Type 2 Slowly Changing Dimensions:

A Case Study Using the Co>Operating System

Craig Stanfill
Ab Initio Software
DOLAP'12, Maui

Overview

The Co>Operating System

- Ab Initio's parallel computing framework
- Based on partitioned dataflow
- Graphic programming

A little about what graphs really look like

- Primary computation
- Secondary computations: "Salad Dressing"
- Five solutions to "Type 2 Slowly Changing Dimensions"

Performance:

- Scalability
- Insights into the important tradeoffs for optimization

The Co>Operating System

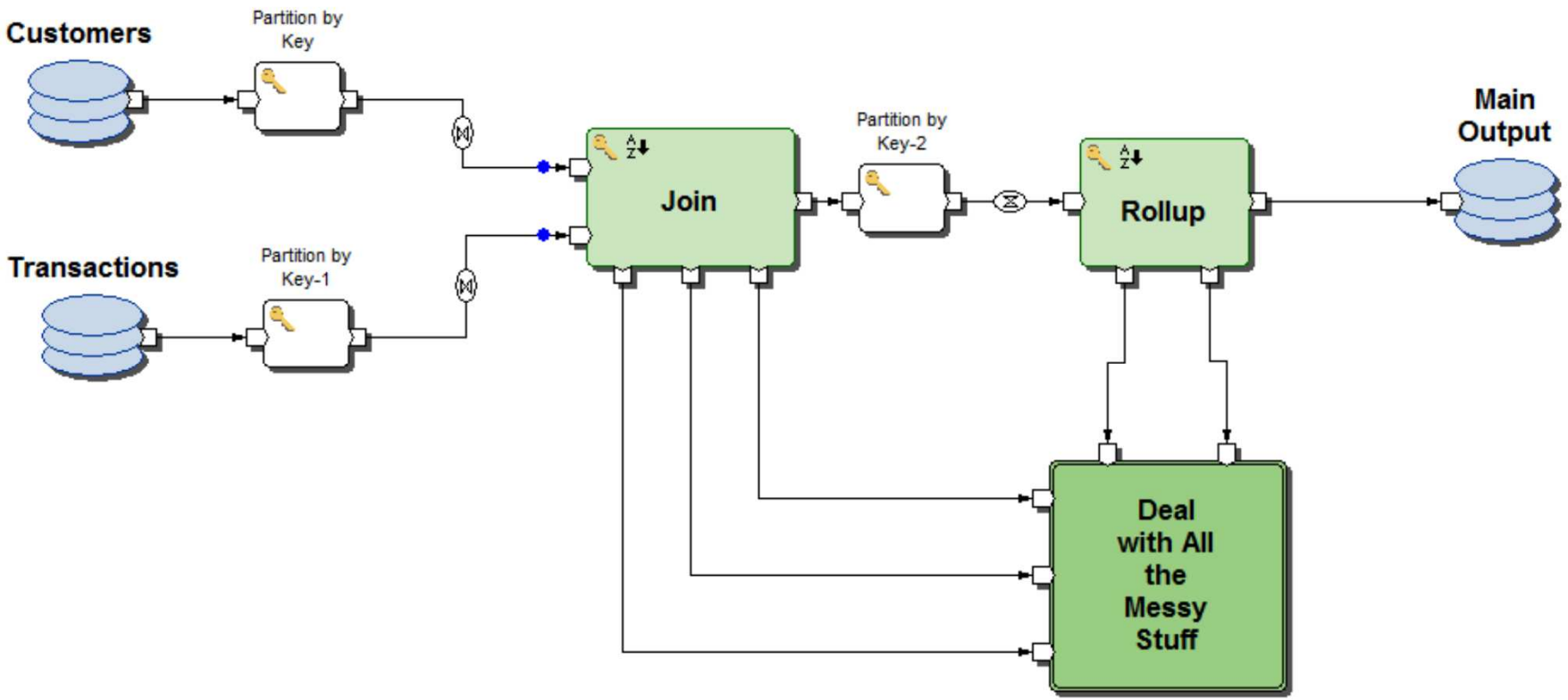
Parallel framework for Enterprise Computing

- Widely used for ETL, Data Warehousing
- Widely used for Mission Critical Realtime Apps
- Stock Exchanges, Telecommunications, Credit Card Processing
- Batch, Streaming, Service, Transactional Modes

Compared with MapReduce:

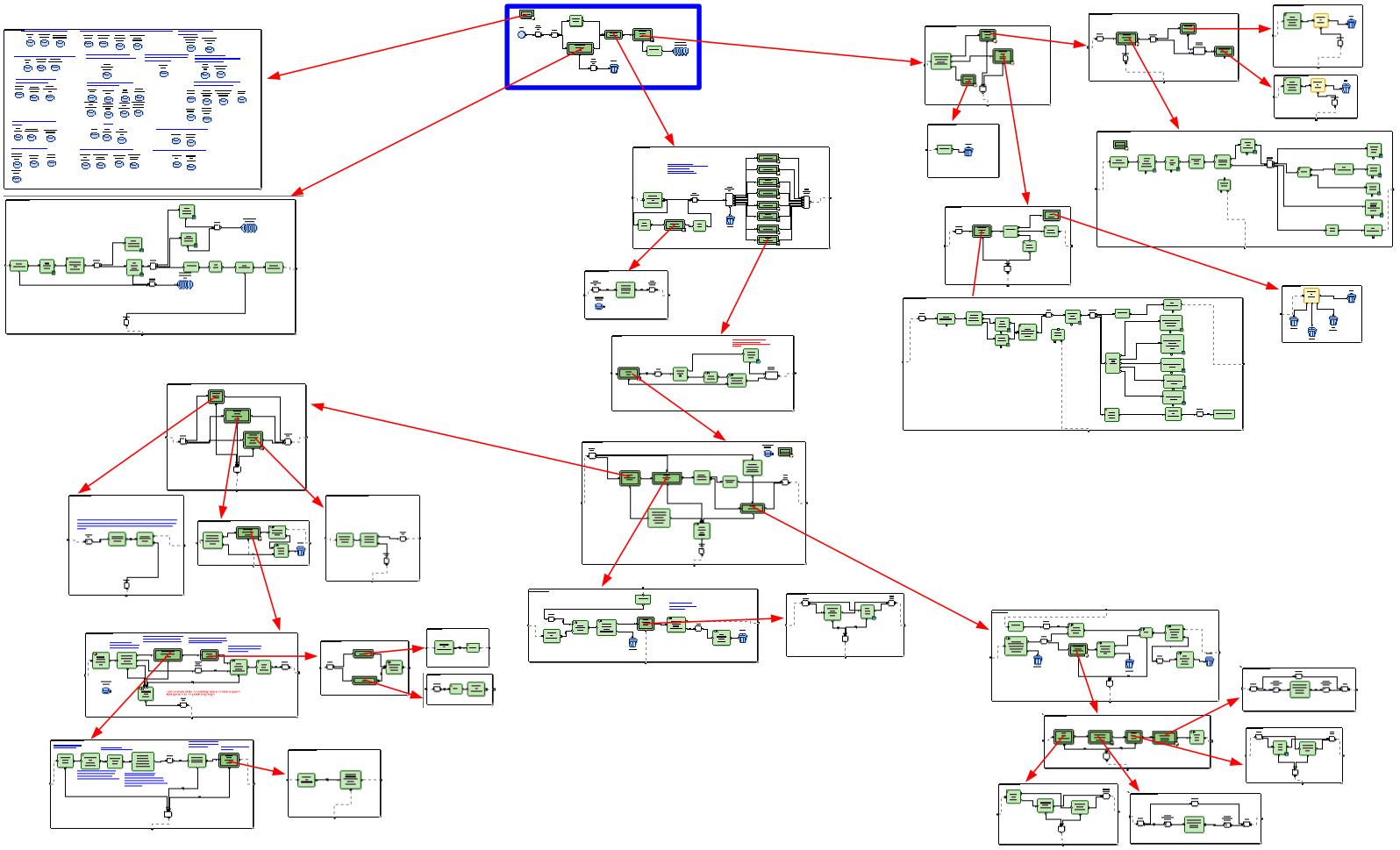
- Broader applicability
- More built-in functionality
- Handles extreme levels of complexity

