



FAKULTÄT FÜR  
INFORMATIK

# Seminar: Optimierungs- und Modellierungstechniken für Datenbanken

## Spezielle DBMS zur Stromverbrauchsminimierung

# Gliederung

- Einleitung
- Motivation
- Vergleich verschiedener Algorithmen
- Optimierte Algorithmenutzung
- Ein Ansatz zur Evaluierung: JouleSort
- Zusammenfassung
- Ausblick
- Quellen

# Einleitung

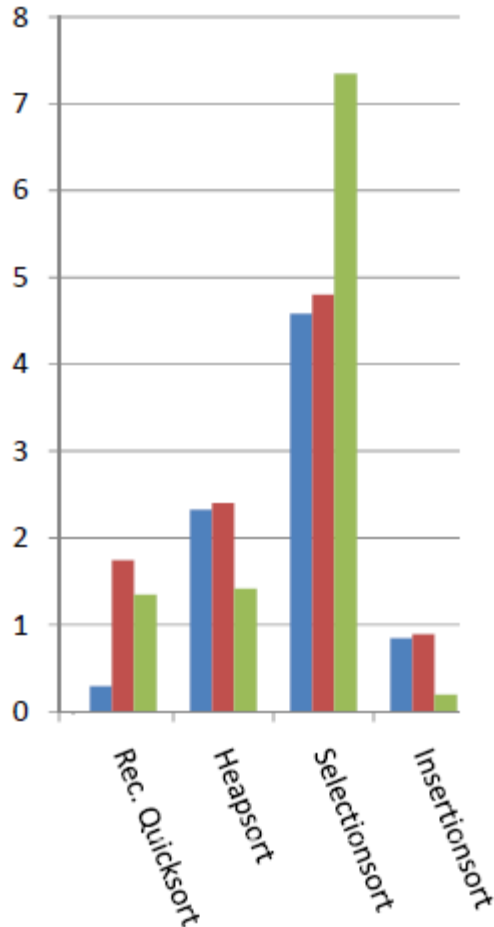
- Verschiedene Ansätze zur Stromverbrauchsminimierung
- Hardwareebene
  - ▶ Beispielsweise innovative Speichermedien
- Softwareebene
  - ▶ Optimierung von standardmäßigen Datenbankfunktionalitäten bezüglich der Energieeffizienz
  - ▶ Bisher: effiziente Algorithmen  $\rightarrow$  max. Rechengeschwindigkeit, min. Komplexität
  - ▶ Neue Messgröße: Energieverbrauch pro Algorithmen durchlauf

# Motivation

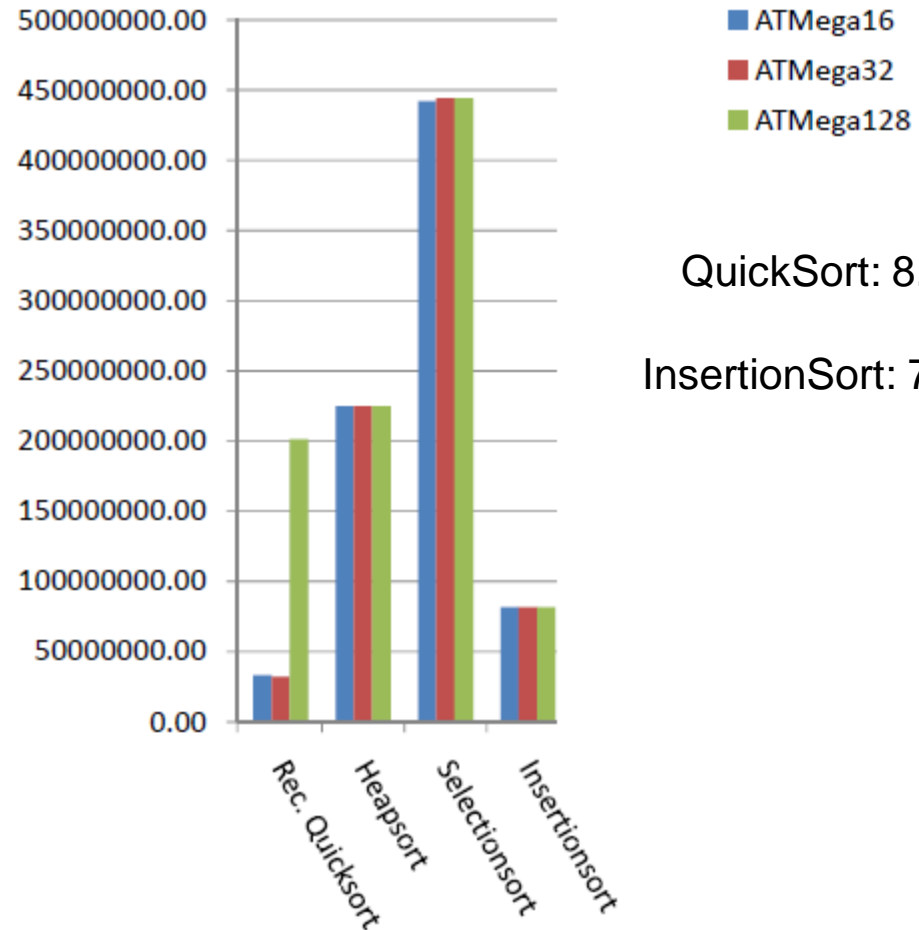
- DBMS hauptsächlich für Desktop-PCs oder leistungsstarke Server optimiert
- Kosten für Energie und Kühlung steigen mit der Leistung
- Steigende Anzahl von mobilen und eingebetteten Systemen (PDAs, Laptops, Sensornetzwerke...)
- Energieeffiziente Optimierung hauptsächlich auf Hardwareebene
- Software-/algorithmusbasierte Optimierung unabhängiger

