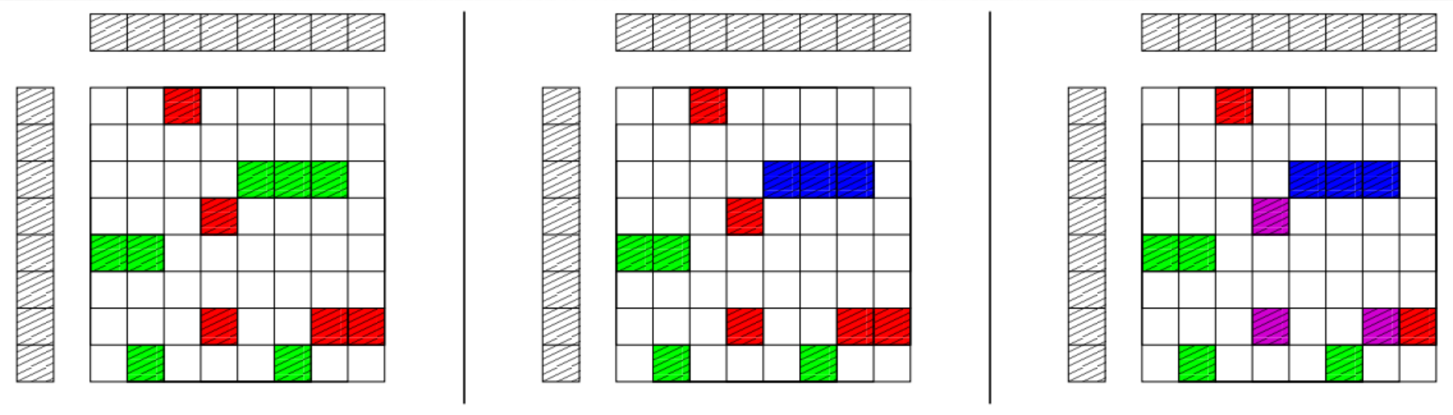
- FIRST RELEASE:

  Version 1.0 May 10, 2002 by Rob Bisseling

Piet Mondrian 1872 - 1944

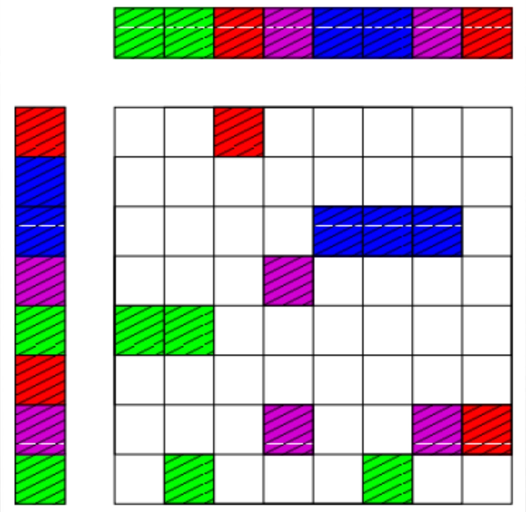


MONDRIAAN 2D MATRIX PARTITIONING



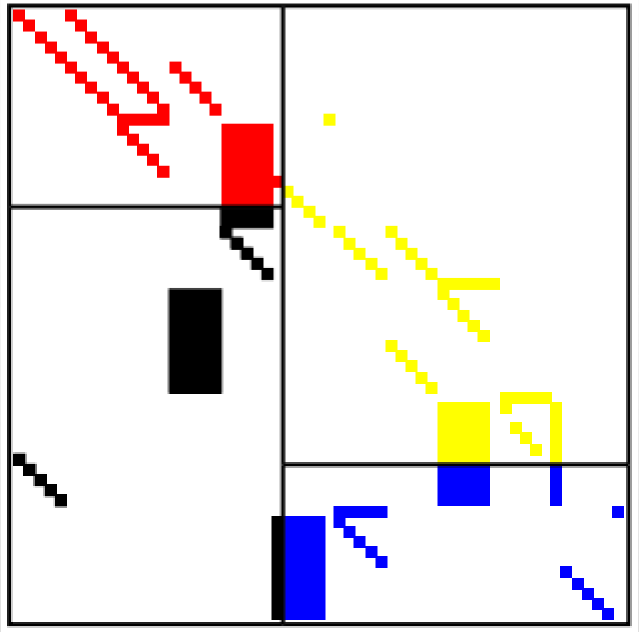
- Model the sparse matrix using a hypergraph
- Partition the vertices of the hypergraph (in two)
- Recursively keep partitioning the vertex parts

MONDRIAAN 2D MATRIX PARTITIONING



- Model the sparse matrix using a hypergraph
- Partition the vertices of the hypergraph (in two)
- Recursively keep partitioning the vertex parts
- Partition the vector elements