Parallel Graphic Dataflow



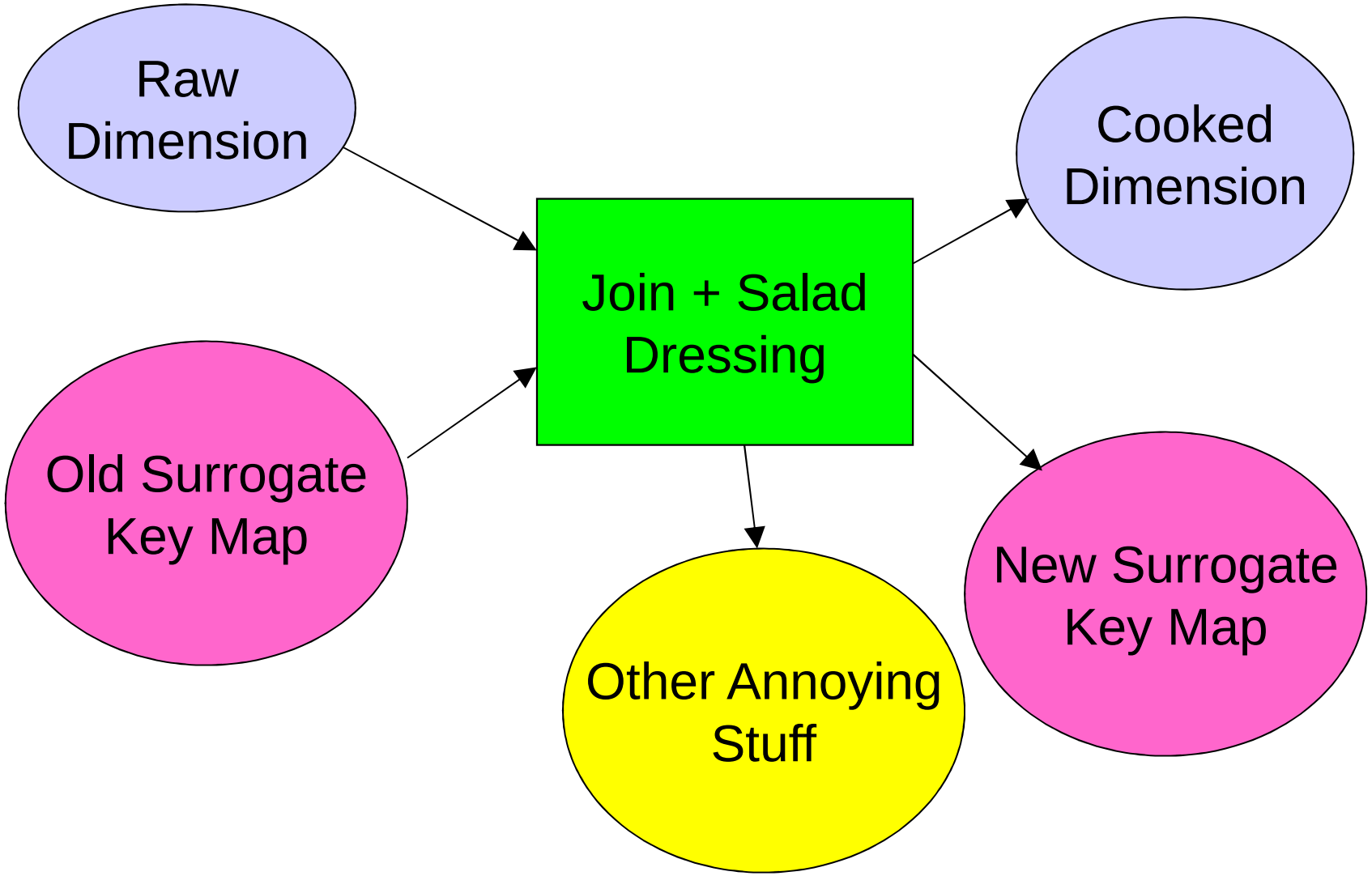
Real World: Deal with Errors, Log Output, Auditing Etc

Graphs Nest Very Deeply



9 Levels Deep; 33 Subgraphs; 259 Components

The Problem: Type 2 Slowly Changing Dimensi



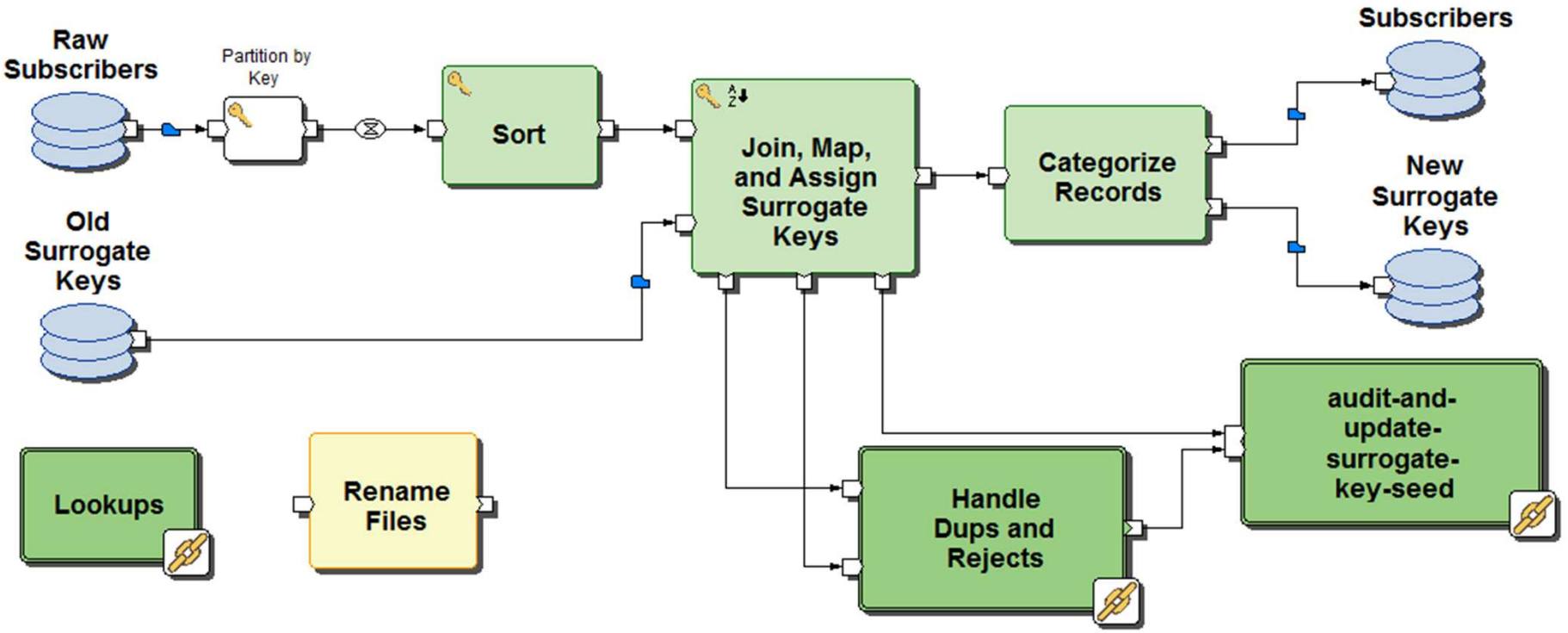
3 Cases

	Raw Dimension	Old Keymap	Cooked Dimension	New Keymap
Initial	Huge	Empty	Huge	Big
Full Reload	Huge	Big	Small	Big
Incremental Reload	Small	Big	Small	Big

Which cases do you optimize for?