- Sortierverfahren
  - ▶ Duplikateliminierung, Gruppierung, Joins...
  - ▶ QuickSort, InsertionSort, HashSort, SelectionSort...
- Joinverfahren
  - ▶ Nested-Loop-Join (NLJ), Hash-Join (HJ), Sort-Merge-Join (SMJ)
- Test verschiedener Algorithmen bezüglich 3 Kriterien
  - ▶ CPU Zyklen
  - ▶ Ausführungszeit
  - ▶ Energieverbrauch

# Testergebnisse Sortierverfahren



Energieverbrauch in Joule



CPU Zyklen

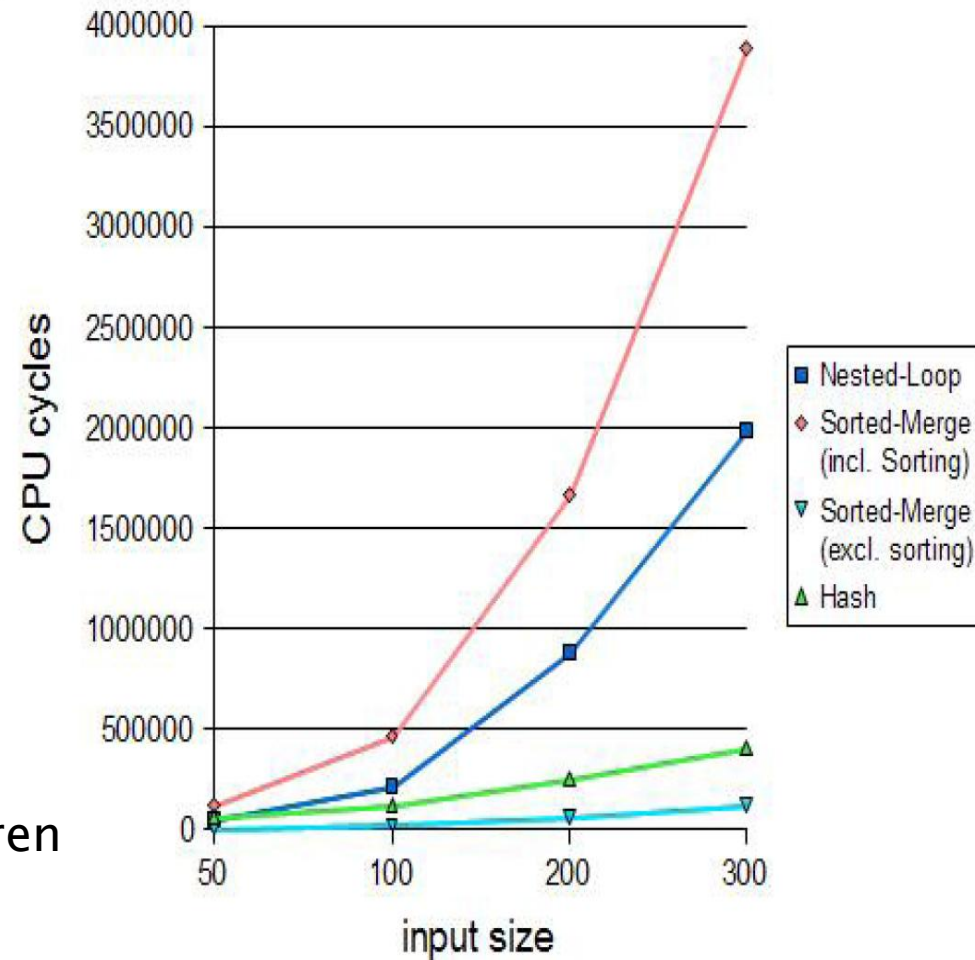
QuickSort: 8.8 ms

InsertionSort: 71.3 ms

[1]“Energy Consumptions of Sorting and Join Algorithms“

# Joinverfahren

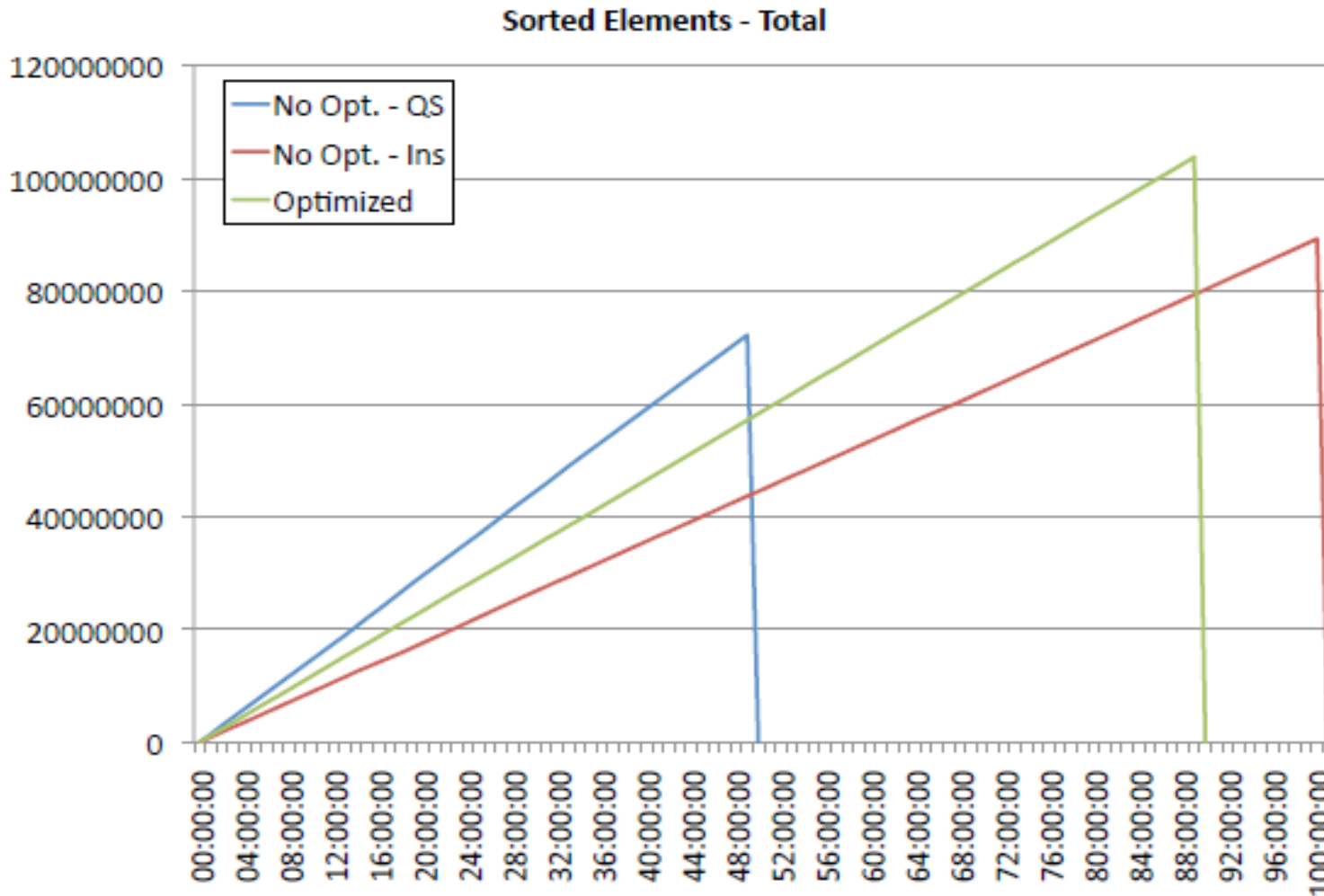
- SMJ und NLJ haben so gut wie keine I/O Operationen
- HJ hingegen benötigt zusätzlichen Speicher
- NLJ hat hohen Energieverbrauch
- SMJ abhängig vom Sortierverfahren



