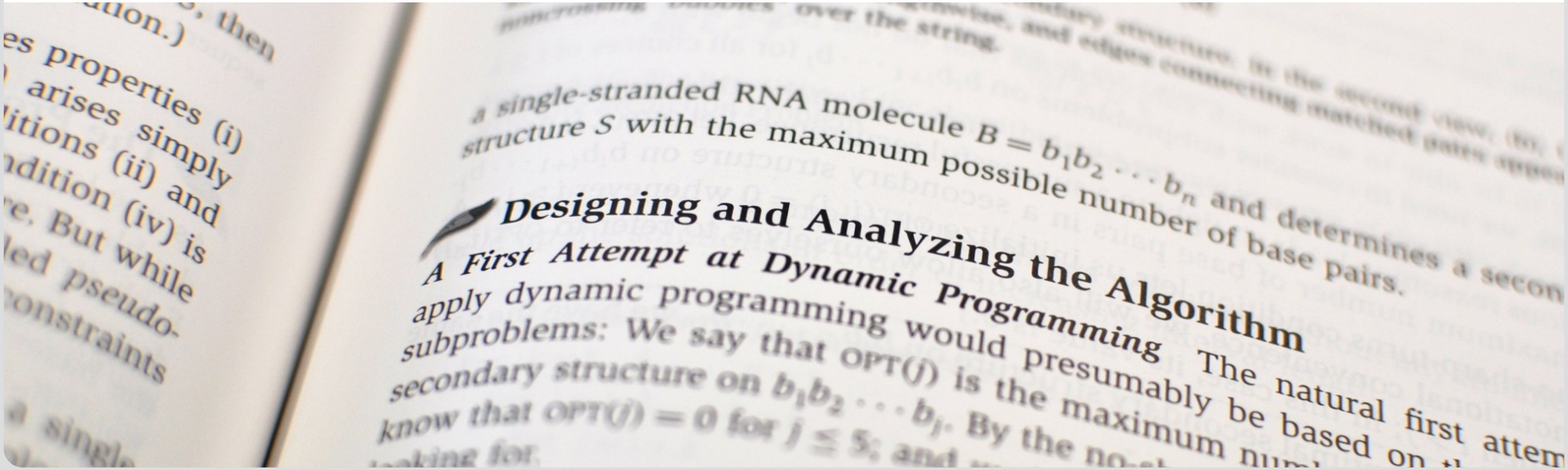
Karlsruher Institut für Technologie

Big Data aus Sicht der Algorithmentechnik

Parallele Analyse- und Optimierungsmethoden für große Graphen

Juniorprof. Dr. Henning Meyerhenke, ITI

Antrittsvorlesung, 28. Januar 2013, Fakultät für Informatik, KIT

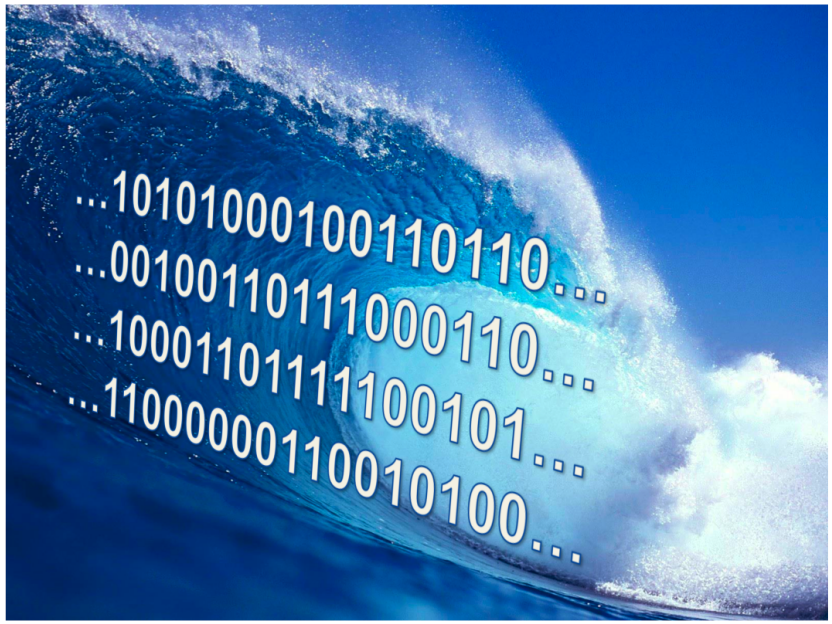


Inhalt

Inhalt

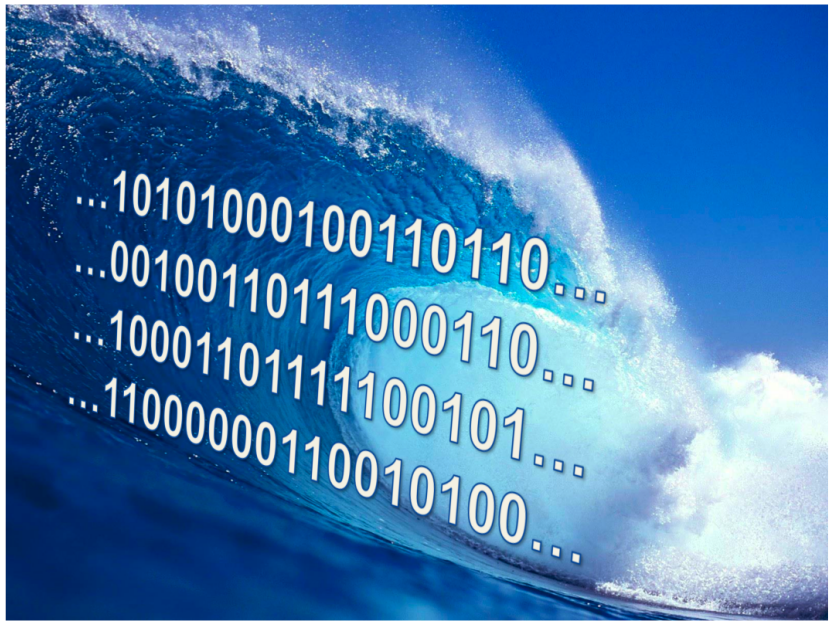
Stetig wachsende Datenflut

- Schlagwort Big Data in aller Munde
- Rasantes Wachstum der Menge von (irregulär strukturierten) Daten:
  - Teilchenbeschleuniger, Teleskope: Terabytes / Tag
  - Facebook: 900M+ Mitglieder (+50% in einem Jahr), 1G+ Aktionen/Tag
  - Web-Graph, Log-Dateien, Smartphone-Aktionen
- Big Data: Nicht nur Graphdaten, aber auch!