- Initial Load: Big Job, Only Once
- Full Reload: Room for optimization
- Incremental: Lots of room for optimization
  (May not be a viable option)

Solution 1: Sort Merge Join



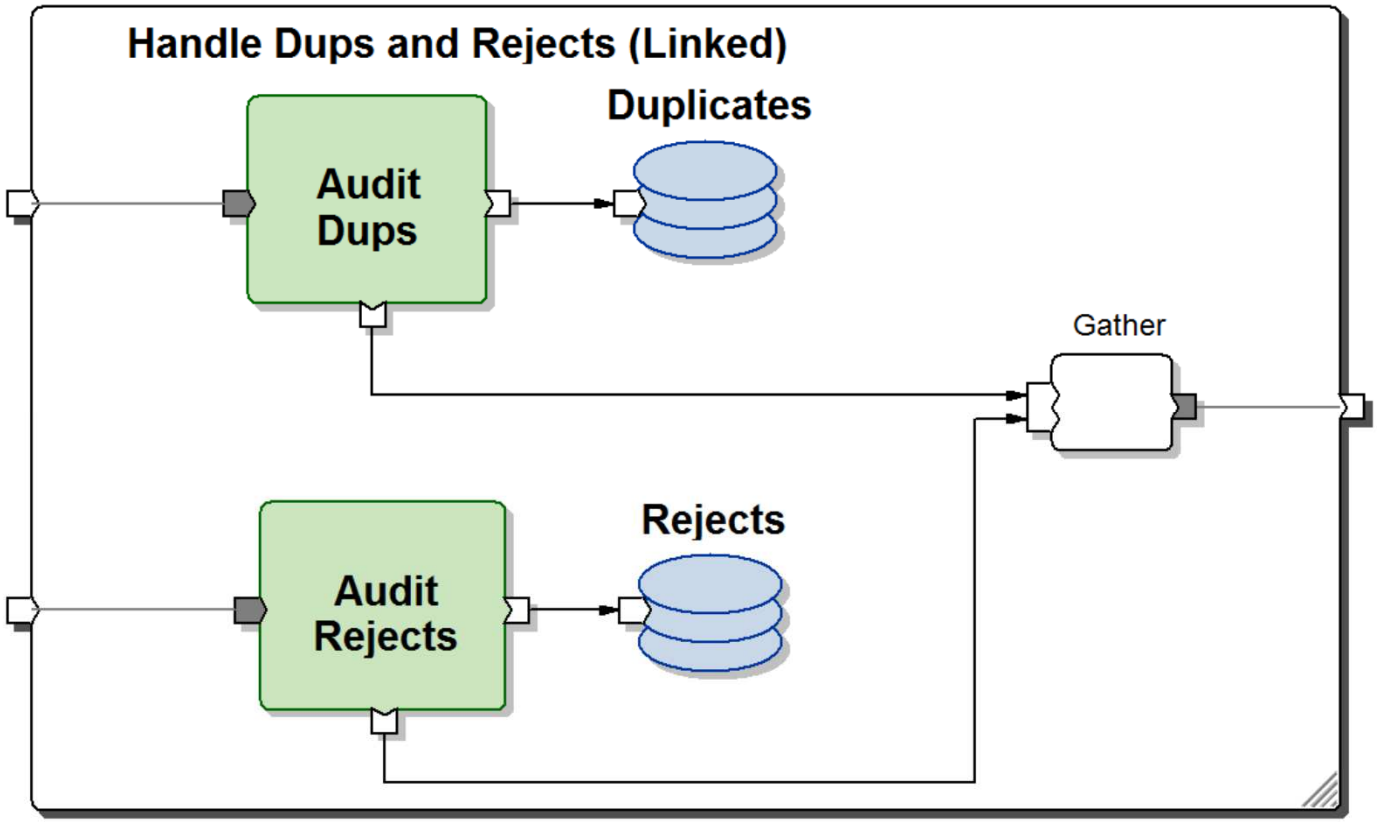
Depth: 2

Subgraphs: 3 (all shared)