[1]“Energy Consumptions of Sorting and Join Algorithms“

- Energieeffiziente Algorithmen sind langsam
- Schnelle Algorithmen verbrauchen zu viel Energie und verringern dadurch die Betriebszeit
- ▶ Dynamischer Ansatz
- „Mischen“ beider Verfahren
  - ▶ Zusätzliche Komponente, die entscheidet welches Verfahren angewandt werden soll
  - ▶ Liegt die Differenz zwischen Energieverbrauch des „schnellsten“ und des „stromsparenden“ Algorithmus unter einem vordefinierten Schwellenwert, wird der Schnellere gewählt

# Optimierte Algorithmenutzung (2)



[1]“Energy Consumptions of Sorting and Join Algorithms“

- Benchmark zur Bewertung verschiedener Sortierverfahren bezüglich Energieeffizienz, Performance und Kosten
- Bisherige Benchmarks zur Energieeffizienz:
  - ▶ Meist nur eine Metrik betrachtet
  - ▶ Schwerpunkt liegt auf der HW-Ebene
  - ▶ I/O Operationen nur wenig beachtet
  - ▶ Meist unvollständig

## Kriterien:

- Messung erfolgt während „Peak Use“
  - kein idle-state-Benchmark
- Balancierte Nutzung der Systemkomponenten
  - CPU, RAM, I/O und Festspeicher
- Messung auf verschiedenen Plattformen
  - Portabilität gewährleisten
- Repräsentativ

## Metrik:

- Feste Eingangsgröße (MB, GB, TB)
- Minimierung des Verbrauchs
- Maximierung der Rechengeschwindigkeit

# Warum Sortierverfahren?

- Alle Kriterien werden erfüllt
  - ▶ „Peak Use“
  - ▶ Portabilität
  - ▶ Repräsentativ
  - ▶ Nutzung aller Systemkomponenten
  
- Andere Benchmarks für Sortierverfahren
  - ▶ PennySort
  - ▶ MinuteSort
  - ▶ TerabyteSort
  
  - ▶ Jedoch nicht auf Energiekostensenkung ausgelegt
  - ▶ JouleSort als Erweiterung bisheriger Benchmarks

- Ablauf:
  - ▶ Sortieren einer festen Anzahl von Datensätzen der Größe 100byte (Key 10byte)
  - ▶ Datensätze zufällig gemischt
  - ▶ Beginn durch Input eines externen Speichers
  - ▶ Ende durch Output auf externen Speicher
  - ▶ Drei Kategorien: 10/100/1000GB Datensätze
  - ▶ SortedRecs/Joule als Messgröße
  
- Regelmäßige Wettbewerbe
  - ▶ Aufteilung in zwei Kategorien: Daytona (COTS) und Indy
  - ▶ „Gewinner“ war 3.5x effizienter als PennySort (Stand 2007)
  
- Kritik: zu hohe Last auf I/O System

- Gute Performance eines Algorithmus geht mit erhöhtem Energieverbrauch einher
- Dynamische Auswahl des „besten“ Algorithmus im DBMS erforderlich
- Durch optimierte Algorithmenutzung annähernd gleiche Energieeffizienz wie InsertionSort, jedoch schneller
- JouleSort als Ansatz zu einem balancierten, plattformunabhängigen Benchmark
- „anwendungsnahe“ Tests

- Weitreichende Forschungen und Tests im Bereich der Energieeffizienzsteigerung
- Verschiedene Ansätze eine „Energie Management Komponente“ in DBMS zu realisieren
- JouleSort als zukünftiges Mittel zur Evaluierung verschiedener Sortierverfahren

# Vielen Dank für Ihre Aufmerksamkeit!

## Quellen

[1][H. Höpfner, C. Bunse: „Towards an Energy Aware DBMS – Energy Consumptions of Sorting and Join Algorithms“, 2009]

[2][R.J. Bestgen, W.Hu, S. Kethireddy, A.P.Passe, U.Thiemann: „Organizing Databases für Energy Efficiency“, 2009 ]

[3][C. Bunse, H.Höpfner, S. Roychoudhury, E. Mansour: „Choosing the ‚best‘ sorting algorithm for optimal energy consumption“, 2009]

[4][S. Rivoire, M. A. Shah, P. Ranganathan, C. Kozyrakis: „JouleSort: A Balanced Energy–Efficiency Benchmark“, 2007]