Stetig wachsende Datenflut

- Schlagwort Big Data in aller Munde
- Rasantes Wachstum der Menge von (irregulär strukturierten) Daten:
  - Teilchenbeschleuniger, Teleskope: Terabytes / Tag
  - Facebook: 900M+ Mitglieder (+50% in einem Jahr), 1G+ Aktionen/Tag
  - Web-Graph, Log-Dateien, Smartphone-Aktionen
- Big Data: Nicht nur Graphdaten, aber auch!



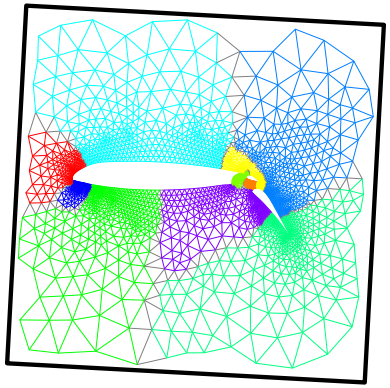
Graphen mischen (fast) überall mit!

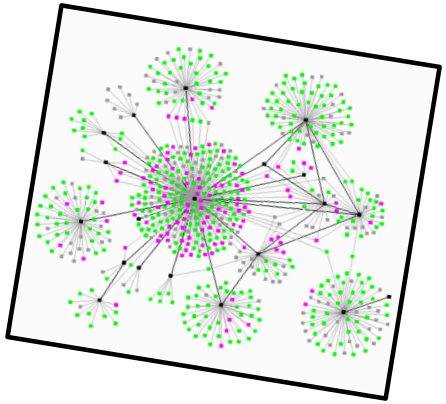
- Energie- und Transportnetzwerke
  - Stromtrassen
  - Verkehrsnetze
- Finanztransaktionen
  - Börsenhandel
  - “Kunden kauften auch...”
- Numerische Simulationen
- Soziale Netzwerke
- Biologische Zusammenhänge
- ...

Graphen mischen (fast) überall mit!

- Energie- und Transportnetzwerke
  - Stromtrassen
  - Verkehrsnetze
- Finanztransaktionen
  - Börsenhandel
  - “Kunden kauften auch...”

- Numerische Simulationen
- Soziale Netzwerke
- Biologische Zusammenhänge
- ...





Paradigmenwechsel

Vom “Zahlenfressen” zu Graphenalgorithmen

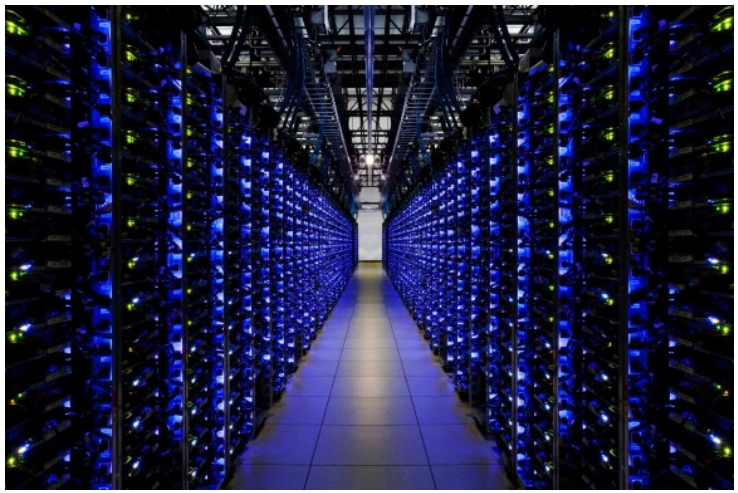
“Traditionelles” HPC

- Numerische Simulationen
- Hohe Lokalität
- Beschränkter Knotengrad
- Rechenintensiv
- Top500-Liste

Titan-Supercomputer am ORNL

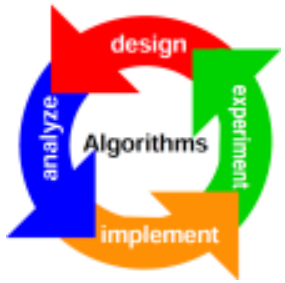
“Neuartiges” HPC

- Soziale Netzwerke, Biologie, ...
- Schlechte Lokalität
- Gradverteilung: Power-Law
- Kommunikationsintensiv
- Graph500-Liste



Google-Serverraum

Herausforderungen
für die Algorithmentechnik



Algorithmentechnik

Zyklus aus

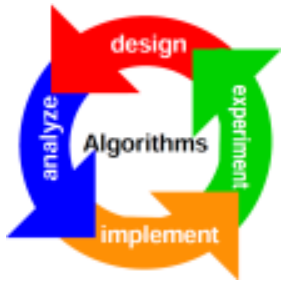
- Entwurf,
- Analyse,
- Implementierung,
- systematische experimentelle Bewertung

von Algorithmen

Neue Herausforderungen

- Größe
- Komplexität
- Dynamik
- Parallelität **und** Heterogenität
- Mobilität
- Wenig zentrale Kontrolle

Herausforderungen
für die Algorithmentechnik



Algorithmentechnik

Zyklus aus

- Entwurf,
- Analyse,
- Implementierung,
- systematische experimentelle Bewertung

von Algorithmen

Neue Herausforderungen

- Größe
- Komplexität
- Dynamik
- Parallelität **und** Heterogenität
- Mobilität
- Wenig zentrale Kontrolle

Inhalt

Kombinatorisches wissenschaftliches Rechnen

Versuch einer Definition

- Algorithmen und Software-Werkzeuge, um kombinatorische Probleme auf High-End-Parallelrechnern zu lösen
- Lösung des kombinatorischen Problems häufig hilfreich beim klassischen wissenschaftlichen Rechnen (Numerik)