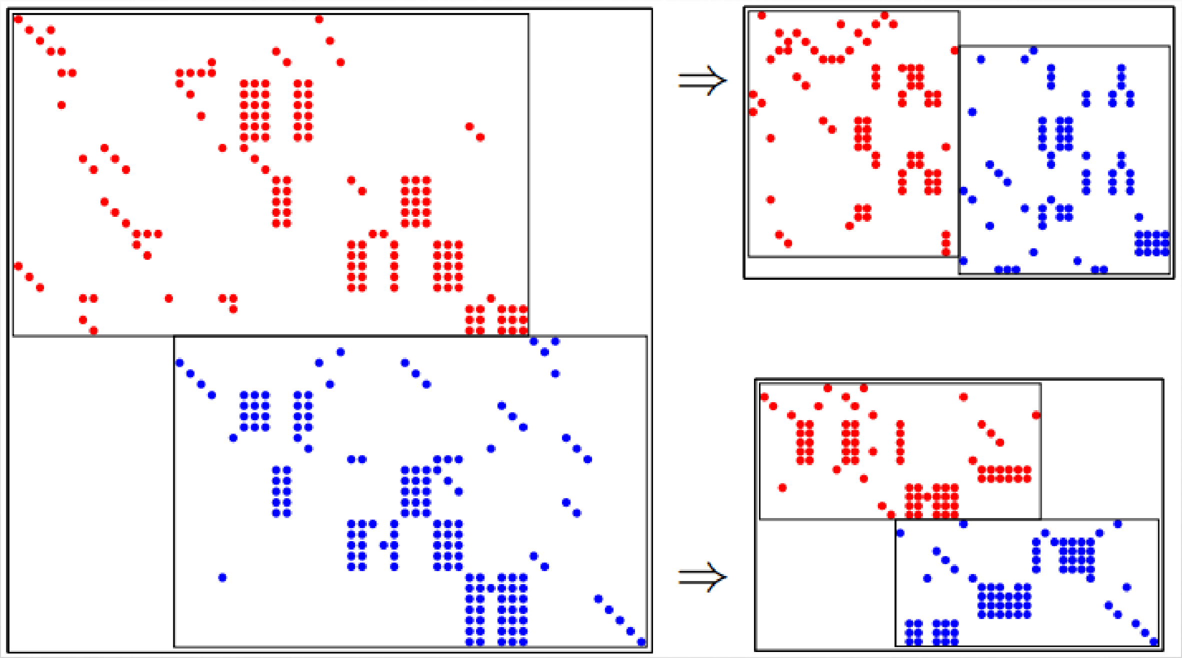
MONDRIAAN 2D MATRIX PARTITIONING



Block distribution (without row/column permutations) of 59 × 59 matrix impcol_b with 312 nonzeros, for p = 4

Mondriaan package v1.0 (May 2002). Originally developed by Vastenhouw and Bisseling for partitioning term-by-document matrices for a parallel web search machine.

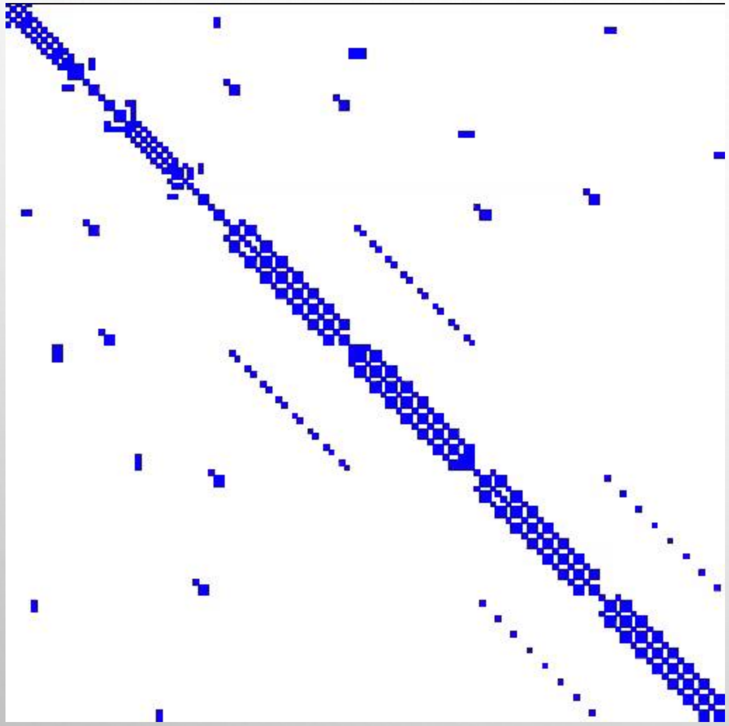
MONDRIAAN 2D MATRIX PARTITIONING

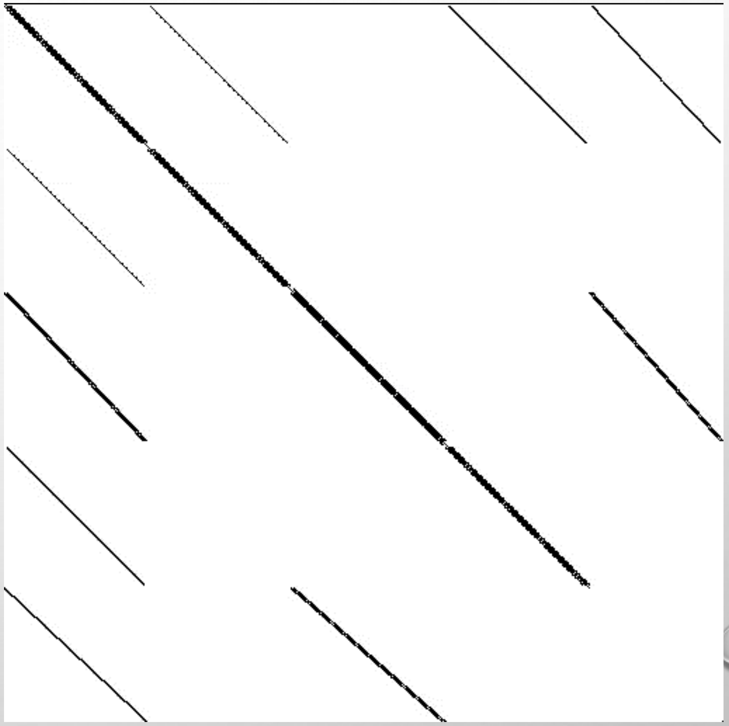


Recursively split the matrix into 2 parts

Try splits in row and column directions, allowing permutations. Each time, choose the best direction

