

# Using Census Data for Grid Partitioning

Jonathan White, Dale Thompson  
University of Arkansas

## Motivation

With the growth of grid technologies, more and more companies are moving from large scale, centralized databases to databases that reside on grid based systems. One of the most important benefits that a grid can provide to the users of a database is the ability to process requests with a high degree of parallelism in order to minimize response time. One of the challenges that these companies face when migrating their database from a centralized environment to a grid environment is how to partition their data across the computers in the grid to promote load balancing and parallel retrieval.

We want to know if we can achieve a partitioning scheme with a high degree of parallelism using a grid-based database. We want to use public information from the US Census Bureau about the population distribution of the US. The Census Bureau has nearly every American first and last name, with their relative percentage in the population. We also have a Census Bureau file with every US zip code and the population that lives in that zip code.

## Approach

We performed an experiment comparing 2 partitioning schemes:

- One based on the distribution of **US Zip Codes**
- One based on **US Zip Codes and US Last Names**

### Experimental Design:

- Simulate a CORBA-based distributed system.
- System receives requests and directs them to be processed by a computer in the grid. The system can only process 2048 records at a time.
- System will be tested against a wide range of files:
  - Files from the general US population
  - Files from certain states
  - Files from certain cities
- The client files can be sorted or randomized.
- The client files will either be simulated or from actual sources.

**Goal** – See what partitioning scheme is more effective at balancing the load between the grid nodes.

**Result** – The partitioning scheme that was based on the distribution of both last names and zip codes was much more effective.

## The 6 most populous US zip codes

- |                       |                           |
|-----------------------|---------------------------|
| 1. 60623 CHICAGO, IL  | 4. 10025 NEW YORK, NY     |
| 2. 11226 BROOKLYN, NY | 5. 90201 BELL GARDENS, CA |
| 3. 10021 NEW YORK, NY | 6. 60617 CHICAGO, IL      |

## 40 most common US names, is yours here?

	Last	Female	Male
1.	SMITH	MARY	JAMES
2.	JOHNSON	PATRICIA	JOHN
3.	WILLIAMS	LINDA	ROBERT
4.	JONES	BARBARA	MICHAEL
5.	BROWN	ELIZABETH	WILLIAM
6.	DAVIS	JENNIFER	DAVID
7.	MILLER	MARIA	RICHARD
8.	WILSON	SUSAN	CHARLES
9.	MOORE	MARGARET	JOSEPH
10.	TAYLOR	DOROTHY	THOMAS
11.	ANDERSON	LISA	CHRISTOPHER
12.	THOMAS	NANCY	DANIEL
13.	JACKSON	KAREN	PAUL
14.	WHITE	BETTY	MARK
15.	HARRIS	HELEN	DONALD
16.	MARTIN	SANDRA	GEORGE
17.	THOMPSON	DONNA	KENNETH
18.	GARCIA	CAROL	STEVEN
19.	MARTINEZ	RUTH	EDWARD
20.	ROBINSON	SHARON	BRIAN
21.	CLARK	MICHELLE	RONALD
22.	RODRIGUEZ	LAURA	ANTHONY
23.	LEWIS	SARAH	KEVIN
24.	LEE	KIMBERLY	JASON
25.	WALKER	DEBORAH	MATTHEW
26.	HALL	JESSICA	GARY
27.	ALLEN	SHIRLEY	TIMOTHY
28.	YOUNG	CYNTHIA	JOSE
29.	HERNANDEZ	ANGELA	LARRY
30.	KING	MELISSA	JEFFREY
31.	WRIGHT	BRENDA	FRANK
32.	LOPEZ	AMY	SCOTT
33.	HILL	ANNA	ERIC
34.	SCOTT	REBECCA	STEPHEN
35.	GREEN	VIRGINIA	ANDREW
36.	ADAMS	KATHLEEN	RAYMOND
37.	BAKER	PAMELA	GREGORY
38.	GONZALEZ	MARTHA	JOSHUA
39.	NELSON	DEBRA	JERRY
40.	CARTER	AMANDA	DENNIS

## Did you know...

- There are 67 US zip codes with no one living in them
- The smallest zip code in Oklahoma is in Lawton, zip code 73770
- The most common last name in the world is Chang

## Experimental Solution

We designed our partitioning schemes as follows:

For Partition by only zip code:

1. Mod each zip code by 2 times the number of nodes in the grid. This separates zip codes that come from the same geographical area.
2. Match up the largest geo area with the smallest and put them on a node based on the total population in that area. This achieves balance among the grid.

For Partition by Last Name and Zip code:

1. Do the same as above for every zip code. However, in the above scheme, a zip code only lies on one node. We'd like to spread out each zip code across every node to achieve better parallelism.
2. Spread out each zip code by last name. Using the Census Bureau data, make a system of name ranges based on the population distribution of last names. For example, I used a table similar to this one where 1 percent of the US population lies in each line.

Offset	Name Ranges
1	AAAAAAAA ALI
2	ALICEA ANDERSON
3	ANDERTON AVERETT

## Results

The method that employed information about both the population distribution of last names and zip codes was much better than the method that just used information on zip codes. The chart below compares how the 2 partitioning schemes fared against different input client files. At times, the second scheme was around 6 times faster, a great improvement.

