Clusbench
Clustering Application Benchmark

Oğuz Altun
Nilgün Dursunoğlu
M.Fatih Amasyalı
{oguz, mfatih}@ce.yildiz.edu.tr
nilgunerd@garanti.com.tr

YILDIZ TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPT.
TURKEY

The Benchmark

- Clusbench measures and outputs total and individual run times of six related clustering algorithms.

What is clustering?

- The clustering algorithms' aim is to find clusters from unlabeled data.
- A *cluster* is a collection of objects which are "similar" to each other and are "dissimilar" to the objects that belong to the other clusters.
- Clustering algorithms have quite a number of application areas.

Example Application Areas

- Web domain:
  - Classifying web documents.
  - Discovering groups of similar access patterns from log data.
- Compression:
  - Reducing the number of colors in images. Similar colors are represented with a single colors.
- Marketing applications:
  - The customer groups with similar behavior are found by clustering customer properties and past buying records.
- The list goes on: Biology, city planning, earthquake studies, …

K-Means

- K-Means is probably the most widely used general clustering algorithm.
- Steps can be summarized as
  - Start by K random initial cluster centers,
  - Until cluster centers stop moving, iterate:
    - Reassign each object to the cluster with the closest center
    - Recalculate the position of cluster centers

SOM

- SOM is an artificial neural network model that can be used for clustering.
- It was first described by Teuvo Kohonen.
- It is especially good for visualizing high-dimensional data.
- To get consistent benchmark results on each run, our K-Means and SOM versions are modified to

  a) start with deterministic initial cluster centers,

  b) stop after given number of iterations.

Algorithms in Clusbench

- The algorithms in Clusbench are slightly modified versions of K-Means and SOM:
  - K-Means online
  - K-Means batch (Standart K-Means)
  - SOM-1D
  - SOM-2D
  - Hierarchical K-Means online
  - Hierarchical SOM-1D
- Details of the algorithms can be found in the proceedings and in the MS Thesis of Nilgun Dursunoglu (www.yildiz.edu.tr/~oaltun/clusbench/vq.pdf).

Benchmark Code

- Benchmark code is written in ANSI C.
- All the library code is in a single header file (clusbenc.h).
- Hence it can easily be integrated with other C/C++ benchmark codes.
- A supplied C program (clusbenc.c) serves
  a) as a standalone benchmark application ,
  b) as an example of the library usage.

Default Input Data Set

- By default, clusbenc.c uses an input dataset extracted from 920 Turkish news texts. Hence it has 920 rows.
- Each cell in a row shows the passing count of a word in the corresponding document. 11954 words are counted. Hence the dataset has 11954 columns.
- The dataset consists from 4 classes (economy, sport, politics, popular).

Using Different Dataset

- One can easily supply his/her own input dataset.
- Clusbench expects a .dst file as input.

The .dst File format

- The .dst file format is for storing two dimensional arrays of real values.
- First value in the file is the number of rows.
- Second value is number of columns.
- The rest are the row by row element values of the array.
- Values must be separated by white space.
- Only numbers and white space are allowed.

Portability Issues

- Only ANSI C functions are used in the code.
- All file names are in the 8.3 naming convention in case operating system has such restriction.
- The code is small in size, and easy to build:
  - On Unix like systems: cd to the directory, run “make”, then run “./clusbenc”
  - On other systems: open clusbenc.c in your IDE, compile, and run.

Tested Platforms and Results

Architecture	Operating System	Compiler	Time (Second)
Intel Pentium 4 CPU 3.00GHz 504 MB RAM	MS. WIN. XP PRO version 2002 SP2	MS. Visual Studio 2005 Professional Edition, Debug mode	16.391
"	"	MS. Visual Studio 2005 Professional Edition, Release Mode	11.578
"	"	MS. Visual C++ 6.0, debug mode	17,906
"	"	MS. Visual C++ 6.0, release mode	6.062
"	"	Borland C++ Builder Enterprise Suite, Version 6	16.844
"	"	GCC 3.4.2, thread model win32	15.969
"	"	GCC 3.4.2, thread model win32, with best optimization	15.845
Intel Xeon MP CPU 3.66 GHz 3.95 GB RAM	Suse Linux 9.3	GCC 3.3.5, thread model posix	15.720
Intel Pentium 4 CPU 1.70 GHz 440 MB RAM	Debian Linux Sid	GCC4.4.4, thread model posix	36,260

Availability

- Clusbench will be integrated into MineBench Benchmark Suite.
- It is still available standalone at www.yildiz.edu.tr/~oaltun/clusbench/html/

Questions?

Thank you!

Please direct your questions to Oğuz Altun
oguz@ce.yildiz.edu.tr