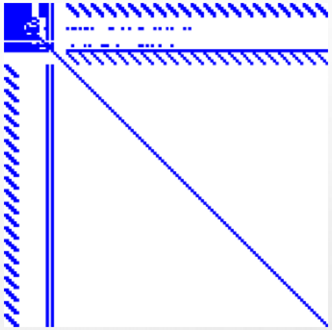
MONDRIAAN GRAPH OUTPUT

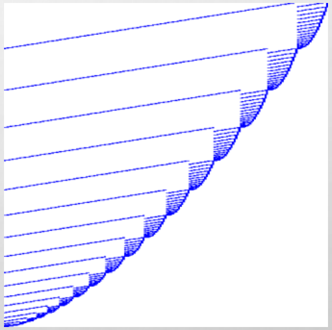


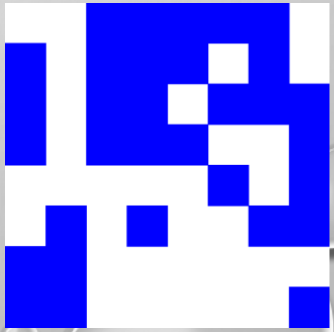


MONDRIAAN

- Run as standalone program or library, and interface for MATLAB.
- Input: Matrix Market format (.mtx)
- Output:
  - DISTRIBUTED MATRIX
  - PROCESSOR INDICES
  - ROW AND COLUMN PERMUTATIONS
  - REORDERED MATRIX
  - STATISTICS ON STANDARD OUTPUT
  - GRAPHICAL OUTPUT
  - ……
- Mondriaan can use PaToH as a hypergraph bipartitioner.
- Nets cut quality are good but not the best, and it is slow (a sequential program).







SCOTCH

The SCOTCH distribution is a set of programs and libraries which implement the static mapping and sparse matrix reordering algorithms developed within the SCOTCH project.

UNIVERSITE BORDEAUX

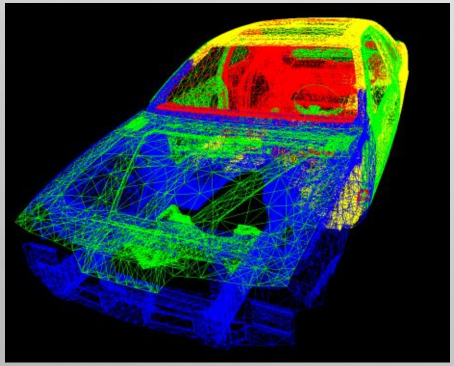
http://www.labri.fr/perso/pelegrin/scotch/

- CURRENT VERSION:

  Version 6.0 December 2 2012

- FIRST RELEASE:

  1992 by François Pellegrini.



SCOTCH AND PT-SCOTCH

- Scotch, a sequential software package devoted to static mapping, graph and mesh partitioning, and sparse matrix block ordering.
- PT-Scotch, a software package devoted to parallel static mapping and sparse matrix block ordering. It is the parallel extension of Scotch.

Both packages share a significant amount of code, because PT-Scotch transfers control to the sequential routines of the ibScotch library when the subgraphs on which it operates are located on a single processor

SCOTCH AND PT-SCOTCH

Algorithms:

- Parallel static mapping by Dual Recursive Bipartitioning
  - Static mapping
  - Cost function and performance criteria
  - The Dual Recursive Bipartitioning algorithm (a divide and conquer approach to recursively allocate subsets of processes to subsets of processors)
  - Partial cost function
  - **Parallel graph bipartitioning methods**
  - Mapping onto variable-sized architectures

- Parallel sparse matrix ordering by hybrid incomplete nested dissection
  - Hybrid incomplete nested dissection
  - Parallel ordering
  - Performance criteria

PARALLEL GRAPH BIPARTITIONING METHODS

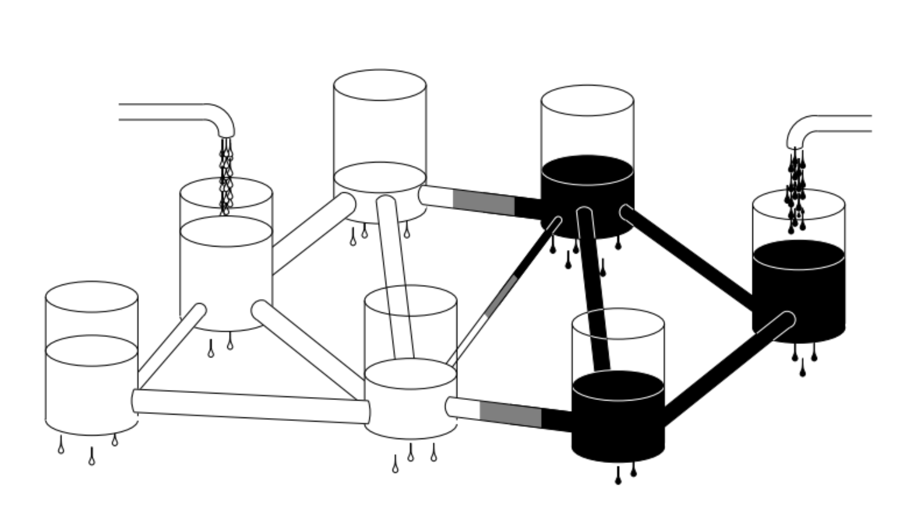
- BAND

  The band method is a meta-algorithm, in the sense that it does not itself compute partitions, but rather helps other partitioning algorithms perform better. It is a refinement algorithm which, from a given initial partition, extracts a band graph of given width (which only contains graph vertices that are at most at this distance from the separator), calls a partitioning strategy on this band graph, and prolongs1 back the refined partition on the original graph.

