Bitanes2 - MPI Version

Author

- André Bannwart Perina
- Betweenness algorithm developed by Ulrik Brandes
    - Brandes, Ulrik. "A faster algorithm for betweenness centrality." Journal of mathematical sociology 25.2 (2001): 163-177.

Introduction

This is a simple MPI implementation of the betweenness centrality algorithm developed by Brandes. It was developed as a coursework for the "Parallel Programming Introduction" module offered at the University of São Paulo - Brazil.

Licence

GNU General Public Licence (see LICENSE file).

Prerequisites

- Usual GNU C compiler and runtime:
    - GNU C Compiler (gcc, tested with version 7.1.1 20170630);
    - GNU C Library (glibc, tested with version 2.25);
- A working installation of MPI (e.g. OpenMPI).

How to Download and Execute

1. Clone repo, access folder and download submodules:

```
git clone -b mpi https://github.com/comododragon/bitanes2.git
cd bitanes2
git submodule update --init --recursive
```

2. (a) Compile using `make` (see ***Description of Compiling Options*** for compiling settings):

```
make bin/bitanes2
```

2. (b) If you want to compile the master-slave version (see ***MPI Versions***):

```
make bin/bitanes2b
```

3. (a) Execute using `mpirun`, passing as argument the number of MPI nodes and the input graph, e.g.:

```
mpirun -np 8 ./bin/bitanes2 data/small/er_20_4_03.net
```

3. (b) For the master-slave version, execute using `mpirun`, passing as argument the number of MPI nodes (must be at least 2) and the input graph, e.g.:

```
mpirun -np 8 ./bin/bitanes2b data/small/er_20_4_03.net
```

4. The results will be available in the same folder as the input file, with the extension `.btw` (e.g. `data/small/er_20_4_03.btw`)

Description of Compiling Options

The Makefile provided with this project has some compilation options:

```
make bin/bitanes2 DEBUG=yes GPROF=yes GCOV=yes
```

where:

- `DEBUG=yes`: Activate debug flag `-g`;
- `GPROF=yes`: Activate flags for profiling with `gprof`;
- `GCOV=yes`: Activate flags for coverage test with `gcov`;

If one wants to change the options after compiling once, run `make clean` ***first.***

MPI Versions

There are two available versions using MPI:

- Standard (`src/bitanes2.c`):
    - Root node reads the graph and broadcast the data;
    - The main loop (which iterates through every node) is partitioned in similar sizes to all nodes;
    - Root node reduces the result and print the data to a file.
- Master-Slave (`src/bitanes2b.c`):
    - At least two MPI nodes are needed;
    - Root node reads the graph and broadcast the data;
    - Root node is responsible for deploying batches of the main loop to each slave node;
    - Slave nodes wait for batches, process them and ask for more after completion;
    - Root node reduces the result and print the data to a file.

The master-slave has an advantage of reducing load imbalance caused by the main loop. However, it is only beneficial on distributed machines. Resources contention on a single computer may mask any advantage of this approach.

Performance

All executions were performed on a Intel Xeon E5-1607. The sequential version (master branch) of this repository was used as baseline.

- Sequential
    - ***Execution times:***

        - ***Large graph 1 (2000 nodes, 31744 edges, filename data/big/ba_2000_32_01.net):*** 2.535 s
        - ***Large graph 2 (10000 nodes, 159744 edges, filename data/big/ba_10000_32_00.net):*** 61.4 s
- MPI Standard (4 nodes)
    - ***Execution times:***
        - ***Large graph 1 (2000 nodes, 31744 edges, filename data/big/ba_2000_32_01.net):*** 1.223 s
        - ***Large graph 2 (10000 nodes, 159744 edges, filename data/big/ba_10000_32_00.net):*** 33.952 s
    - ***Speedup:***
        - ***Large graph 1:*** 2.07x
        - ***Large graph 2:*** 1.81x
- MPI Master-Slave (4 nodes)
    - ***Execution times:***
        - ***Large graph 1 (2000 nodes, 31744 edges, filename data/big/ba_2000_32_01.net):*** 1.223 s
        - ***Large graph 2 (10000 nodes, 159744 edges, filename data/big/ba_10000_32_00.net):*** 33.551 s
    - ***Speedup:***
        - ***Large graph 1:*** 2.07x
        - ***Large graph 2:*** 1.83x

File Structure

- `bin`: folder for executable files;
- `data`: dataset of graphs;
    - `big`: large graphs (up to 10000 nodes);
    - `small`: small graphs (up to 20 nodes);
- `include`;
    - `common`;
        - `common.h`: procedures used for error detection and reporting (e.g. assert);
    - `graph.h`: header of graph data structure;
    - `list.h`: header of list/queue/FIFO data structure;
- `obj`: folder for object files (`.o`);
- `src`:
    - `bitanes2.c`: main function source (standard version);
    - `bitanes2b.c`: main function source (master-slave version);
    - `graph.c`: source of graph data structure;
    - `list.c`: source of list/queue/FIFO data structure;
- `Makefile`: project makefile.