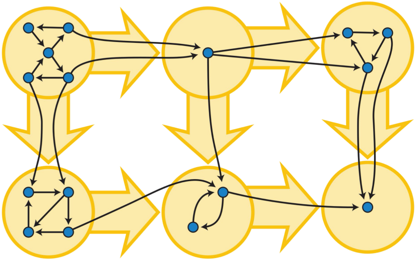
High-Quality
Hierarchical Process Mapping

M. F. Faraj, A. v. d. Grinten, H. Meyerhenke, C. Schulz, J. L. Träff



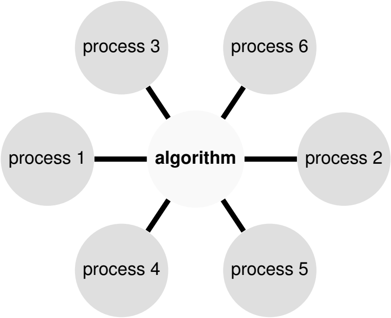
Practical Inspiration / Formulation

Assume parallel algorithm for HPC system

This algorithm = set of processes

Each process

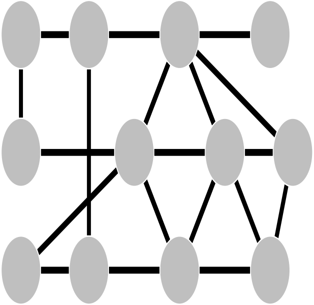
- is atomic
- has computational cost
- can exchange messages



Practical Inspiration / Formulation

A communication graph $G = (V, E)$

$n = |V|$
vertex $v \rightarrow$ process v
vertex weight $c(v) \rightarrow$ duration of v
edge $(v, u) \rightarrow$ message between v, u
edge weight $C_{vu} \rightarrow$ message size



Practical Inspiration / Formulation

HPC system $=$ k PEs $\in$ Topology

Assume:

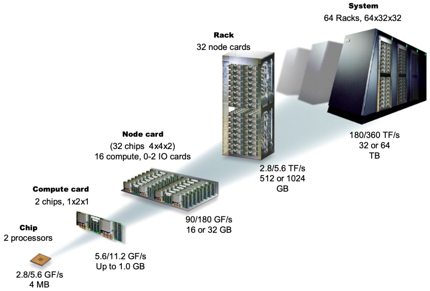
- PEs have same power
- Topology is hierarchical

$D \rightarrow$ distance matrix

$D_{ij} \rightarrow$ distance between PEs i and j

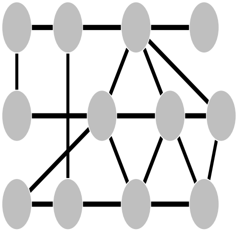
$A = a_1 : a_2 : \ldots : a_\ell \rightarrow$ submodules/level

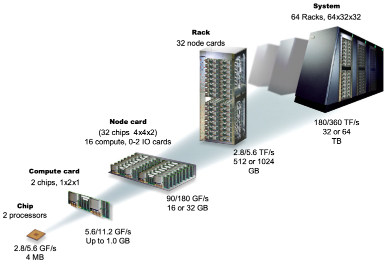
$B = d_1 : d_2 : \ldots : d_\ell \rightarrow$ cost/level



Practical Inspiration / Formulation

General Process Mapping Problem (GPMP)





Obtain map $\pi : V \rightarrow \{\text{PEs}\}$ s.t.

similar workloads

e.g., 3%, 1%, ...

minimized total communication cost

$$J = \sum_{i,j} C_{ij} \cdot D_{\pi(i),\pi(j)}$$

Solution Approaches / Complexity

Most Common Approaches for GPMP

Two-phase Approach (intrinsically not exact)