PARALLEL GRAPH BIPARTITIONING METHODS

- DIFFUSION

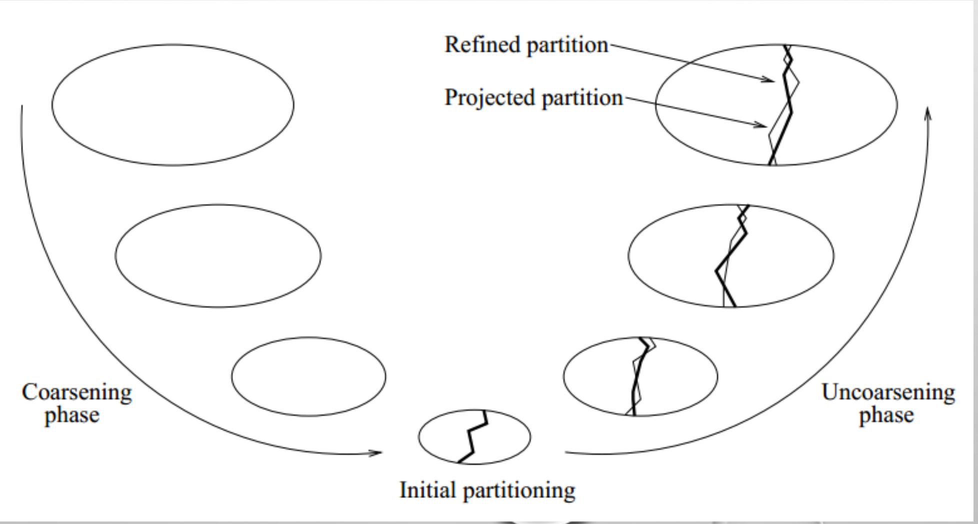
  This global optimization method, flows two kinds of antagonistic liquids, scotch and anti-scotch, from two source vertices, and sets the new frontier as the limit between vertices which contain scotch and the ones which contain anti-scotch. In order to add load-balancing constraints to the algorithm, a constant amount of liquid disappears from every vertex per unit of time, so that no domain can spread across more than half of the vertices.



PARALLEL GRAPH BIPARTITIONING METHODS

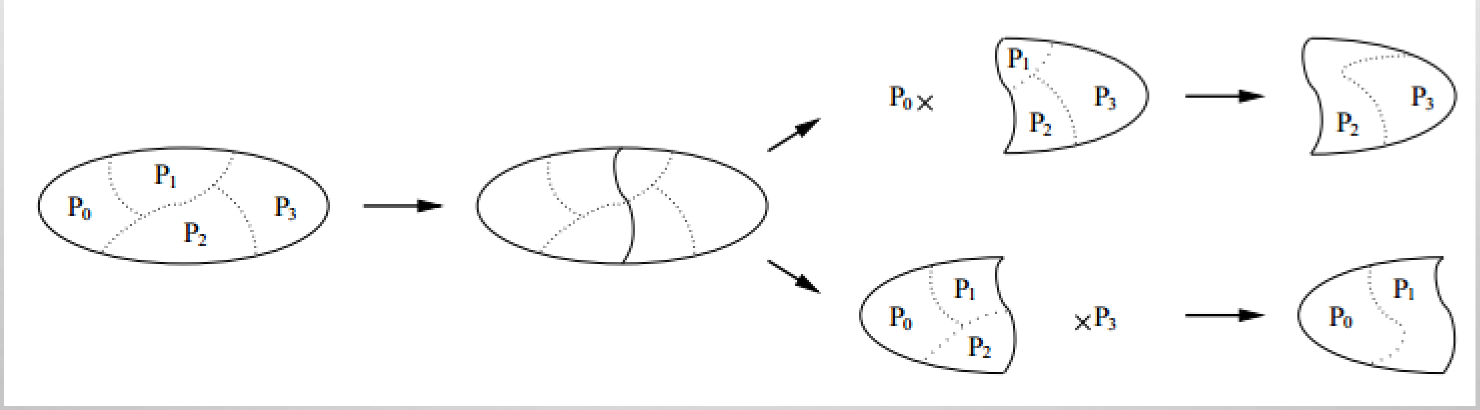
- MULTI-LEVEL

  Repeatedly reduces the size of the graph to bipartitionby finding matchings that collapse vertices and edges, computes a partition for the coarsest graph obtained, and prolongs the result back to the original graph. The multi-level method, when used in conjunction with the banded diffusion method to refine the prolonged partitions at every level, usually stabilizes quality irrespective of the number of processors which run the parallel static mapper.