Numerische Simulationen

Abbildung: http://www.civilaviation.eu/images/Airbus/A319_F-WWDB.jpg

Numerische Simulationen

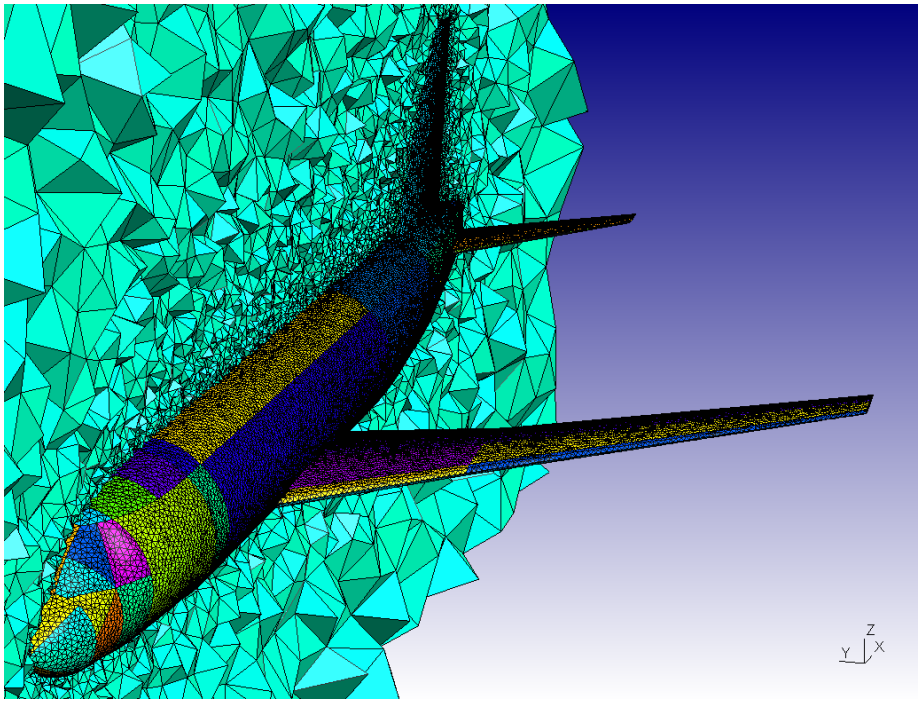


Abbildung: `http://geuz.org/gmsh/gallery/a319_4.png`

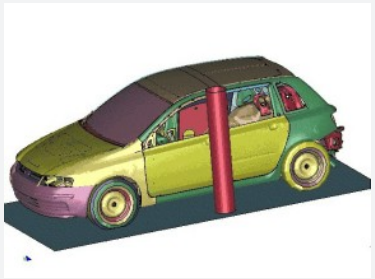
Diskrete Bezüge: Gitternetz-Erzeugung (Meshing), Lastbalancierung

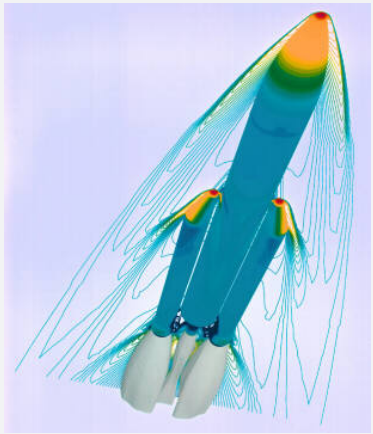
Lastbalancierung

Repartitionierung großer dynamischer Graphen

- **Aufgabe**: Abbildung des Gitternetzes (Diskretisierung) auf die Prozessoren eines Parallelrechners
- **Ziel**: Effiziente parallele Lösung linearer Gleichungssysteme (diskretisierte PDGs)
- Partitioniere Gitternetz derart, dass
  - die Last balanciert wird und
  - die Kommunikation minimiert wird

Gitternetze





Lastbalancierung

Repartitionierung großer dynamischer Graphen

- **Aufgabe**: Abbildung des Gitternetzes (Diskretisierung) auf die Prozessoren eines Parallelrechners
- **Ziel**: Effiziente parallele Lösung linearer Gleichungssysteme (diskretisierte PDGs)
- Partitioniere Gitternetz derart, dass
  - die Last balanciert wird und
  - die Kommunikation minimiert wird

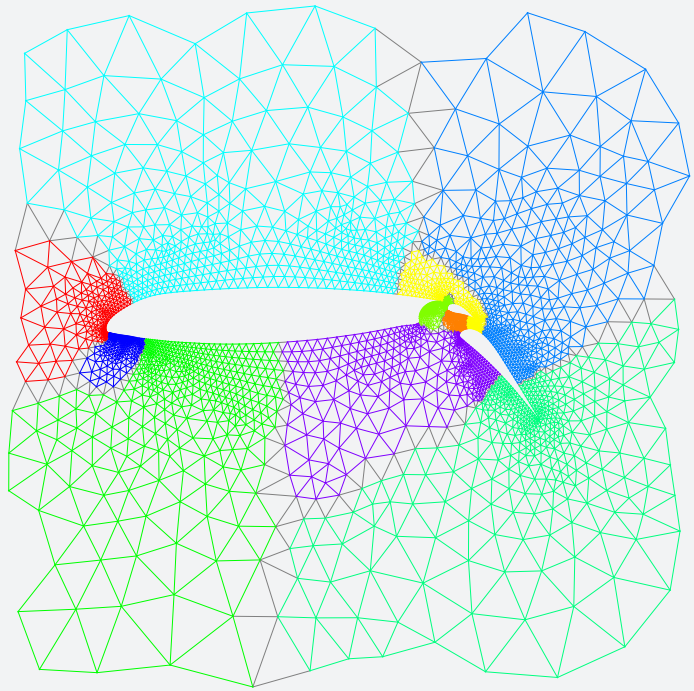
Lastbalancierung

Repartitionierung großer dynamischer Graphen

- **Aufgabe**: Abbildung des Gitternetzes (Diskretisierung) auf die Prozessoren eines Parallelrechners
- **Ziel**: Effiziente parallele Lösung linearer Gleichungssysteme (diskretisierte PDGs)

- Partitioniere Gitternetz derart, dass
  - die Last balanciert wird und
  - die Kommunikation minimiert wird

10-Partitionierung



Graphpartitionierung und -repartitionierung

Statisch: Graphpartitionierungs-Problem

Sei $G = (V, E)$ ein Graph.