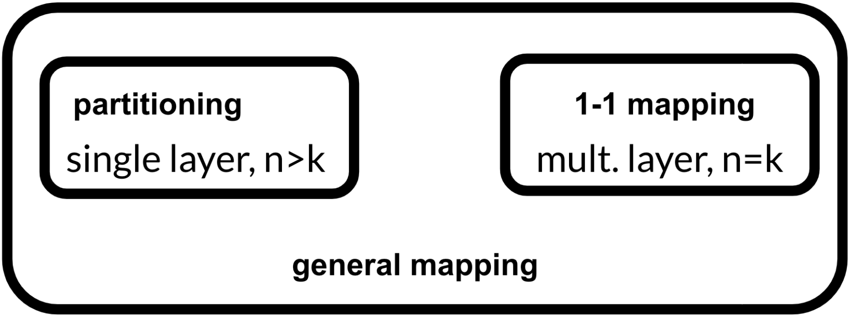
$$\text{graph} \xrightarrow{\text{partitioning}} k \text{ virtual blocks} \xrightarrow{\text{1-1 mapping}} \text{physical map}$$

Integrated Approach

$$\text{graph} \xrightarrow{\text{general mapping}} \text{physical map}$$

Solution Approaches / Complexity

Most Common Approaches for GPMP



All three are NP-hard (Garey e. al,1974; Sahni & Gonzalez, 1976)
Only solvable heuristically for large instances

Literature Remarks

GPMP is a fundamental problem for:

- parallel computing
- distributed computing

But not satisfactorily solved in practice yet.

Literature works include

- formats
- aspects
- approaches
- formulations, etc.

static version $\rightarrow$ most common format

State-of-the-Art

Integrated Approaches

Scotch (Pelegrini, 2008)

Jostle (Walshaw, Cross), software retired/unavailable

Two-phase Approaches

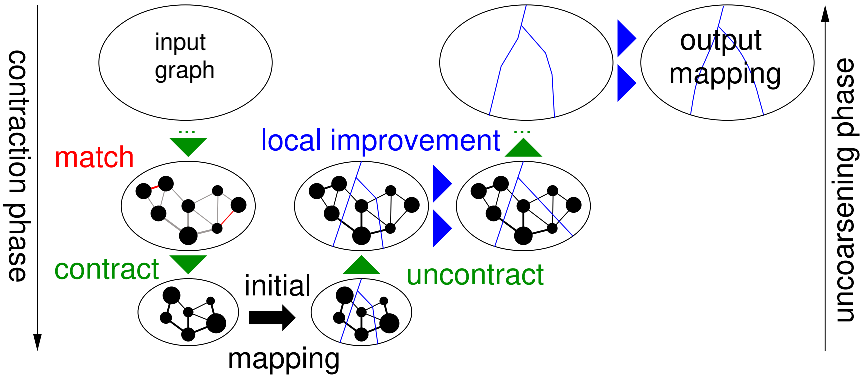
KaHIP (Sanders, Schulz, . . .)

$+$ Mapping (von Kirchbach, Schulz, Träff)

- Identity mapping
- TopDownN
- Müller-Merbach
- Global Multisection

Our Contribution

Multilevel Integrated Mapping



Definitions

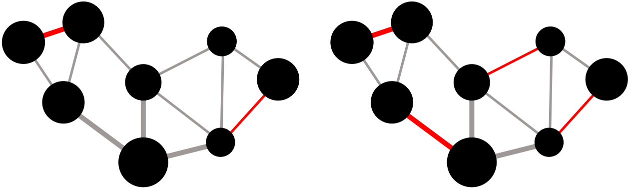
Definition (Matching)

A matching $\mathcal{M} \subseteq E$ is a set of edges not sharing any end point.
I.e. $G = (V, \mathcal{M})$ has maximum degree one.

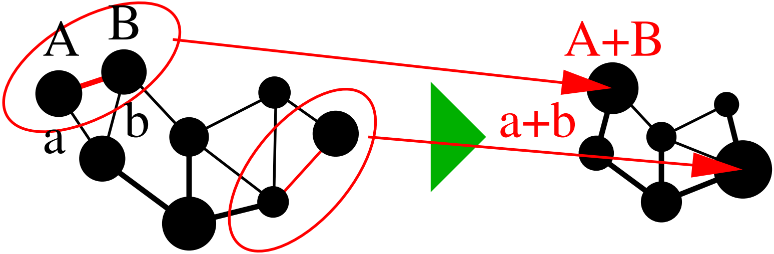
A matching is maximal iff no edge can be added to the matching.