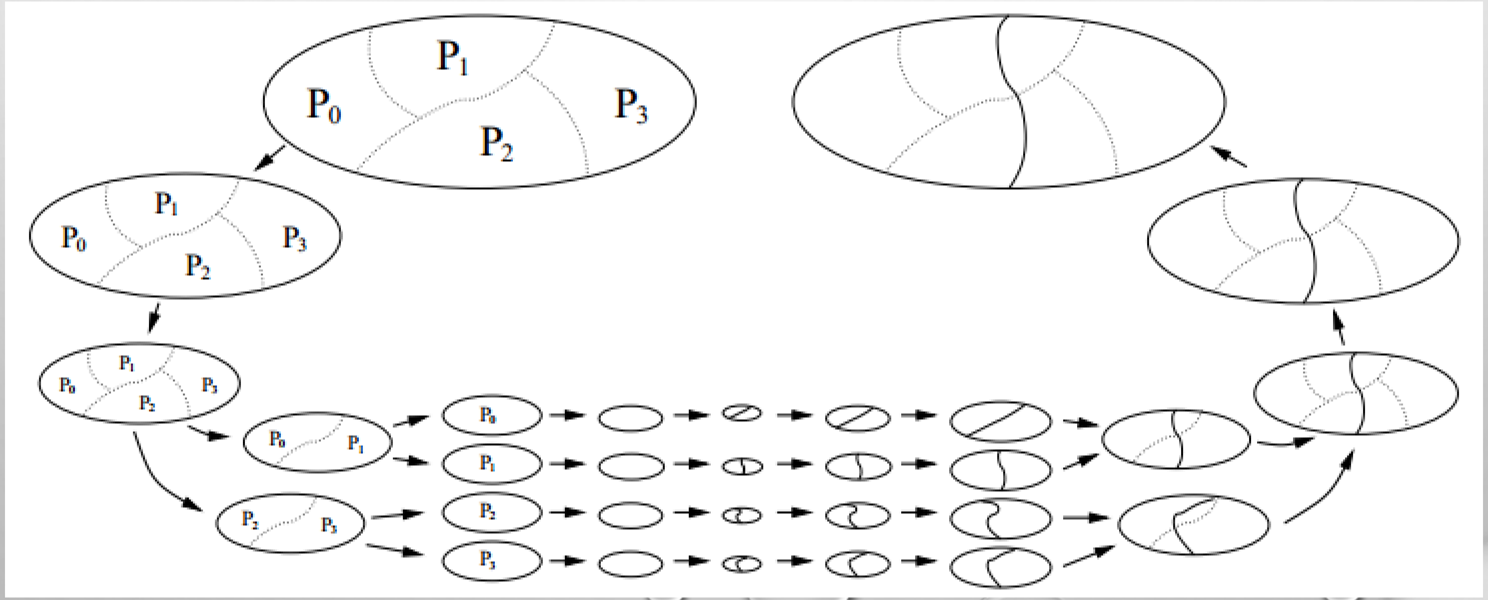
NESTED DISSECTION

Diagram of a nested dissection step for a (sub-)graph distributed across four processors. Once the separator is known, the two induced subgraphs are built and folded (this can be done in parallel for both subgraphs), yielding two subgraphs, each of them distributed across two processors.



PARALLEL COMPUTATION OF THE SEPARATOR

Diagram of the parallel computation of the separator of a graph distributed across four processors, by parallel coarsening with folding-with-duplication in the last stages, multi-sequential computation of initial partitions that are locally prolonged back and refined on every processor, and then parallel uncoarsening of the best partition encountered.



SCOTCH AND PT-SCOTCH

- A set of stand-alone programs as well as through the libSCOTCH library, which offers both C and Fortran interfaces.
- Input output format:
  - centralized graph files in the Scotch format
  - distributed graph format (.dgr)
- It can be easily interfaced to other programs. The programs comprising the SCOTCH project have been designed to run in command-line mode without any interactive prompting, so that they can be called easily from other programs by means of system() or popen() calls, or piped together on a single command line
- It provides many tools to build, check, and display graphs, meshes and matrix patterns.
- It is written in C and uses the POSIX interface, which makes it highly portable. PT-SCOTCH uses the MPI interface, and optionally the POSIX threads.

hMETIS

A SET OF PROGRAMS FOR PARTITIONING HYPERGRAPHS
UNIVERSITY OF MINNESOTA
HTTP://GLAROS.DTC.UMN.EDU/GKHOME/METIS/HMETIS/OVERVIEW

- CURRENT VERSION:

  VER: 2.0PRE1, MAY 25, 2007

- FIRST RELEASE:

  VER: 1.0.1, FEBRUARY 12, 1998 BY KARYPIS LIB

hMETIS is the hpyergraph version of METIS, a multilevel graph partitioning algorithm implemented in C. hMETIS partitioning program introduced several new heuristics that are incorporated into their multilevel partitioning implementation and are reportedly performance-critical

hMETIS

A SET OF PROGRAMS FOR PARTITIONING HYPERGRAPHS
UNIVERSITY OF MINNESOTA
HTTP://GLAROS.DTC.UMN.EDU/GKHOME/METIS/HMETIS/OVERVIEW

- hMETIS contains a set of stand alone programs, as well as livrary interfaces.
- The input is one file with some very simple format, and the output is specified by giving the partition of the $i-th$ vertex on the $i-th$ line of the solution file.
- It clams to provides high quality partitions, and it is extremely fast.

GRACLUS

COMPUTES GRAPH CUTS WITHOUT EIGENVECTORS COMPUTATION

HTTP://WWW.CS.UTEXAS.EDU/USERS/DML/SOFTWARE/GRACLUS.HTML

- CURRENT VERSION:

  VERSION 1.2

  A fast graph clustering software that computes normalized cut and ratio association for a given undirected graph without any eigenvector computation.

  The authors found that there is mathematical equivalence between general cut or association objectives (including normalized cut and ratio association) and the weighted kernel k-means objective. One important implication of this equivalence is that we can run a k-means type of iterative algorithm to minimize general cut or association objectives. Therefore unlike spectral methods, our algorithm totally avoids time-consuming eigenvector computation. We have embedded the weighted kernel k-means algorithm in a multilevel framework to develop very fast software for graph clustering.

GRACLUS

COMPUTES GRAPH CUTS WITHOUT EIGENVECTORS COMPUTATION

- SPECTRAL PARTITIONING:

  Set of eigenvalues of a matrix A is called the spectrum of A.

  Spectral partitioning is a global partitioning method.

  In general, k-th smallest eigenvalue/eigenvector gives a K-way partitioning.

  The expense is too large.

- GRACLUS:

  Achieves an even better(quality and speed) global solution without using eigenvectors.

  IMDB movie data set test: (The IMDB contains 1.4 million nodes and 4.3 million edges.)