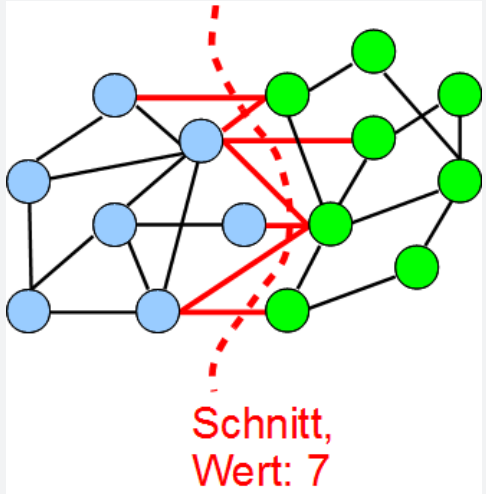
Partitioniere V in $V = \pi_1 \dot\cup \ldots \dot\cup \pi_k$ durch eine Abbildung $\Pi : V \to \{1, \ldots, k\}$ derart, dass

- Π balanciert ist ($|\pi_1| \approx \cdots \approx |\pi_k|$) und
- der Kantenschnitt minimiert wird.

Dynamisch: Repartitionierungs-Problem

Löse das obige Problem mit zusätzlichem Ziel: Minimiere Migrationskosten

Kantenschnitt



Graphpartitionierung und -repartitionierung

Statisch: Graphpartitionierungs-Problem

Sei $G = (V, E)$ ein Graph.

Partitioniere V in $V = \pi_1 \dot\cup \ldots \dot\cup \pi_k$ durch eine Abbildung $\Pi : V \to \{1, \ldots, k\}$ derart, dass

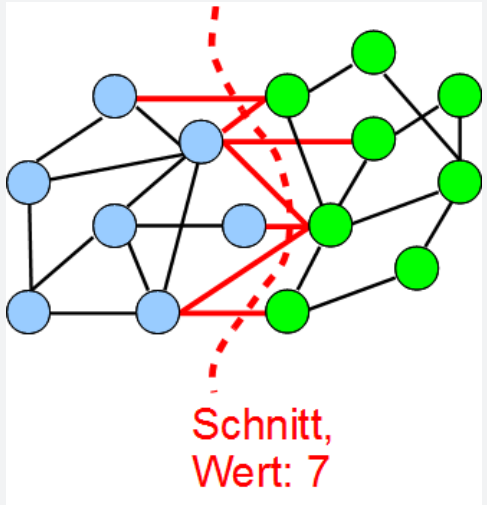
- Π balanciert ist ($|\pi_1| \approx \cdots \approx |\pi_k|$) und
- der Kantenschnitt minimiert wird.

Dynamisch: Repartitionierungs-Problem

Löse das obige Problem mit zusätzlichem Ziel: Minimiere Migrationskosten

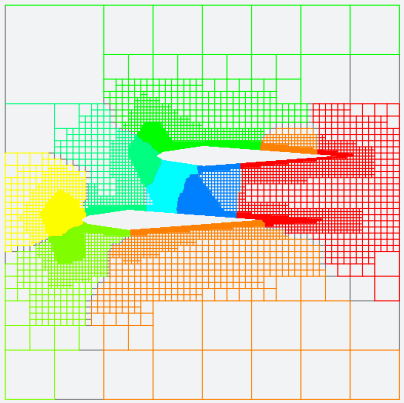
$\mathcal{NP}$-schwer $\rightarrow$ In der Praxis: Heuristiken

Kantenschnitt

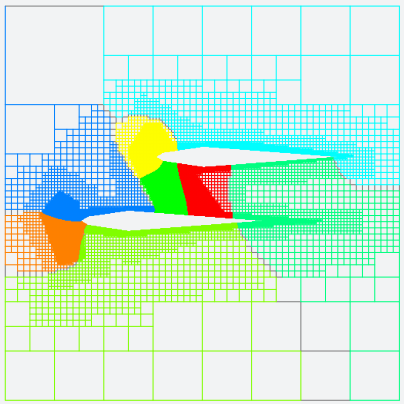


Stand der Technik

MeTiS (KL)



DibaP



- Schnelle Bibliotheken zur (Re-)Partitionierung benutzen sequentielle lokale Suche (etwa Kernighan-Lin (KL) oder MaxFlows) in Multilevel-Prozess
- KL fokussiert zu stark auf den Kantenschnitt $\Rightarrow$ Partitionen sind oft unzusammenhängend und/oder nicht wohlgeformt
- KL ist inhärent sequentiell (es existieren parallele KL-Implementierungen, u. a. ParMeTiS, Parallel Jostle, KaPPa)
- Bester Kantenschnitt für statische Probleme zur Zeit von Sanders und Schulz

Formoptimierende Graphpartitionierung

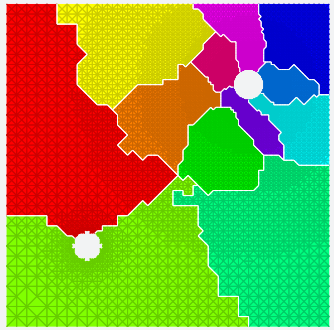
Erste Ansätze: [Diekmann et al., J. ParCo'00]

Idee: Berechne gute Partitionsformen mit kleiner Oberfläche!

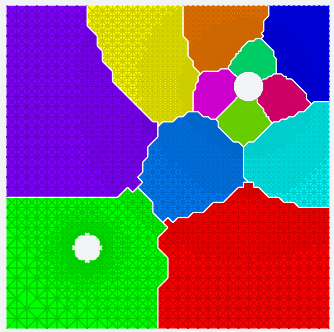
Experimente: Partitionierungen mit guten Formen haben gewünschte Eigenschaften:

- Kleine Partitionsdurchmesser
- Kleine Partitionsränder
- Oft zusammenhängend
- Geringe Migrationskosten bei Repartitionierung

METIS (KL)

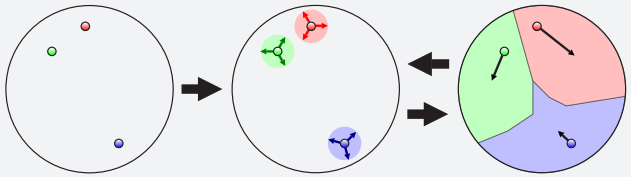


DIBAP



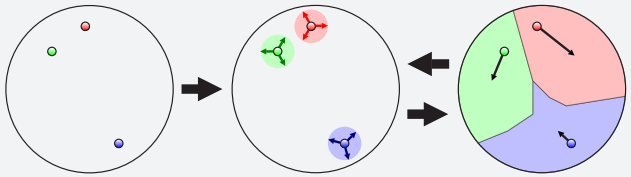
Formoptimierung mit diffusions-basierten Methoden