Components: 27

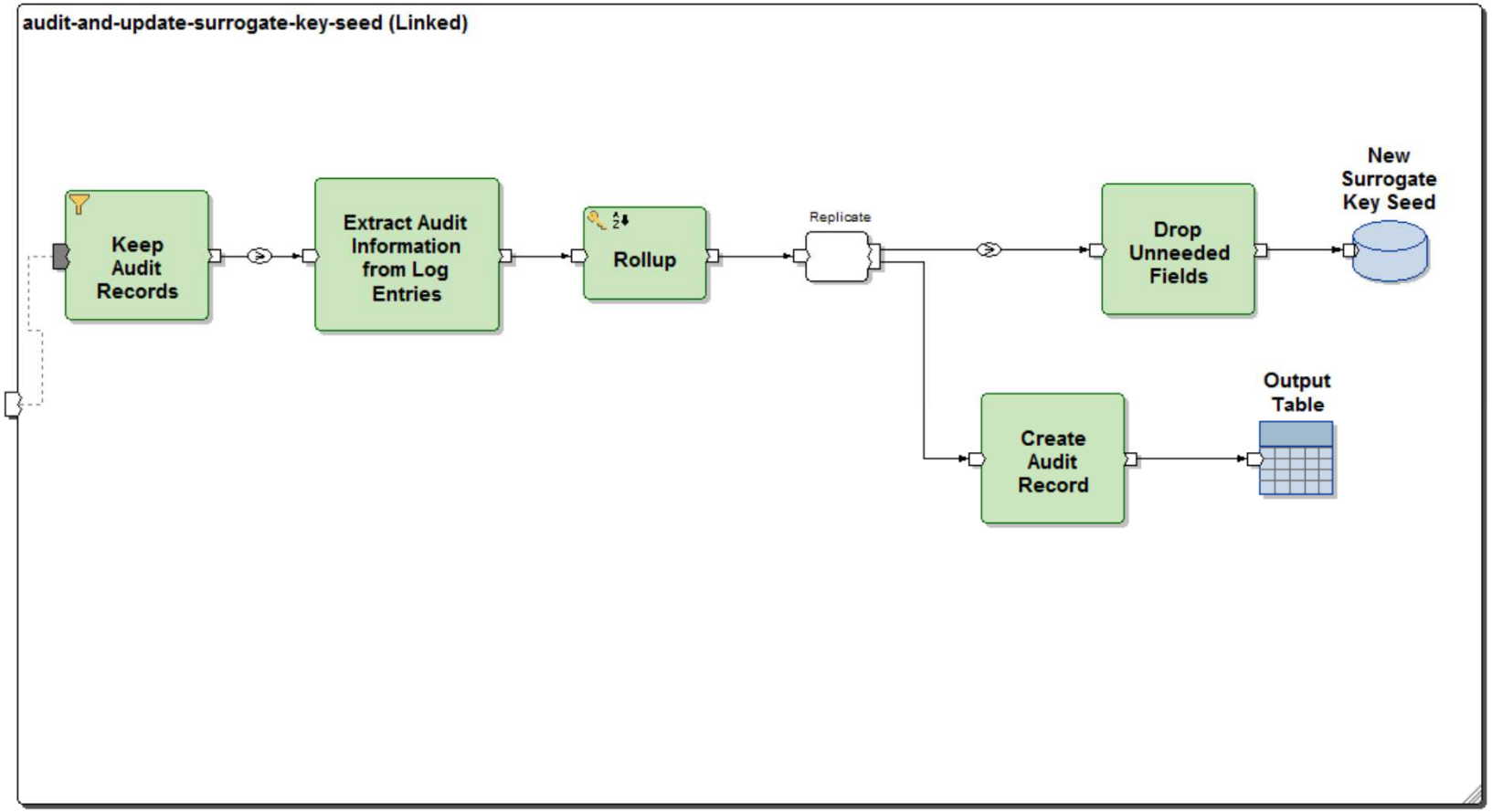
Salad Dressing: Handle Dups and Rejects

This is a **reusable subgraph**

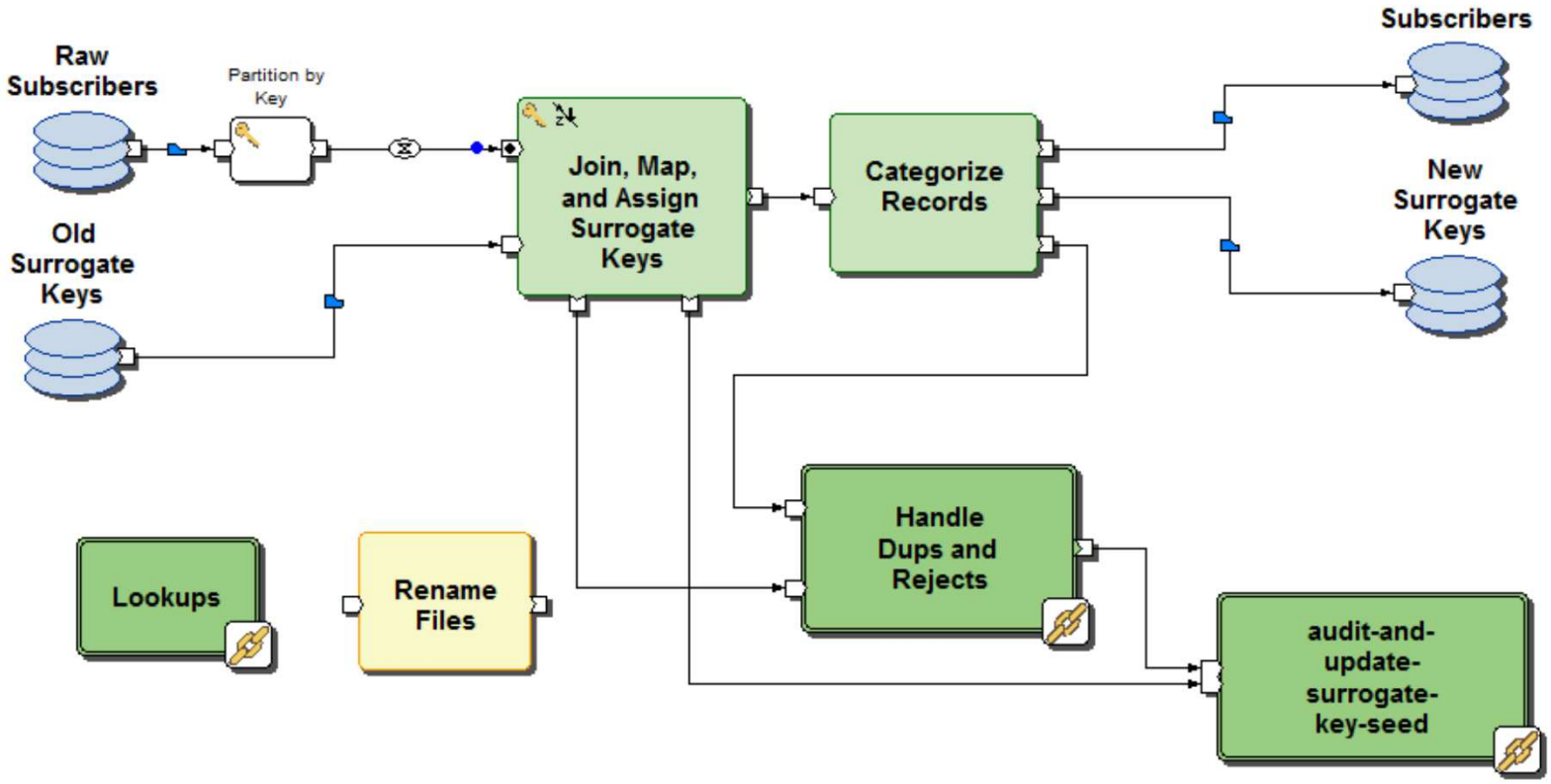


alad Dressing: Audit and Update Surrogate Key

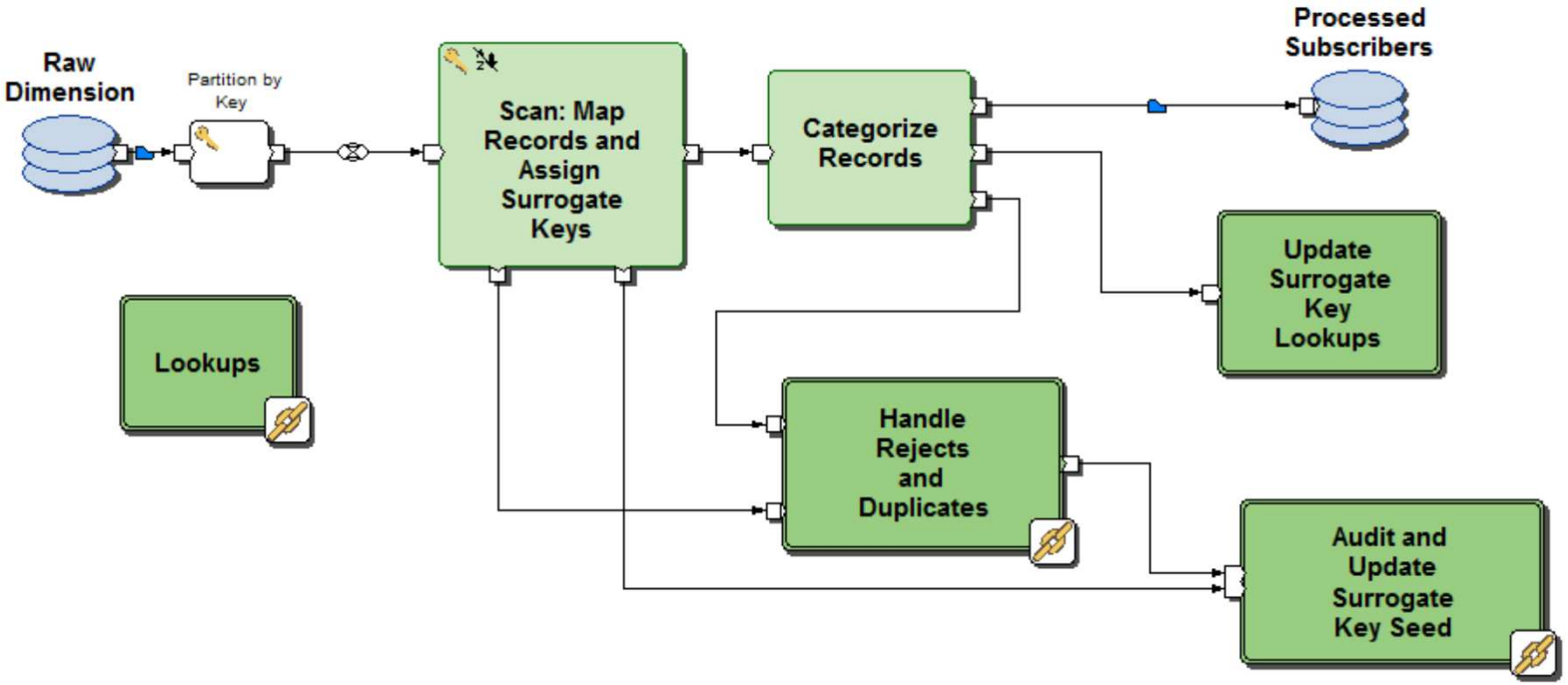
Read paper for how we generate surrogate keys