The weight of a matching is $\sum_{e \in \mathcal{M}} \omega(e)$.

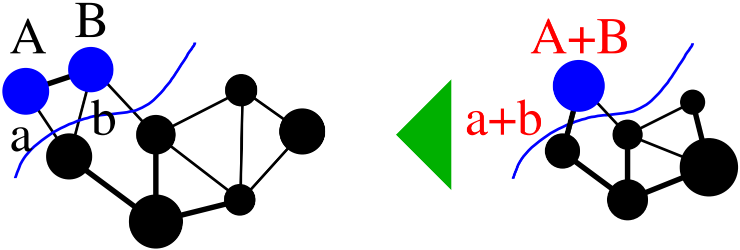
A maximum weight matching has the largest weight possible weight.



Matching-based Coarsening



Matching-based Coarsening



Initial Mapping Algorithm

Our initial mapping is a two-phase process

$$\text{graph} \xrightarrow{\text{partitioning}} k \text{ virtual blocks} \xrightarrow{\text{1-1 mapping}} \text{physical map}$$

Partitioning algorithms:

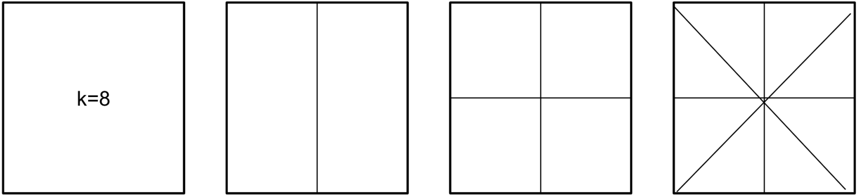
- Recursive Bissections
- Multisections

1-1 mapping algorithms:

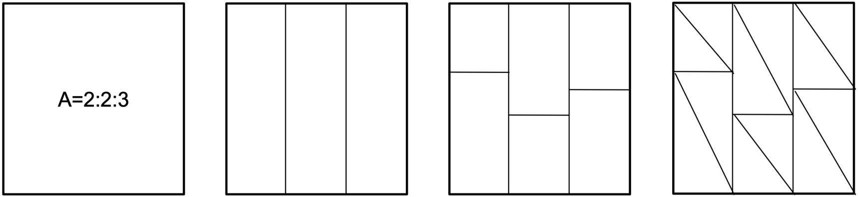
- TopDownN
- Identity

Initial Mapping Algorithm

Recursive Bisections



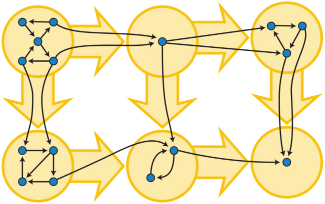
Multisections



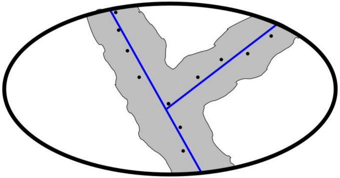
Initial Mapping Algorithm

Config	partitioning		1-1 mapping	
	StdBisec	Multisec	Ident	TopDown
Bsec	yes	no	yes	yes
BsecN	yes	no	yes	yes
MsecT	no	yes	no	yes
MsecTN	no	yes	no	yes
MsecI	no	yes	yes	no
MsecIN	no	yes	yes	no

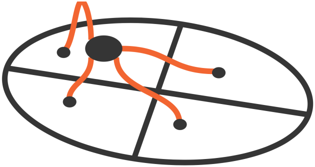
Local Refinement



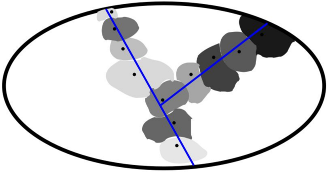
(1)



(2)



(3)



(4)

(1) Quotient graph refinement

(2) K-way FM refinement

(3) Label propagation

(4) Multitry FM refinement

Techniques for Performance

Improve Memory and Runtime Consumption

Delta-gain updates:

Store potential gains
For all nodes throughout execution
It avoids recomputation
At the cost of $O(m+n)$ extra memory