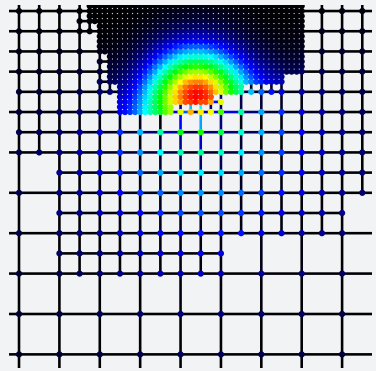
Grundidee: Lloyds *k*-means-Algorithmus



Formoptimierung mit diffusions-basierten Methoden

Grundidee: Lloyds *k*-means-Algorithmus

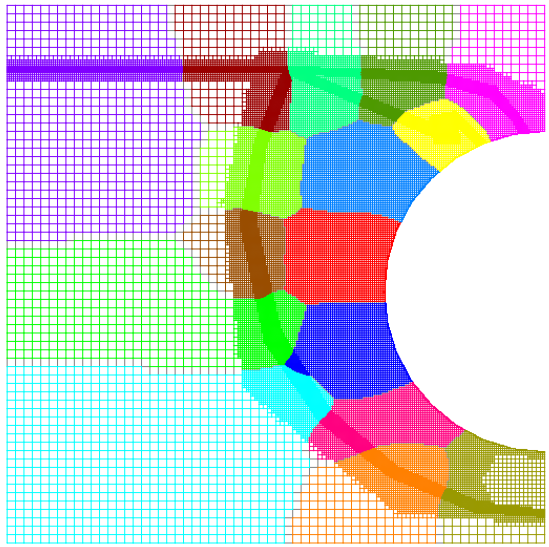




- Ähnlichkeitsmaß soll ausdrücken, wie gut verbunden zwei Knoten/Regionen sind
- Diffusion: Bestreben eines Stoffes, sich gleichmäßig auszubreiten
- Verwandt zu Random Walks, Diffusion breitet sich schneller in dichten Graphregionen aus

Diffusionsbasierte Partitionierung

- Verschiedene diffusionsbasierte Heuristiken entwickelt und implementiert
- Parallelität durch unabhängige Diffusionsprozesse
- Verzahnt in Multilevel-Prozess
- Hohe Lösungsqualität bei akzeptabler Laufzeit, theoretischer Unterbau
- Bei Repartitionierung Stand der Technik bzgl. Qualität (bei Partitionierung Konkurrenz im eigenen Hause)



H. Meyerhenke, B. Monien, T. Sauerwald: A New Diffusion-based Multilevel Algorithm for Computing Graph Partitions. Journal of Parallel and Distributed Computing, 69(9):750-761, 2009. Best Paper Awards and Panel Summary: 22nd International Parallel and Distributed Processing Symposium (IPDPS 2008).

ParMeTiS

Parallel JOSTLE

Zwischenfazit

Beitrag

- Semi-lokaler Prozess Diffusion berechnet gut geformte Partitionen
- Ansatz mit hohem Potential für Parallelität
- Bei Veröffentlichung beste Ergebnisse in Benchmarks

Zwischenfazit

Beitrag

- Semi-lokaler Prozess Diffusion berechnet gut geformte Partitionen
- Ansatz mit hohem Potential für Parallelität
- Bei Veröffentlichung beste Ergebnisse in Benchmarks

Offene Probleme

- Bessere Laufzeit und Skalierbarkeit
- Verbessertes Mapping auf heterogene, nicht-uniforme Architekturen

$\Rightarrow$ DFG-ANR-Projekt zusammen mit INRIA Bordeaux beantragt

$\Rightarrow$ Kombination aus strikt lokalen und semi-lokalen Methoden

Inhalt

Algorithmenanalyse

- Algorithmentechnik beinhaltet Analyse
- Viele erfolgreiche GP-Heuristiken entziehen sich weitgehend einer Analyse

$\Rightarrow$ Lücke zwischen Theorie und Praxis

- **Ziel:** Diese Lücke kleiner machen...

Verwandte Arbeiten

- Microsoft Research: Exakte Bisektionierung in akzeptabler Zeit (bei kleinem Schnitt) mit Branch-and-Cut von Delling et al.
- Lang et al.: Implementierung von Approximationsalgorithmen
- Nachteil bei beiden: Parallelisierung unklar

Diffusionsbasierte Partitionierung

- Arithmetisierung des diffusionsbasierten Partitionierungsalgorithmus
- Anwendung von Ergebnissen der konvexen Optimierung

Theorem

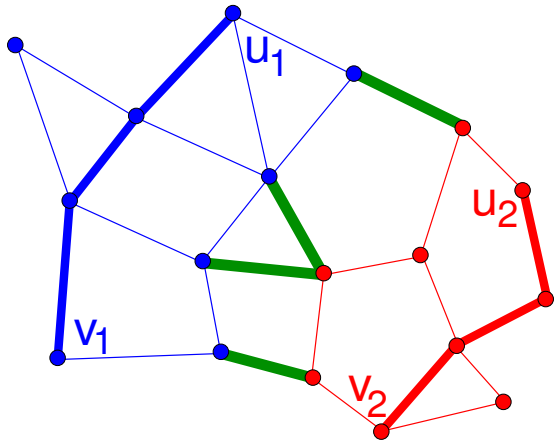
- *Diffusionsbasierte Partitionierung mit* BUBBLE-FOS/C *berechnet das globale Minimum eines konvexen und kontinuierlichen quadratischen Optimierungsproblems.*
- *Dieses Optimierungsproblem ist die relaxierte Variante eines ganzzahligen Optimierungsproblems, das bei passender Wahl der Zentrumsknoten den Kantenschnitt minimiert.*

H. Meyerhenke, T. Sauerwald: Beyond Good Partition Shapes: An Analysis of Diffusive Graph Partitioning. Algorithmica 64(3):329-361, special issue on ISAAC'10.

Konvexe Schnitte in Graphen

Definition (Konvexer Teilgraph, konvexer Schnitt)

- Ein Teilgraph G' ist konvex, wenn für alle $u, v \in V(G')$ gilt:
  Alle kürzesten Wege zwischen u und v liegen vollständig in G'.
- Ein Schnitt (G_1, G_2) ist konvex, falls G_1 *und* G_2 konvex sind.