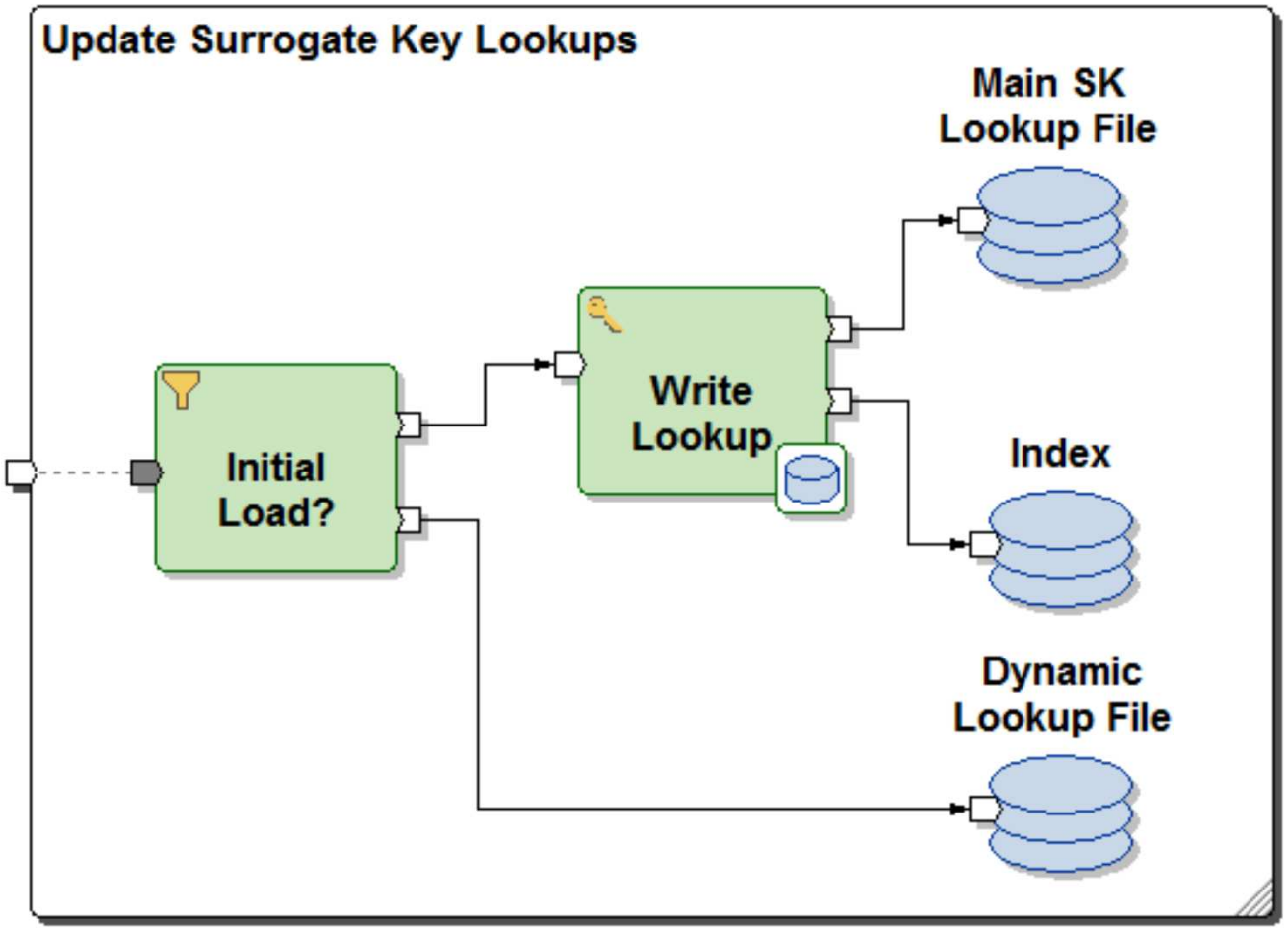
Solution 2: Hybrid Hash Join



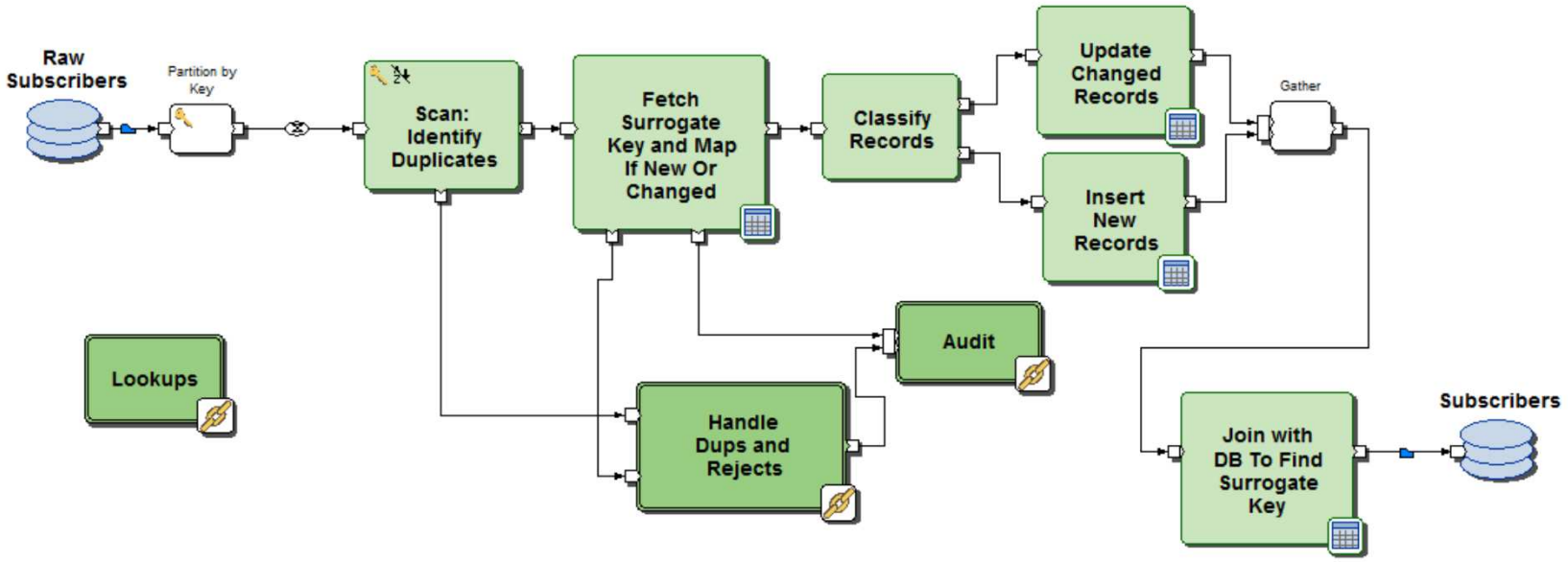
Solution 3: Lookup Files