Normalized cut values—lower cut values are better

# clusters	2	4	8	16	32	64	128	256
Graclus	.049	.163	.456	1.39	3.72	9.42	24.13	64.04
Spectral	.00	.016	.775	2.34	5.65	-	-	-

Computation time (in seconds)

# clusters	2	4	8	16	32	64	128	256
Graclus	34.57	37.3	37.96	46.61	49.93	53.95	64.83	81.42
Spectral	261.32	521.69	597.23	1678.05	5817.96	-	-	-

http://www.cs.utexas.edu/users/inderjit/Talks/multilevel.pdf

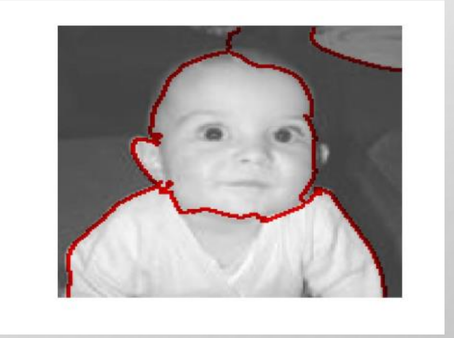
GRACLUS

COMPUTES GRAPH CUTS WITHOUT EIGENVECTORS COMPUTATION

SOURCE CODE:

A stand alone program

Matlab interface



PATOH

PARTITIONING **TO**OLS FOR **H**YPERGRAPH

HTTP://BMI.OSU.EDU/~UMIT/SOFTWARE.HTML

- CURRENT VERSION:

  PaToH v3.2 Mar 22, 2011

- FIRST VERSION:

  1999 by Ümit V. Çatalyürek

"It was the fastest hypergraph partitioner when I wrote it, and probably it is still the fastest sequential partitioner today."

You can also use patoh via ZOLTAN for hypergraph partitioning or via MONDRIAAN for matrix partitioning.

PATOH

Comparison of hMeTiS and PaToH on 134 hypergraphs arising from different areas, Sparse-Matrix Vector Multiplication, LP, VLSI (including ISPD98).

	PaToH param=SPEED				PaToH param=DEFAULT				PaToH param=QUALITY			
	Cut Size			Exec	Cut Size			Exec	Cut Size			Exec
	min	max	avg	Time	min	max	avg	Time	min	max	avg	Time
Others	1.01	2.25	1.12	0.07	1.01	0.88	1	0.15	1	0.88	0.99	0.61
LP	0.98	1.39	1.05	0.15	0.96	0.95	0.94	0.25	0.95	0.91	0.91	0.88
Matrix	1.01	1.46	1.13	0.16	1.01	1	1	0.25	1	0.94	0.97	0.78
VLSI	1.05	1.43	1.22	0.11	1.03	1.05	1.04	0.27	1	0.98	0.99	1.04
Overall	1.01	1.45	1.13	0.14	0.99	0.99	0.99	0.25	0.98	0.94	0.95	0.89

PATOH

- Could perform hypergraph partitioning with fixed cells.
- Multi-constraint hypergraph partitioner.
- Very flexible (could run really fast if you don't care about the quality of the result).
- User friendly software.
- Stand alone software as well as Matlab interface
- Offer source files and binary distributions for different architectures.

    Linux: 32-bit x86-based, 64-bit x86-based

    Mac OS X 10.6: 32-bit x86-based , 64-bit x86-based

    Mac OS X 10.5: 32-bit x86-based , 64-bit x86-based (last update: Oct 9, 2009)

    Linux: 64-bit IA64-based (last update: Nov 24, 2008)

    Mac OS X: 32-bit PowerPC (last update: May 12, 2006)

    Sun Solaris (last update: May 12, 2006)

    IBM AIX (last update: Nov 11, 2004)

ZOLTAN

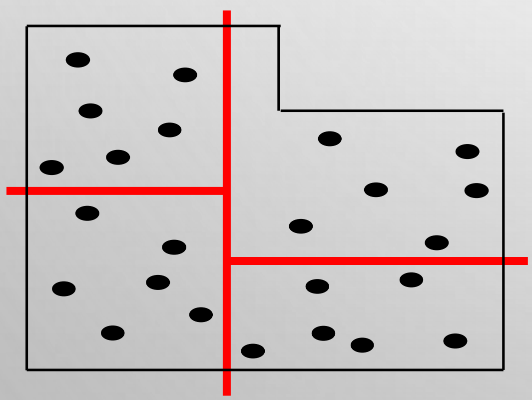
[HTTP://WWW.CS.SANDIA.GOV/ZOLTAN/](http://www.cs.sandia.gov/zoltan/)

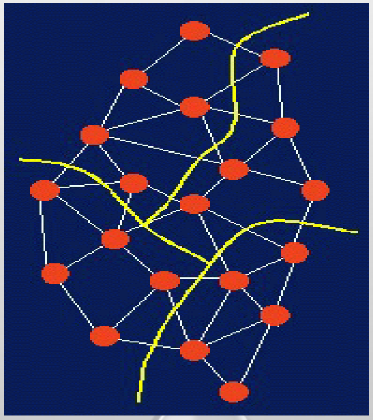
- CURRENT VERSION:

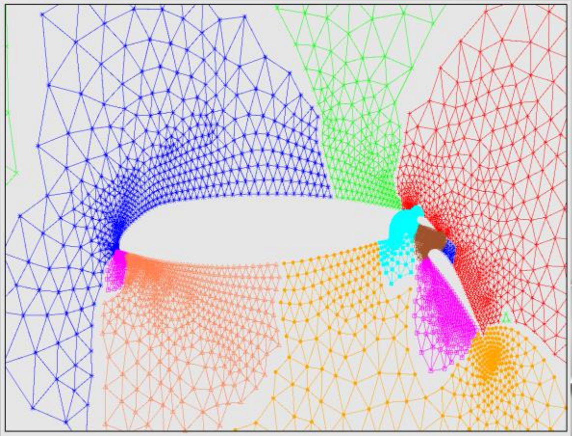
  V3.8 OCTOBER 2013

  BY SANDIA NATIONAL LABORATORIES

Sandia National Laboratories







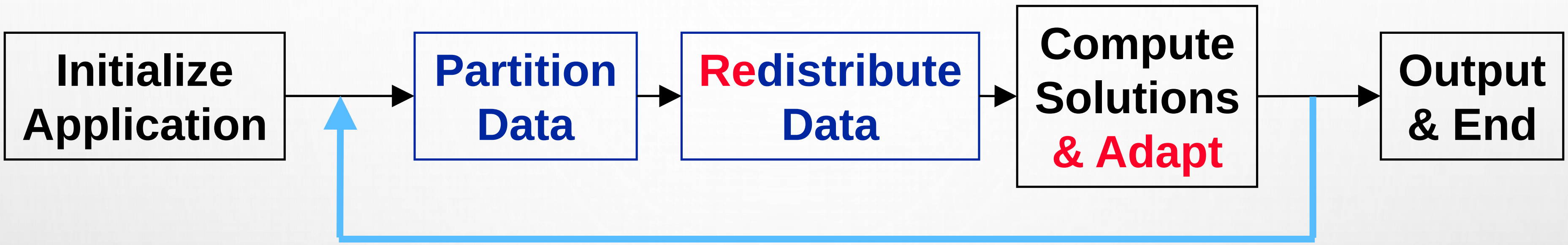
ZOLTAN

- DYNAMIC LOAD BALANCING
- Between each iteration the structure of the problems (computation) may change slightly, but usually not much. After a certain number of iterations, a load balancer is called to rebalance the workloads. The required data is then moved (migrated) between parts to establish the new partitioning, and the computation continues.

STATIC PARTITIONING

- STATIC PARTITIONING IN AN APPLICATION:
  - DATA PARTITION IS COMPUTED.
  - DATA ARE DISTRIBUTED ACCORDING TO PARTITION MAP.
  - APPLICATION COMPUTES.
- IDEAL PARTITION:
  - PROCESSOR IDLE TIME IS MINIMIZED.
  - INTER-PROCESSOR COMMUNICATION COSTS ARE KEPT LOW.

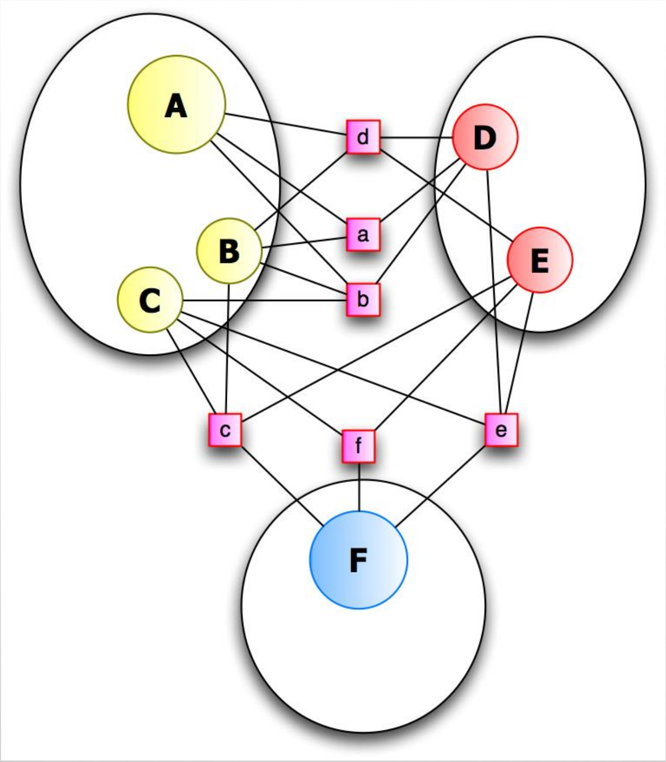
DYNAMIC REPARTITIONING



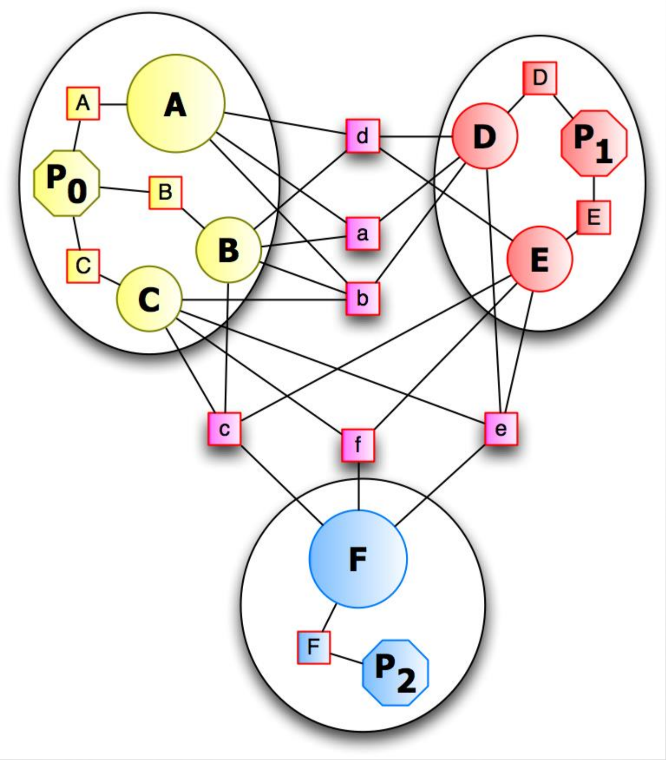
- DYNAMIC REPARTITIONING (LOAD BALANCING) IN AN APPLICATION:
  - DATA PARTITION IS COMPUTED.
  - DATA ARE DISTRIBUTED ACCORDING TO PARTITION MAP.
  - APPLICATION COMPUTES AND, PERHAPS, ADAPTS.
  - PROCESS REPEATS UNTIL THE APPLICATION IS DONE.