- Konvexe Schnitte induzieren gute Partitionsformen
- Allgemeine Graphen: Berechnung konvexer Schnitte $\mathcal{NP}$-schwer

Konvexe Schnitte in Graphen

Theorem

Die konvexen Schnitte eines ebenen Graphen können in Zeit $\mathcal{O}(|V|^3)$ berechnet werden.

R. Glantz, H. Meyerhenke: Finding all Convex Cuts of a Plane Graph in Cubic Time.
Accepted at 8th International Conference on Algorithms and Complexity (CIAC'13).

Offene Probleme

- Verwendung von (relaxierten) konvexen Schnitten in der Praxis

$\Rightarrow$ DFG-ANR-Projekt zusammen mit INRIA Bordeaux beantragt

Inhalt

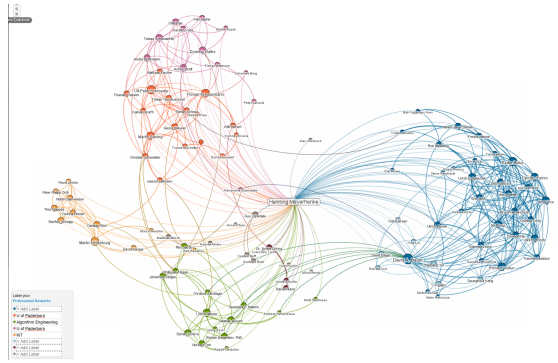
Analyse dynamischer Netzwerke

Mögliche Fragestellungen

- Welche Personen haben hohen Einfluss?
- Wie zerfällt das Netzwerk in “natürliche” Gruppen?
- Wie wird sich das Netzwerk voraussichtlich weiterentwickeln?

Kommerzielle Interessen

- Online-Marketing und -Werbung
- Empfehlungssysteme (“Kunden kauften auch...”)
- Veränderung technischer Infrastruktur



Mein LinkedIn-Netzwerk

Gesundheitspolitische Erwägungen

- Epidemien (Grippe, ...) sind in sozialen Netzwerken nachvollziehbar

⇒ Lassen sich (gesundheits)politische Entwicklungen durch Analyse von sozialen Medien vorhersagen?

Zu bedenken:

- Twitter: 1G+ Tweets/Woche
- Andere Dienste auch relevant
- Datenschutz

⇒ Zielkonflikt zwischen Laufzeit und Genauigkeit

http://thecursedvalley.wordpress.com

Community Detection

“Natürliche” Gruppen eines Netzwerks identifizieren

Komplexität reduzieren: Anwendung teurer Algorithmen nur auf Teile des Netzwerks – **aber welche**?

Clusteranalyse

- Daten desselben Clusters sind sich ähnlich
- Daten verschiedener Cluster sind sich unähnlich

Community Detection

“Natürliche” Gruppen eines Netzwerks identifizieren

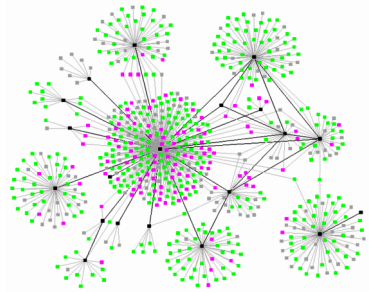
Komplexität reduzieren: Anwendung teurer Algorithmen nur auf Teile des Netzwerks – **aber welche**?

Clusteranalyse

- Daten desselben Clusters sind sich ähnlich
- Daten verschiedener Cluster sind sich unähnlich

Community Detection

- Knoten desselben Clusters sind **stark** miteinander verbunden
- Knoten verschiedener Cluster sind **schwach** miteinander verbunden



http://renardteipelke.blogspot.com

Community Detection

Problem

Eingabe: *Graph* $G = (V, E)$

Ausgabe: *Partition von* V*, die Zielfunktion optimiert*

- Oft: Zielfunktion (ZF) wägt Anteil der internen Kanten, Anteil der externen Kanten und Clustergrößen ab
- Fast alle (interessanten) ZF sind $\mathcal{NP}$-schwer zu optimieren

Community Detection

Problem

Eingabe: *Graph* $G = (V, E)$

Ausgabe: *Partition von* V*, die Zielfunktion optimiert*

- Oft: Zielfunktion (ZF) wägt Anteil der internen Kanten, Anteil der externen Kanten und Clustergrößen ab
- Fast alle (interessanten) ZF sind $\mathcal{NP}$-schwer zu optimieren

Weitere Anwendungen

- Ähnliche Objekte finden (Gene, Produkte, Personen, ...)
- Verteiltes Rechnen, Speichern
- Visualisierung

Parallele Heuristiken

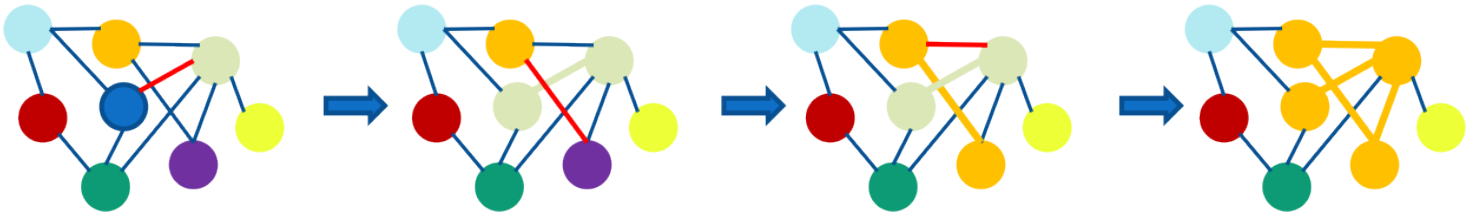
Community Detection

- 10th DIMACS Implementation Challenge: Graphpartitionierung und Graphclustering
- Nur zwei parallele Clustering-Verfahren im Wettbewerb
- Beide agglomerativ, Unterschiede in den Details

Parallele Heuristiken

Community Detection

- 10th DIMACS Implementation Challenge: Graphpartitionierung und Graphclustering
- Nur zwei parallele Clustering-Verfahren im Wettbewerb
- Beide agglomerativ, Unterschiede in den Details



- Statt Kontraktion einzelner Kanten: Parallele Kontraktion von Kanten eines möglichst schweren Matchings

E.J. Riedy, H. Meyerhenke. D. Ediger, D.A. Bader: Parallel Community Detection for Massive Graphs. To appear in Graph Partitioning and Graph Clustering. Proc. 10th DIMACS Implementation Challenge. AMS Contemporary Mathematics, 2013.