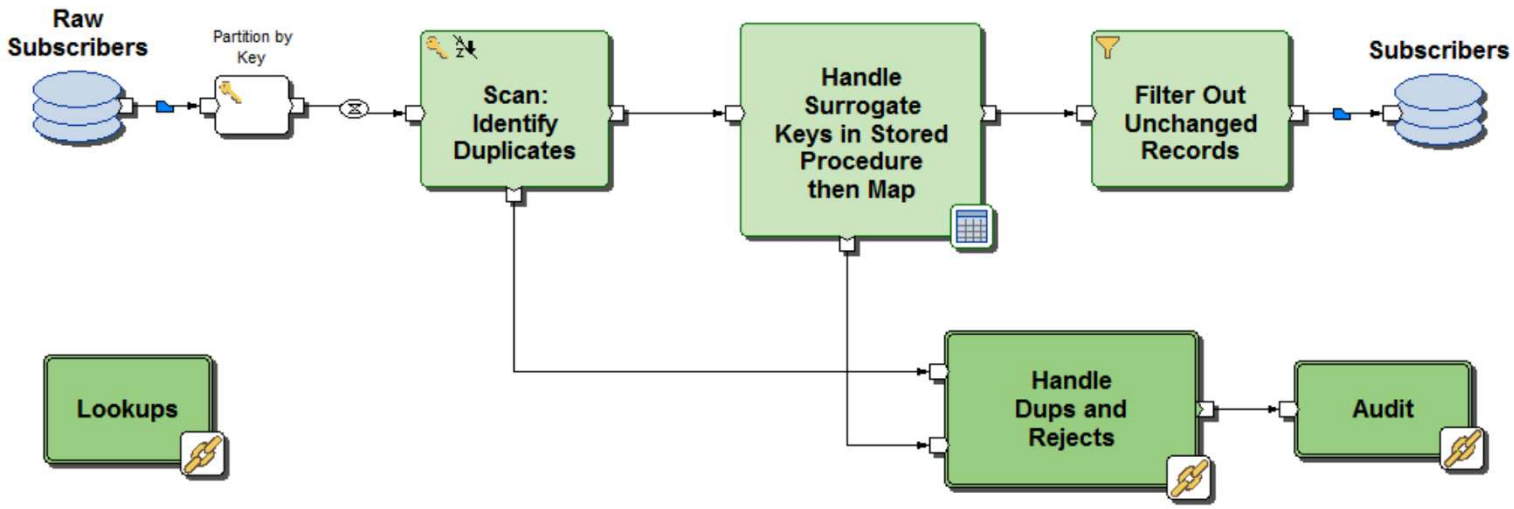
Solution 3: Lookup File Optimization



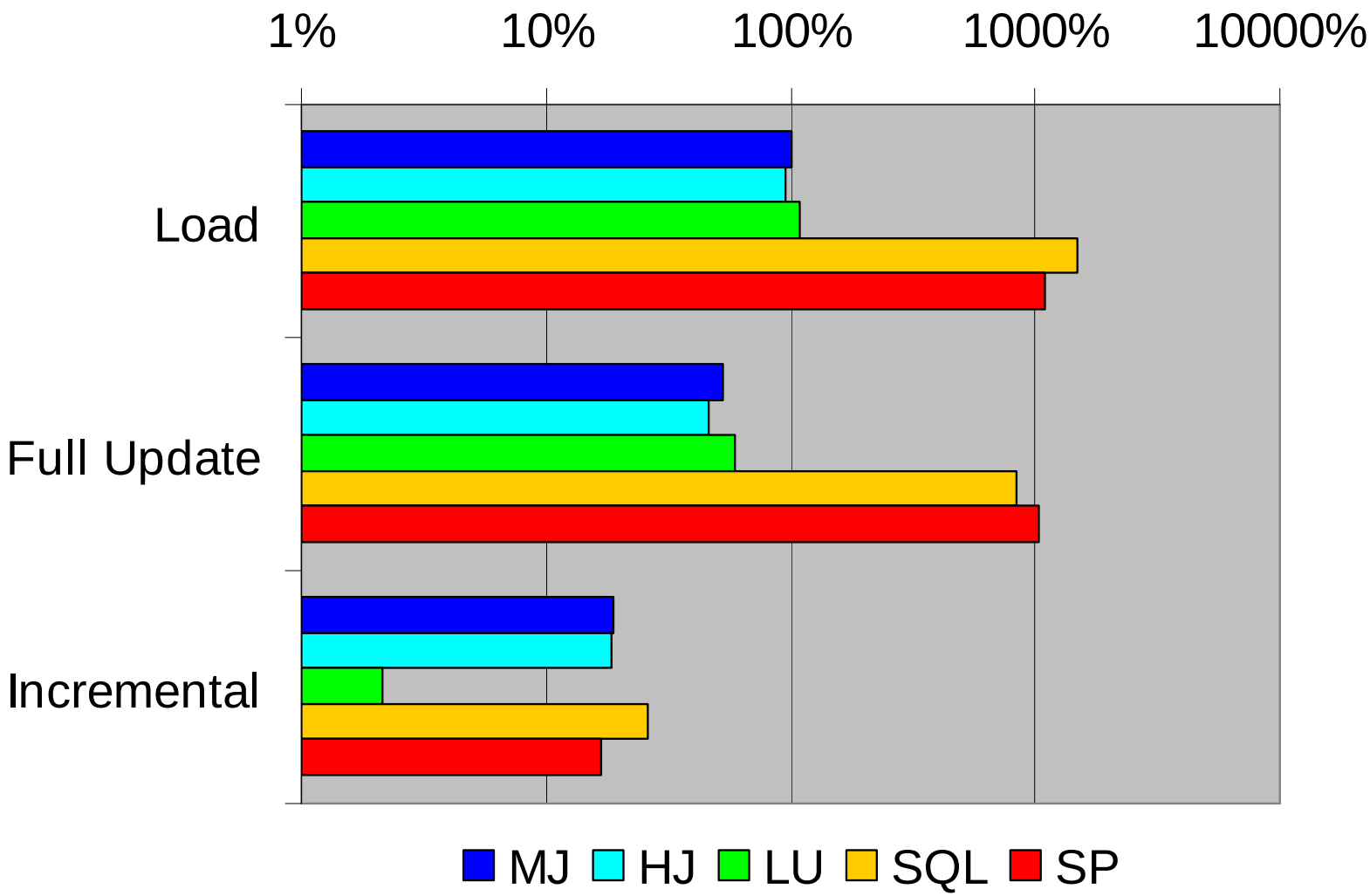
Solution 4: Keep Surrogate Keys in Database