Represent distance matrix D

With degrees of implicity

Implicit Distance Matrix

Full matrix:

$O(1)$ time, $O(k^2)$ space

Division:

$O(\ell)$ divisions + $O(\ell)$ comparisons

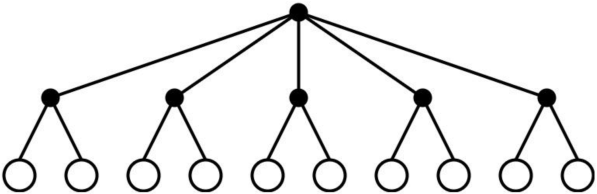
Stored division:

$O(\ell)$ comparisons, $O(k\ell)$ space

Binary notation:

$O(\log \ell)$ time, $O(k)$ space

$$D = \begin{bmatrix} a_{11} & a_{12} & \dots \\ \vdots & \ddots & \\ a_{k1} & & a_k \end{bmatrix}$$



Experimental Evalutation

Parameter Tuning

Graph	n	m
Tuning Graphs		
ecology2	≈1.0M	1 997 996
G3_circuit	≈1.6M	3 037 674
fe_rotor	99 617	662 431
598a	110 971	741 934
del22	≈4.2M	≈12.6M
rgg22	≈4.2M	≈30.4M

$k = 64 \cdot x$, $\forall x \in \{1, \ldots, 128\}$

$A = 4 : 16 : x$

$B = 1 : 10 : 100$

10 repetitions / instance

Geometric mean grouped by (algorithm, k)