Cray XMT/XMT2

HPC-Architektur für Graphenprobleme

- Latenz wird durch massives Multithreading versteckt
  - Hardware-Unterstützung für 128 Threads pro Prozessor
  - Kein Daten-Cache!
  - Kontext-Wechsel in nur einem Zyklus
- Feingranulare Synchronisation auf Speicherwortebene
- 1 TB globaler gemeinsamer Speicher

Abbildung: `http://www.cray.com/`

Cray XMT/XMT2

HPC-Architektur für Graphenprobleme

- Latenz wird durch massives Multithreading versteckt
  - Hardware-Unterstützung für 128 Threads pro Prozessor
  - Kein Daten-Cache!
  - Kontext-Wechsel in nur einem Zyklus
- Feingranulare Synchronisation auf Speicherwortebene
- 1 TB globaler gemeinsamer Speicher

Abbildung: `http://www.cray.com/`

- **Unsere Experimente:** Vergleich
  - sequentiell vs. parallel und
  - XMT vs. 40-Kern-Intel-Workstation

Resultate

Community Detection

- Qualitätsvergleich sequentiell vs. parallel: Eher wenig Communitys parallel (gut!), aber sequentiell ist Zielfunktionswert besser

Laufzeit

- Cray XMT profitiert von großem Hauptspeicher
- Einziger DIMACS-Löser, der größte Instanzen bearbeiten konnte
- Kleinere Instanzen: Höhere Performance auf 40-Kern-Workstation
- **Erreicht:** Laufzeit gesenkt von Kaffeepause auf Email-Check

Resultate

Community Detection

- Qualitätsvergleich sequentiell vs. parallel: Eher wenig Communitys parallel (gut!), aber sequentiell ist Zielfunktionswert besser

Laufzeit

- Cray XMT profitiert von großem Hauptspeicher
- Einziger DIMACS-Löser, der größte Instanzen bearbeiten konnte
- Kleinere Instanzen: Höhere Performance auf 40-Kern-Workstation
- **Erreicht:** Laufzeit gesenkt von Kaffeepause auf Email-Check

Aktuell: MWK-Projekt

Ziel: Geringe Laufzeit **und** höhere Qualität
Methoden: Bessere Matchings für mehr Parallelität, Ensemble-Konzept

Inhalt

Anwendung: Sequenzassemblierung

- Aufgabe: Rekonstruktion eines Genoms
- Durchbruch für das Verständnis von biologischen Mechanismen
- Prognose: Flächendeckender Einsatz im Klinikalltag bei schweren Krankheiten

Anwendung: Sequenzassemblierung

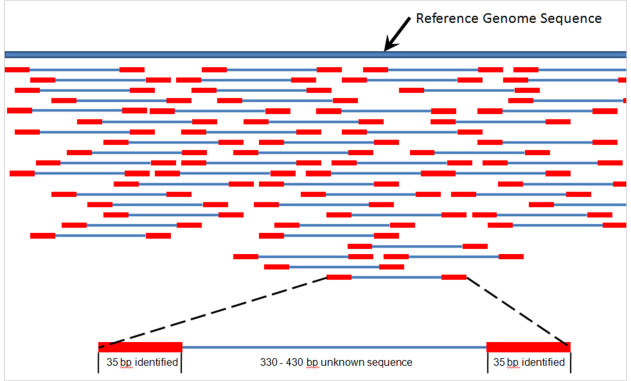
- Aufgabe: Rekonstruktion eines Genoms
- Durchbruch für das Verständnis von biologischen Mechanismen
- Prognose: Flächendeckender Einsatz im Klinikalltag bei schweren Krankheiten

Biologische Motivation

Schnellere Analyse von Krankheiten,
Entwicklung von Medikamenten und Impfstoffen

Im Labor:
Sequenzierungsmaschinen liefern sehr viele kurze Genom-Abschnitte

`http://www.mn.uio.no/ifi/studier/masteroppgaver/bio/benchmarking-system.html`



Problemstellung

Idealisierte Problemstellung für die Informatik

Gegeben: Menge S von Strings (kurzer Länge)

Gesucht: Finde kürzesten String $\overline{S}$, der alle Strings aus S enthält!

Anschaulich: Rekonstruiere Genom durch Lösen eines “Puzzles”, d. h. finde die korrekten Überlappungen und die richtige Reihenfolge der DNA-Abschnitte!

Ziel in der Praxis: Finde möglichst lange kontinuierliche Abschnitte (und das möglichst schnell)!

Abbildung: http://www.cdc.gov/genomics/update/file/images/spotlight/2010-05_DNA.jpg

Herausforderungen und Stand der Technik zu Beginn des Projekts

Herausforderungen

- Milliarden Basenpaare gängige Eingabegröße
- Selbst unter idealen Bedingungen $\mathcal{NP}$-schweres Problem
- Sequenzierungsfehler, Wiederholungen im Genom

Herausforderungen und Stand der Technik zu Beginn des Projekts

Herausforderungen

- Milliarden Basenpaare gängige Eingabegröße
- Selbst unter idealen Bedingungen $\mathcal{NP}$-schweres Problem
- Sequenzierungsfehler, Wiederholungen im Genom