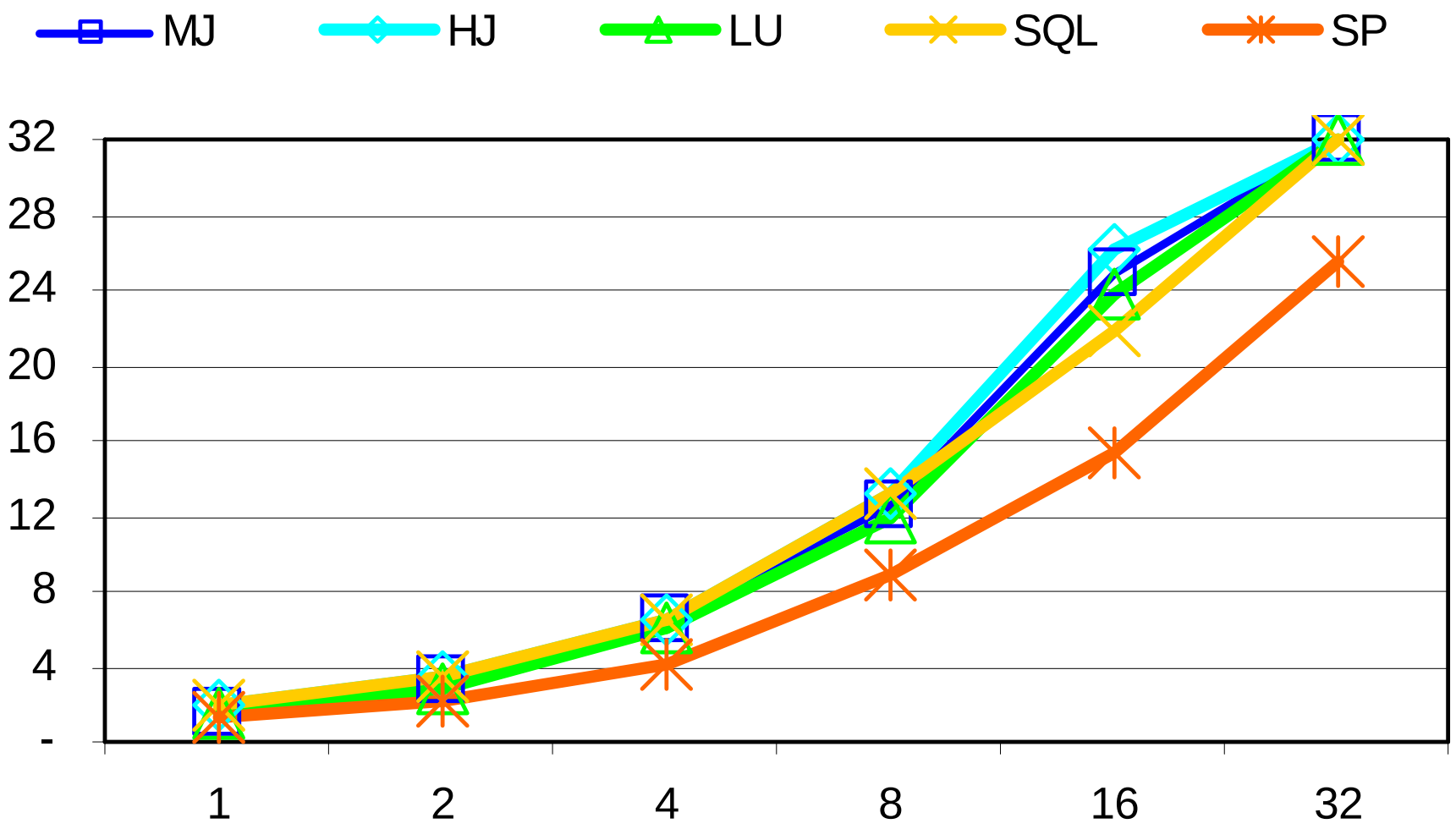
Solution 5: Stored Procedure



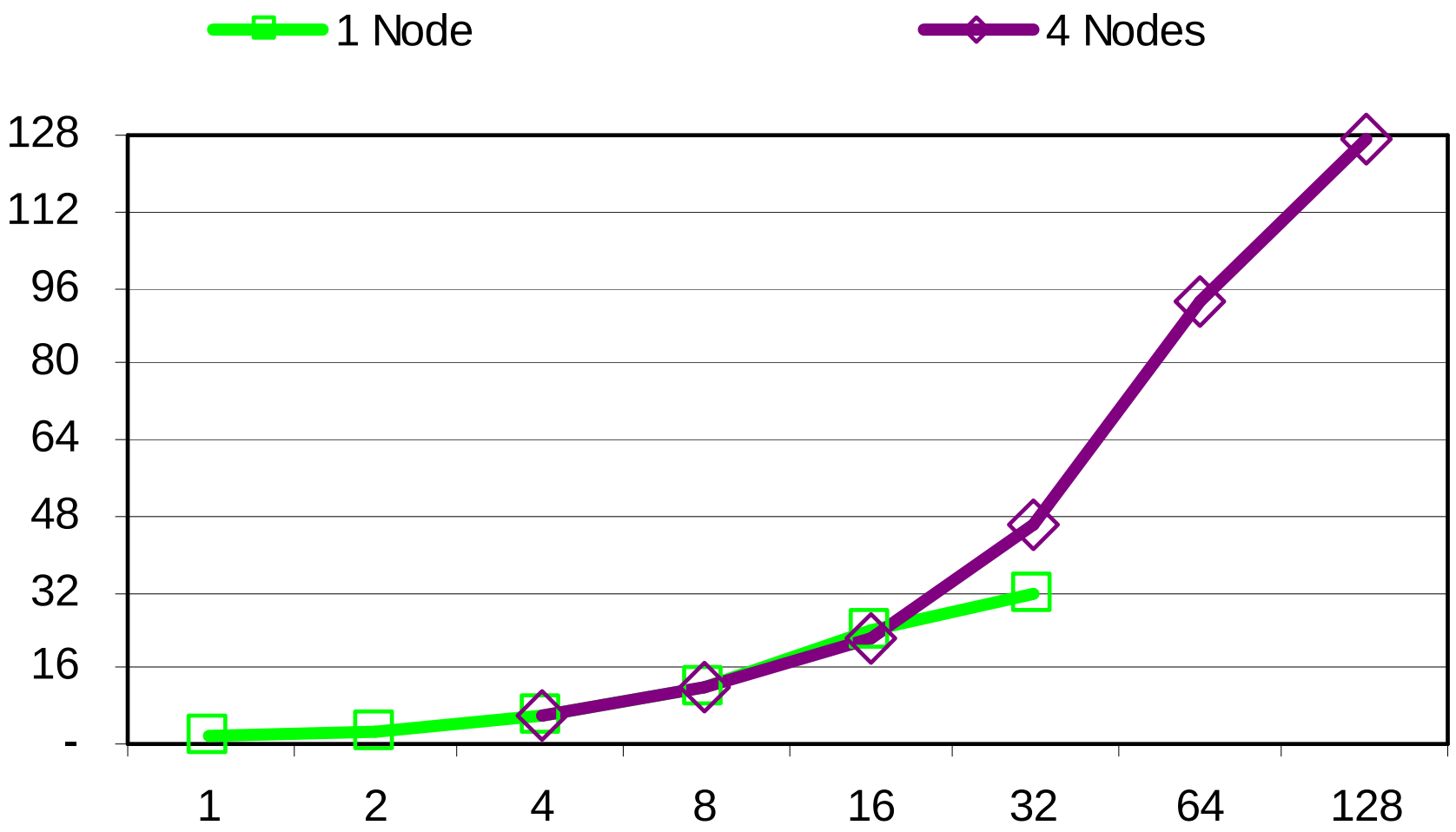
Wall Clock Time (32 ways parallel)