Contributions

We tune 4 algorithms

with different purposes

<u>Strong</u>

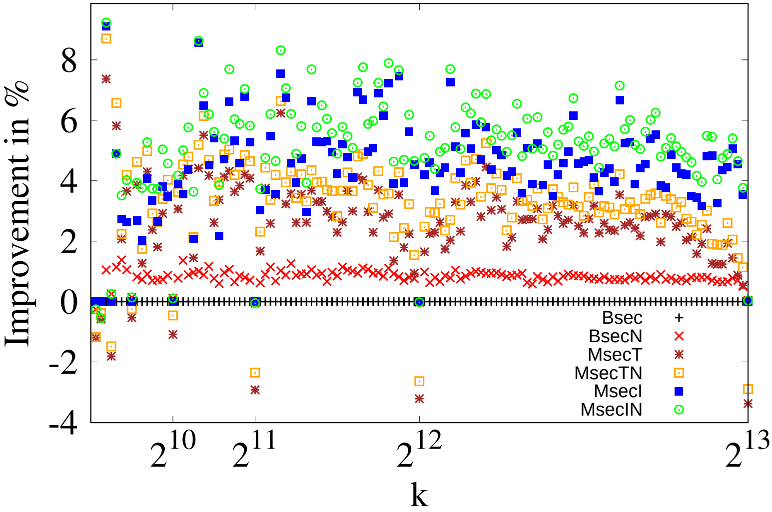
<u>Eco</u>

<u>Fast</u>

<u>Fastest</u>

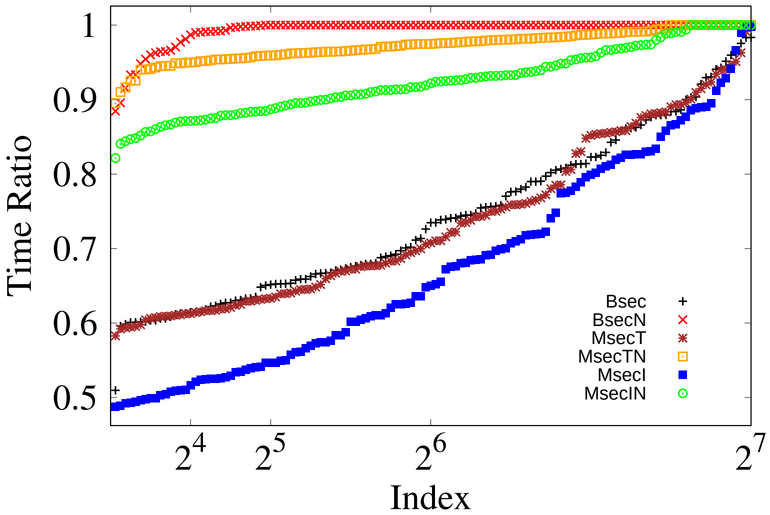
Initial Mapping Algorithm

Solution Quality



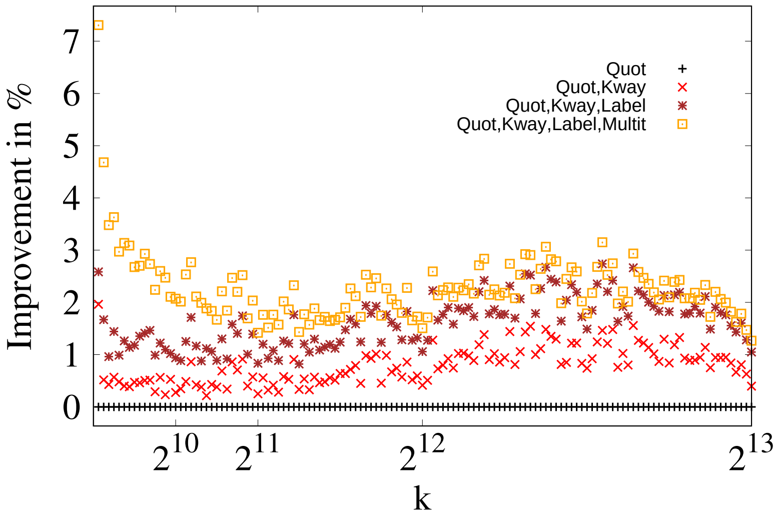
Initial Mapping Algorithm

Performance Profile Time



Local Refinement

Solution Quality



Local Search

Algorithm Configurations

Strong:

MsecIN, quot, kway, label, multitry

Eco:

MsecI, quot, kway, label

Fast:

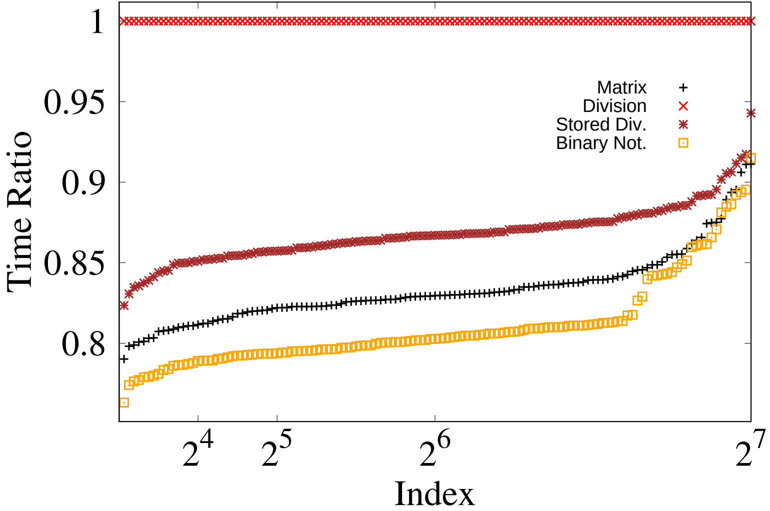
MsecI, label

Fastest:

MsecI

Implicit Distance Matrix

Performance Profile



Comparison with State-of-the-Art

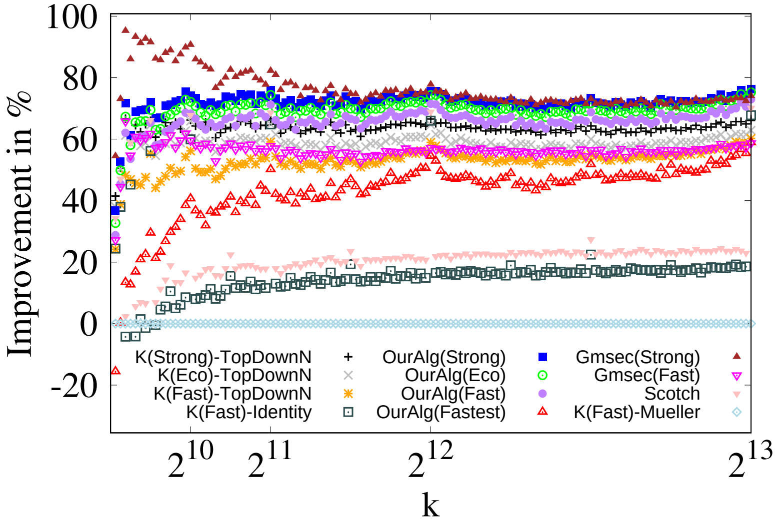
Benchmark Graphs

Graph	*n*	*m*
UF Graphs		
cop20k_A	99 843	1 262 244
2cubes_sphere	101 492	772 886
thermomech_TC	102 158	304 700
cfd2	123 440	1 482 229
boneS01	127 224	3 293 964
Dubcova3	146 689	1 744 980
bmwcra_1	148 770	5 247 616
G2_circuit	150 102	288 286
shipsec5	179 860	4 966 618
cont-300	180 895	448 799

Graph	*n*	*m*
Large Walshaw Graphs		
598a	110 971	741 934
fe_ocean	143 437	409 593
144	144 649	1 074 393
wave	156 317	1 059 331
m14b	214 765	1 679 018
auto	448 695	3 314 611
Large Other Graphs		
del23	≈8.4M	≈25.2M
del24	≈16.7M	≈50.3M
rgg23	≈8.4M	≈63.5M
rgg24	≈16.7M	≈132.6M
deu	≈4.4M	≈5.5M
eur	≈18.0M	≈22.2M
af_shell9	≈504K	≈8.5M
thermal2	≈1.2M	≈3.7M
nlr	≈4.2M	≈12.5M

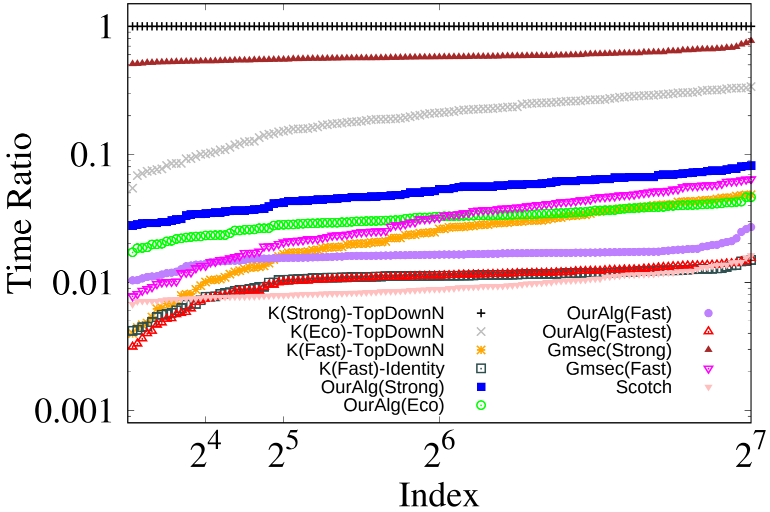
Comparison with State-of-the-Art

Solution Quality



Comparison with State-of-the-Art

Performance Profile Time



Conclusion and Future Work

Summary

- Engineered all components of multilevel integrated mapping
- Experiments $\rightarrow$ our algorithms are new state-of-the-art
  - Strong: as good and order of magnitude faster than previous best,
  - Eco: 2.4% better and 2.8 times faster than prev. 2^{nd} best
  - Fast: better than all but 2 prev. best, faster than all but 2 previous fastest
  - Fastest: as time-efficient as previous fastest, 16% better results on avg
- Soon available as open source in KaHIP and VieM

Future Work

- Parallel and distributed algorithms for GPMP

Funding

- Austrian Science Fund (FWF, project P 31763-N31)
- DFG grant FINCA (ME-3619/3-2, SPP 1736 Algorithms for Big Data)
- German Federal Ministry of Education and Research (BMBF, project WAVE, grant 01—H15004B)

Thank you!

If you need anything, email me: **marcelo.fonseca-faraj@univie.ac.at**