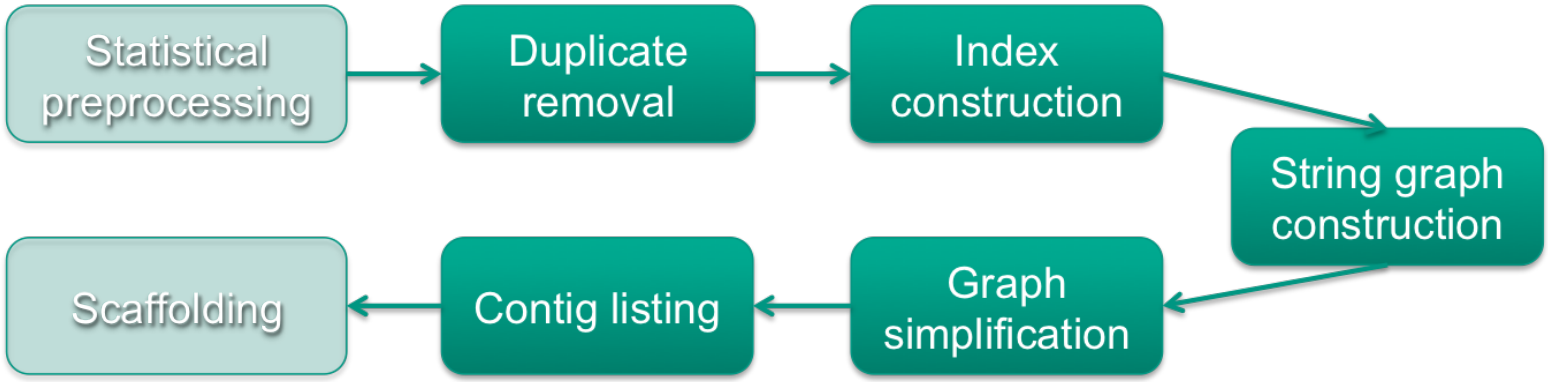
Stand der Technik

- Wenige parallele Codes
- Fast alle davon benutzen zur Modellierung der Überlappungen de-Bruijn-Graphen (DBG)
- DBG-Modell nur günstig für Eingaben mit kurzen Reads

Aufbau der Assembly-Pipeline

Beitrag: Parallelisierung der inneren Abschnitte

Assemblierungs-Pipeline:



Unser Ansatz, basierend auf OpenMP

- Parallel konstruierter komprimierter Index
- Parallele Konstruktion eines String-Graphen zur Modellierung von Überlappungen
- Parallele Traversierungsalgorithmen suchen lange Touren

Ergebnisse

- Experimente: Unsere Software PASQUAL (Parallel Sequence Assembler) bietet besten Kompromiss aus
  - Laufzeit
  - Speicherverbrauch und
  - Qualität
- Eine Größenordnung schneller als der Stand der Technik zu Projektbeginn (mehr im Vergleich zu sequentiellen Verfahren)
- Offenes Problem (nicht nur für PASQUAL): Verbesserung der Lösungsqualität für reale Datensätze

X. Liu, P. R. Pande, H. Meyerhenke, D.A. Bader:
PASQUAL: Parallel Techniques for Next Generation Genome Sequence Assembly.
Accepted for publication in IEEE Transactions on Parallel and Distributed Systems, June 2012.

Inhalt

Lehre und YIN

Meine Lehrveranstaltungen am KIT

- V, M: Algorithmische Methoden zur Netzwerkanalyse
- V, M: Graphenalgorithmen und lineare Algebra Hand in Hand
- S, M: Algorithmentechnik
- V, M: Kombinatorische Optimierung)
- PS, B: Graphpartitionierung (mit P. Sanders)
- **SS 2013:** V, B: **Algorithmische Methoden für schwierige Optimierungsprobleme**

Lehre und YIN

Meine Lehrveranstaltungen am KIT

- V, M: Algorithmische Methoden zur Netzwerkanalyse
- V, M: Graphenalgorithmen und lineare Algebra Hand in Hand
- S, M: Algorithmentechnik
- V, M: Kombinatorische Optimierung)
- PS, B: Graphpartitionierung (mit P. Sanders)
- **SS 2013:** V, B: **Algorithmische Methoden für schwierige Optimierungsprobleme**

YIN: Young Investigator Network

- Zusammenschluss der KIT-Nachwuchsgruppen
- Meine Funktion: Ausschuss für Alumni und PR, Planung YIN-Day

Schlussbemerkungen

Danksagungen

- **Co-Autoren:**
  D. Ajwani, D.A. Bader, D. Ediger, J. Gehweiler, R. Glantz, R. Görke, X. Liu, T. Mattson, B. Monien, P.R. Pande, E.J. Riedy, P. Sanders, T. Sauerwald, S. Schamberger, U.-P. Schroeder, A. Schumm, C. Staudt, C. Schulz, D. Wagner
- **Geldgeber:**
  MWK, DFG, DARPA, PNNL, EU

Schlussbemerkungen

Ihnen vielen Dank fürs Kommen!

Literaturverweise I

H. Meyerhenke.

Beyond good shapes: Diffusion-based graph partitioning is relaxed cut optimization.

In *Proc. 21st International Symposium on Algorithms and Computation (ISAAC)*. Springer, 2010.

J. Gehweiler, H. Meyerhenke.

A Distributed Diffusive Heuristic for Clustering a Virtual P2P Supercomputer.

In *Proc. High Performance Grid Computing Workshop (HPGC'10), in conjunction with 24th Intl. Parallel and Distributed Processing Symposium (IPDPS'10)*. IEEE, 2010.

Literaturverweise II

H. Meyerhenke, B. Monien, and S. Schamberger.
Graph partitioning and disturbed diffusion.
Parallel Computing, 35(10-11):544–569, 2009.

H. Meyerhenke, B. Monien, and T. Sauerwald.
A new diffusion-based multilevel algorithm for computing graph partitions.
Journal of Parallel and Distributed Computing, 69(9):750–761, 2009.
Best Paper Awards and Panel Summary: IPDPS 2008.

Literaturverweise III

H. Meyerhenke.
Dynamic Load Balancing for Parallel Numerical Simulations Based on Repartitioning with Disturbed Diffusion.
In *Proc. Internatl. Conference on Parallel and Distributed Systems (ICPADS'09)*. IEEE, 2009.

R. Andersen, F. R. K. Chung, and K. J. Lang.
Local graph partitioning using pagerank vectors.
In *Proceedings of the 47th Annual Symposium on Foundaions of Computer Science (FOCS)*, pages 475–486, 2006.

Literaturverweise IV

M. Meila and J. Shi.
A random walks view of spectral segmentation.
In *8th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.

L. Grady, Eric L. Schwartz,
Isoperimetric graph partitioning for image segmentation.
IEEE Trans. on Pat. Anal. and Mach. Int, 28:469–475, 2006.

R. Diekmann, R. Preis, F. Schlimbach, and C. Walshaw.
Shape-optimized mesh partitioning and load balancing for parallel adaptive FEM.
J. Parallel Computing, 26:1555–1581, 2000.

Literaturverweise V

B. Hendrickson.
“Graph Partitioning and Parallel Solvers: Has the Emperor No Clothes?”.
In: *Proceedings of Irregular’98*, pp. 218–225, Springer-Verlag, 1998.

C. Walshaw, M. Cross, and M. G. Everett.
“A Localised Algorithm for Optimising Unstructured Mesh Partitions”.
Intl. J. Supercomputer Appl., Vol. 9, No. 4, pp. 280–295, 1995.