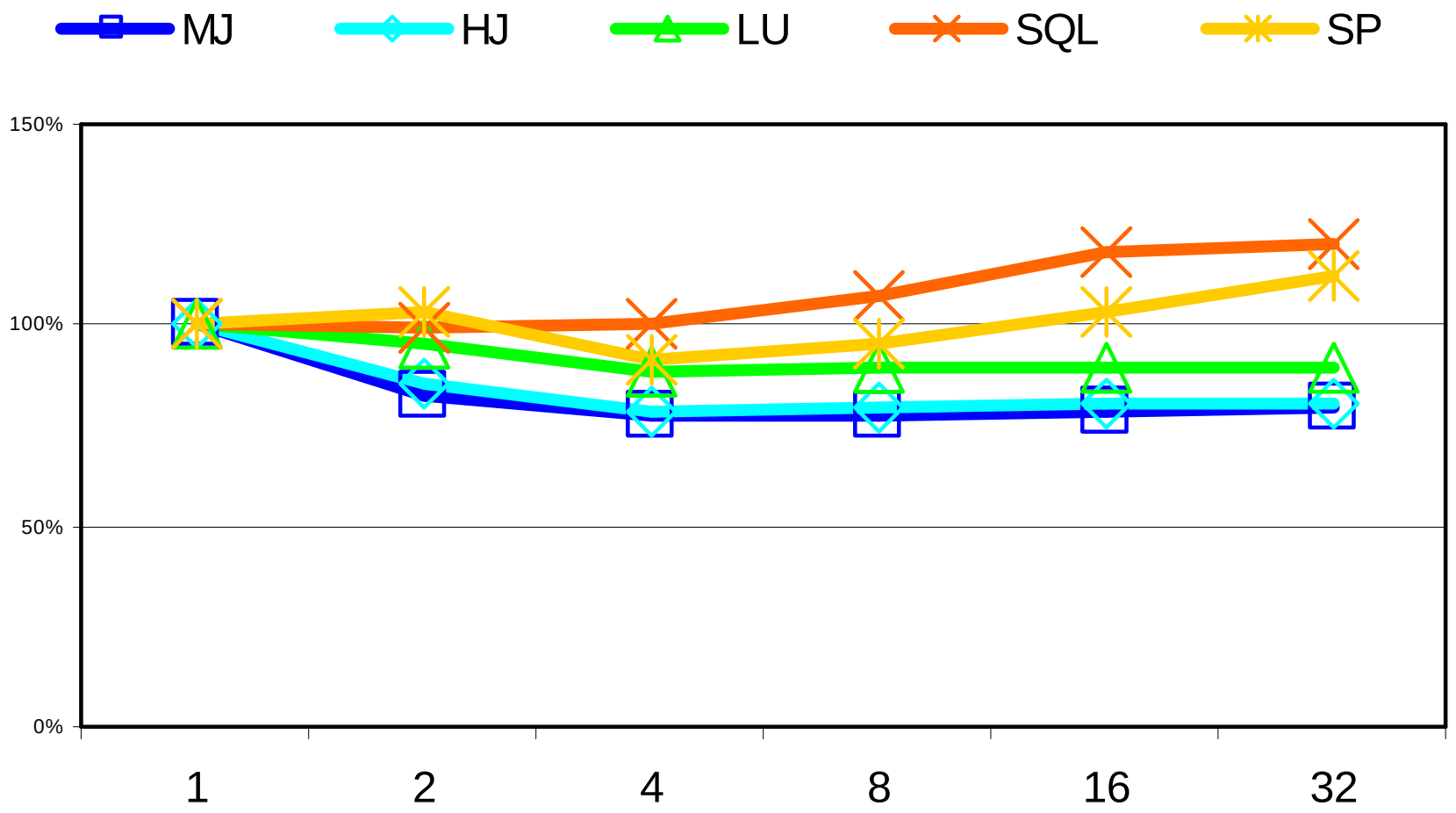
Cores Used



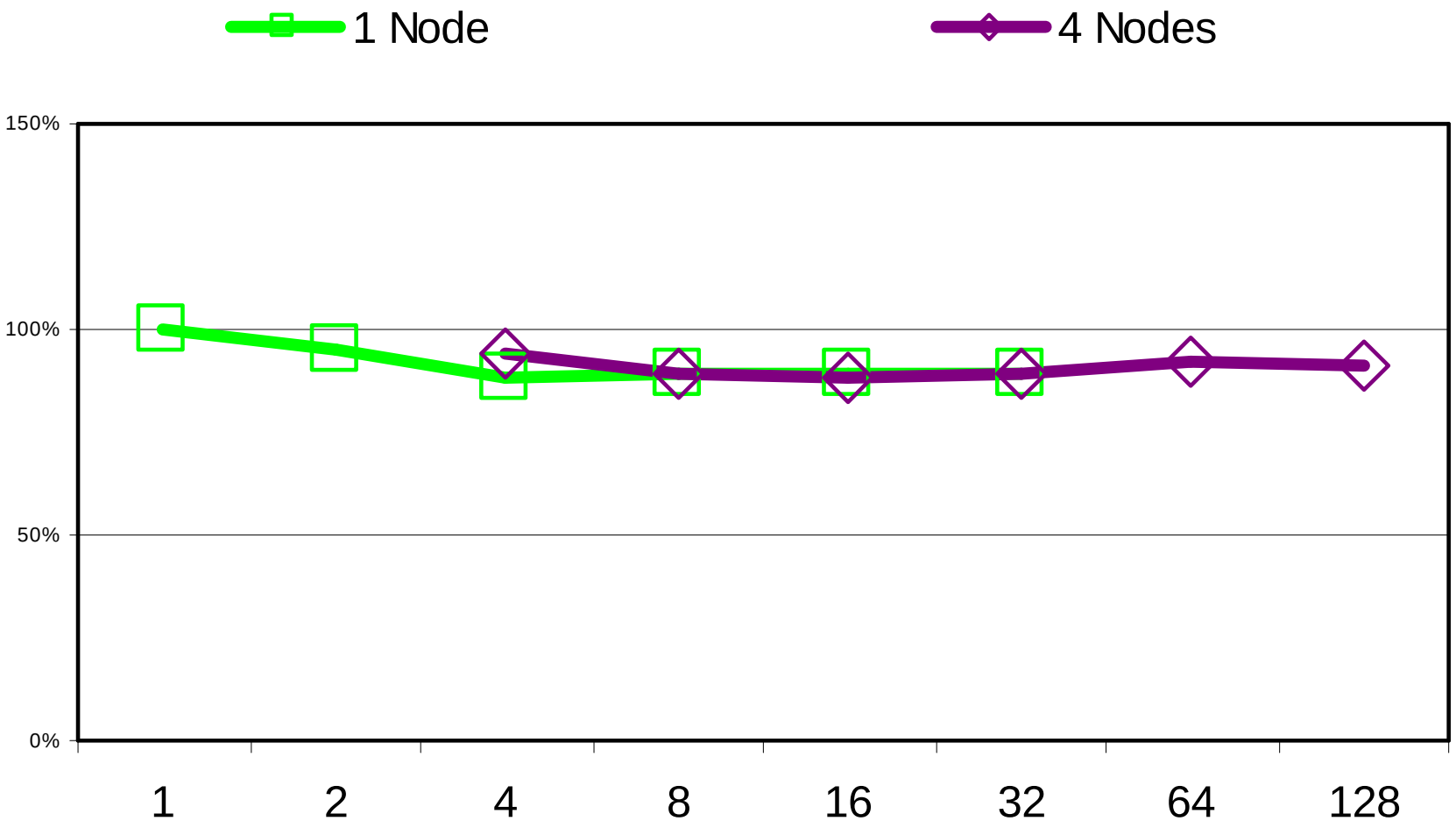
Cores Used: MPP (Lookup Solution, Initial Loa



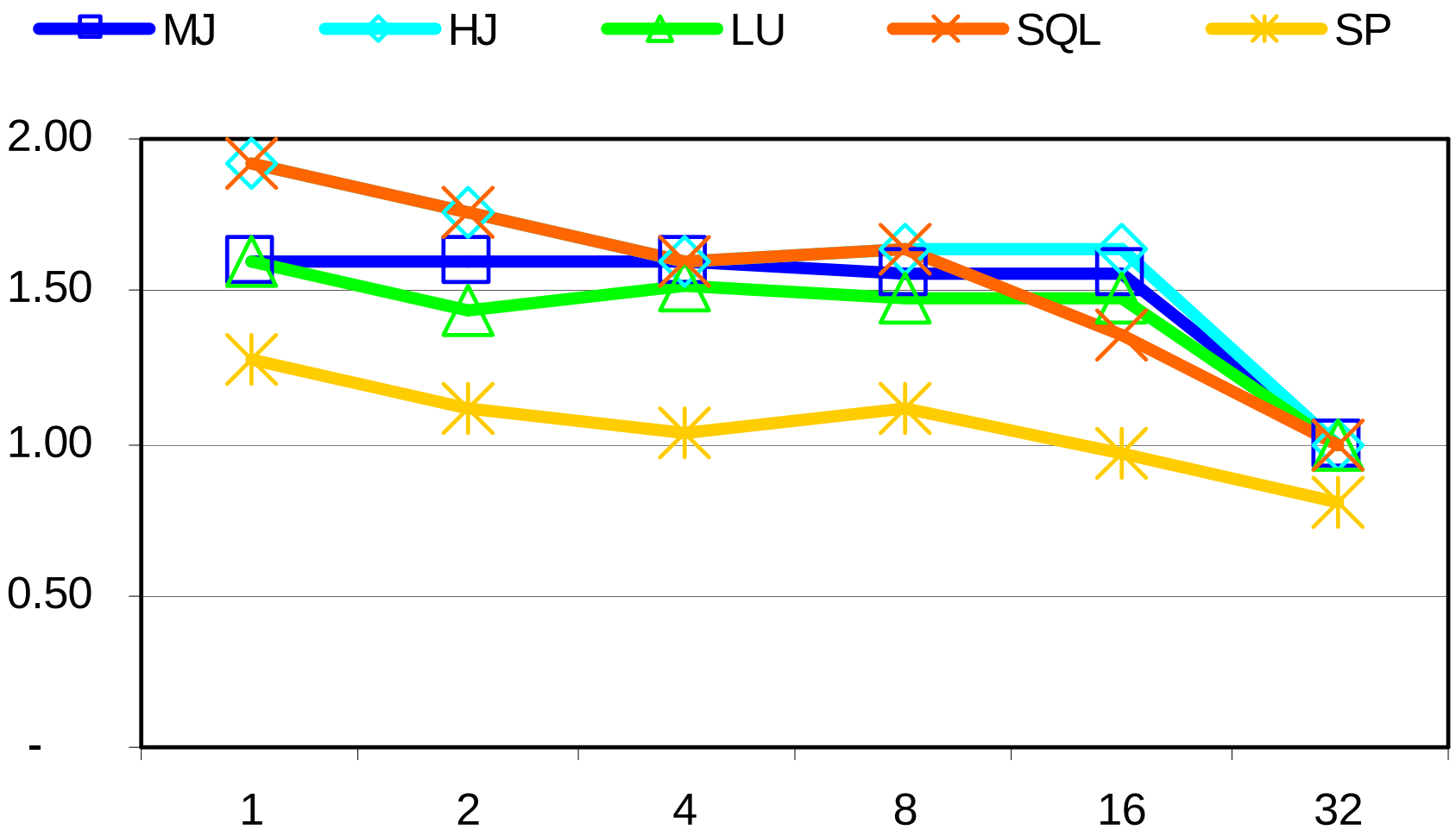
CPU Time/Record



CPU Time/Record (Lookup Solution, Initial Loa



Pipeline Factor



Questions?