- IDEAL PARTITION:
  - PROCESSOR IDLE TIME IS MINIMIZED.
  - INTER-PROCESSOR COMMUNICATION COSTS ARE KEPT LOW.
  - COST TO REDISTRIBUTE DATA IS ALSO KEPT LOW.

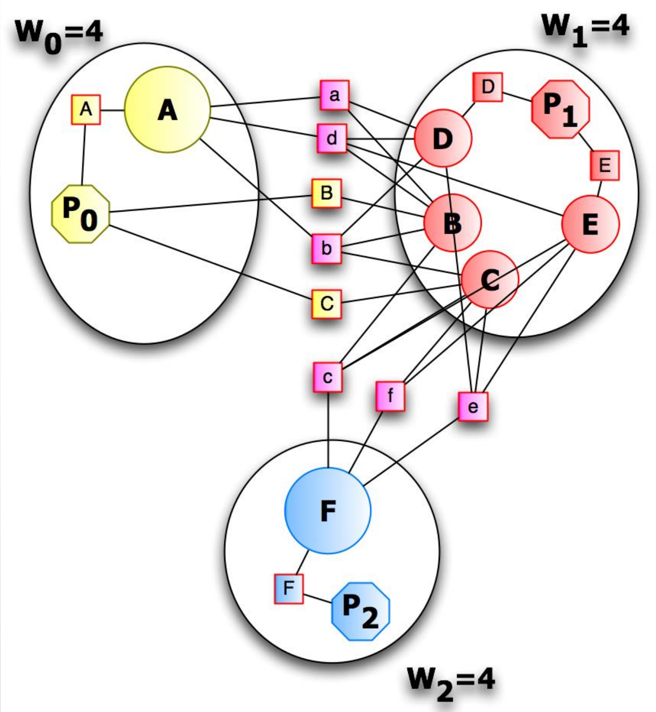
COST TO REDISTRIBUTE DATA



COST TO REDISTRIBUTE DATA



COST TO REDISTRIBUTE DATA



Comm Vol in App = 6 x (2-1) = 6 elements
Mig Cost = 2 x (2-1) = 2 objects
cut(P) = 8

ZOLTAN

- Large program oriented.
  - Based on MPI
  - Successfully used on up to 20K cores
- Fairly simple, easy-to-use interface.
  - Small number of callable Zoltan functions.
  - Callable from C, C++, Fortran.
- Supply easy access for experimentation and comparisons.
  - Application changes only one parameter to switch methods.
- Multilevel and parallel program
- Contains features from mondriaan and patoh and many other algorithm

REFERENCE

- DUFF, I.S., "A SURVEY OF SPARSE MATRIX RESEARCH," PROCEEDINGS OF THE IEEE , VOL.65, NO.4, PP.500,535, APRIL 1977
- DEVINE, KAREN D., ET AL. "PARALLEL HYPERGRAPH PARTITIONING FOR SCIENTIFIC COMPUTING." PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM, 2006. IPDPS 2006. 20TH INTERNATIONAL. IEEE, 2006.
- ILYA SAFRO. “MULTILEVEL ALGORITHMS FOR COMBINATORIAL OPTIMIZATION PROBLEMS” PHD THESIS 2007.
- BOMAN, ERIK G., ET AL. "ADVANCES IN PARALLEL PARTITIONING, LOAD BALANCING AND MATRIX ORDERING FOR SCIENTIFIC COMPUTING." JOURNAL OF PHYSICS: CONFERENCE SERIES. VOL. 180. NO. 1. INSTITUTE OF PHYSICS PUBLISHING, 2009.
- HTTP://WWW.CS.BERKELEY.EDU/~DEMMEL/CS267/LECTURE20/LECTURE20.HTML
- HTTP://PEOPLE.CS.KULEUVEN.BE/~ALBERT-JAN.YZELMAN/SLIDES/BASEL10.PDF
- HTTP://WWW.STAFF.SCIENCE.UU.NL/~BISSE101/SLIDES/SANFRANCISCO06.PDF
- BISSELING, BO FAGGINGER AUER RH. "ABUSING A HYPERGRAPH PARTITIONER FOR UNWEIGHTED GRAPH PARTITIONING." (2012).
- HTTP://MASTERS.DONNTU.EDU.UA/2006/FVTI/SHEPEL/DISS/INDEXE.HTM
- HTTPS://GFORGE.INRIA.FR/DOCMAN/VIEW.PHP/248/8261/PTSCOTCH_USER6.0.PDF
- HTTP://BMI.OSU.EDU/~UMIT/PATOH/RESULTS.HTML
- PAPA, DAVID A., AND IGOR L. MARKOV. "HYPERGRAPH PARTITIONING AND CLUSTERING."APPROXIMATION ALGORITHMS AND METAHEURISTICS. CRC, BOCA RATAN (2006).
- HTTP://WWW.CS.INDIANA.EDU/CLASSES/B673/NOTES/GRAPHPARTITIONING.PDF
- HTTP://WWW.CS.SANDIA.GOV/CSRI/WORKSHOPS/2008/CSCAPES/PRESENTATIONS/CATALYUREK.PPT
- HTTP://WWW.CS.SANDIA.GOV/~KDDEVIN/PAPERS/ZOLTAN_TUTORIAL_DAGSTUHL09.PDF
- HTTP://WWW.CS.SANDIA.GOV/CSRI/WORKSHOPS/2008/CSCAPES/PRESENTATIONS/BOMAN_CSCAPES08.PPT