

程序使用说明

陈 凯

复旦大学

本程序分为 CPU 和 GPU 两个版本的实现，并且实现了一个 UI 界面调用 CPU 和 GPU 内核。

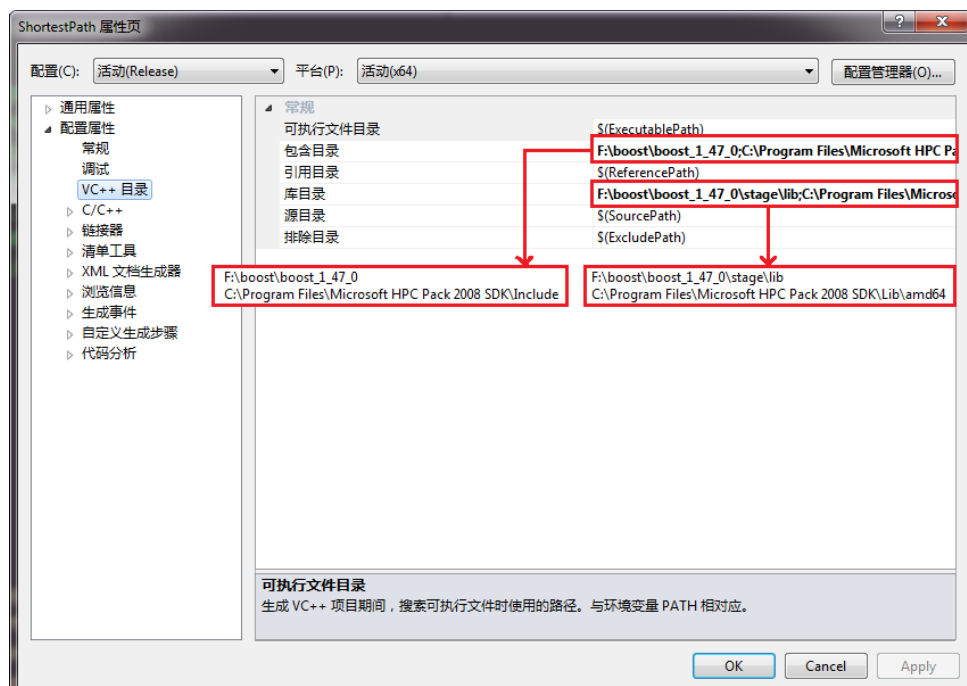
1) CPU

可执行文件存放在 bin/CPU。

软件环境: Windows 7 + VS2010 + Boost 1.47 + Microsoft HPC Pack 2008 SDK (以上软件都为 64 位)

HPC 和 Boost 的编译: 请先安装 Microsoft HPC Pack 2008 SDK, 确保安装在默认路径。之后安装编译 boost 库。Boost 库编译时, 在 boost 根目录下的配置文件 (project-config.jam) 中添加 using mpi ; 开启 mpi 的编译。在 boost_1_47_0\tools\build\v2\tools\mpi.jam 中 local cluster_pack_path_native 参数确保是 HPC 的安装目录。

VS2010 配置：VC++目录选项卡“包含目录”和“库目录”分别配置了 Boost 库和 Microsoft HPC 2008 SDK 的对应文件夹。如测试机器的程序安装目录不同，请做相应修改。



为了便于算法的执行，我调用了 Boost 库中的 METIS 文件格式作为算法的输入格式。需先将大赛中的测试文件*.gr 转换成*.metis 文件（已在执行文件中集成）。转换文件以及读入需一定时间，请耐心等待。

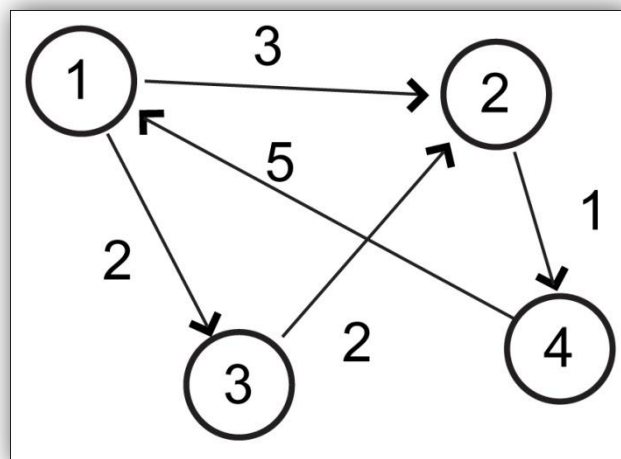
Metis 格式简要说明：

第一行：n m type

其中 n 为节点数，从 1 到 n 编号。m 为总边数。type 是图的类型，1 表示链表。

以下 n 行，第 i 行代表节点 i 的链表。每行中有 p 对数，一对数中，第一个为节点号，第二个为距离。表示节点 i 到此节点的距离。

如图：



metis 文件表示为：

```
4 5 1
3 2 2 3
4 1
2 2
1 5
```

图中有 4 个节点，5 条边。且按照链表方式输出。

另外由于 metis 中的阅读器将每个节点索引分别减 1，所以，节点从 0 开始编号。在算法中，我做了相应处理。

Metis 文件的具体信息可参考[这里](#)。

程序使用方法:

ShortestPath.exe [method] [filename] [source] [target]

ShortestPath.exe -h 得到帮助

命令事例: Shortest.exe dijkstra USA-road-d.NY.gr 或

Shortest.exe 1 USA-road.d.NY.gr

用 dijkstra 算法法计算 USA-road-d.NY.gr 中的节点 1 到最大节点 n 的距离。

其中:

method 表示选用何种最短路径算法:

0) all: 所有的方法都将被测试

1) dijkstra: 串行 Dijkstra 算法。

2) crauser: 并行 Crauser 算法。

3) delta-stepping: 并行 Delta Stepping 算法。

4) bellman-ford: 串行 Bellman Ford 算法。

指定方法名称或索引号皆可。

filename: 输入的图文件。

source: 源节点 (可选)。如果没有指定, 默认节点为 1。

target: 目标节点 (可选)。如果没有指定, 默认节点为 n。(即图中索引号最大的节点)

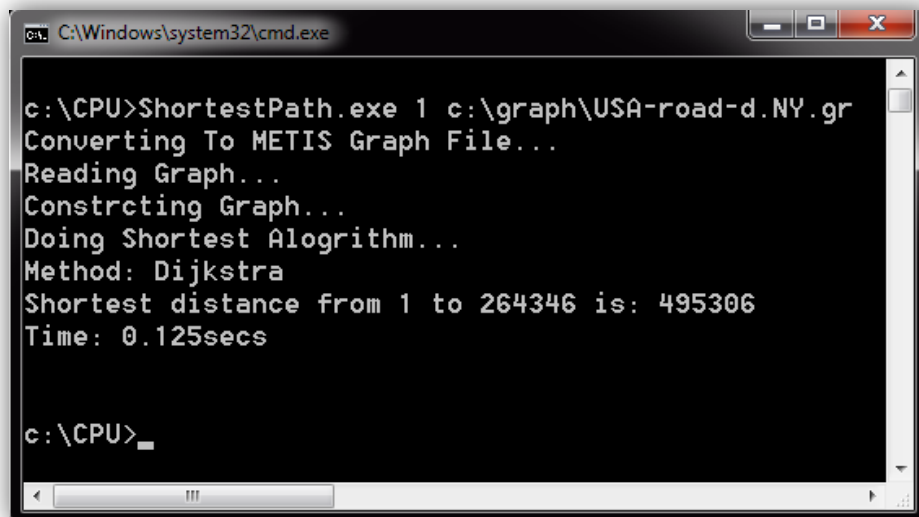
输出格式:

Method: 采用的方法

Shortest distance from [source] to [target] is: 从节点 source 到 target 的最短路径

Time: 算法耗时

以下是输出地图 USA-road-d.NY.gr 中从节点 1 到节点 n 的最短路径。



```
C:\Windows\system32\cmd.exe

c:\CPU>ShortestPath.exe 1 c:\graph\USA-road-d.NY.gr
Converting To METIS Graph File...
Reading Graph...
Constructing Graph...
Doing Shortest Alogrithm...
Method: Dijkstra
Shortest distance from 1 to 264346 is: 495306
Time: 0.125secs

c:\CPU>
```

2) GPU

可执行文件存放在 bin/GPU 中。

软件环境:

Windows 7 + VS2010 + Nvidia GPU Computing SDK 4.0+ Parallel Nsight 2.0 (以上软件都为 64 位)

在 GPU 的算法中, 不需要将文件转换为 metis 格式。

命令格式和 CPU 部分基本相同 (多出 printPath 选项):

ShortestPath.exe [method] [filename] [source] [target][printPath]

ShortestPath.exe -h 得到帮助。

命令事例: Shortest.exe cuda-delta-stepping USA-road-d.NY.gr 或

Shortest.exe 3 USA-road-d.NY.gr

用 CUDA Delta-Stepping 算法计算 USA-road-d.NY.gr 中的节点 1 到最大节点 n 的距离。

其中:

method 表示选用何种最短路径算法:

0) all: 所有的方法都将被测试

1) cuda-dijkstra: CUDA Dijkstra 算法。

2) cuda-bellman: CUDA Bellman Ford 算法。

3) cuda-delta-stepping: CUDA Delta-Stepping 算法。

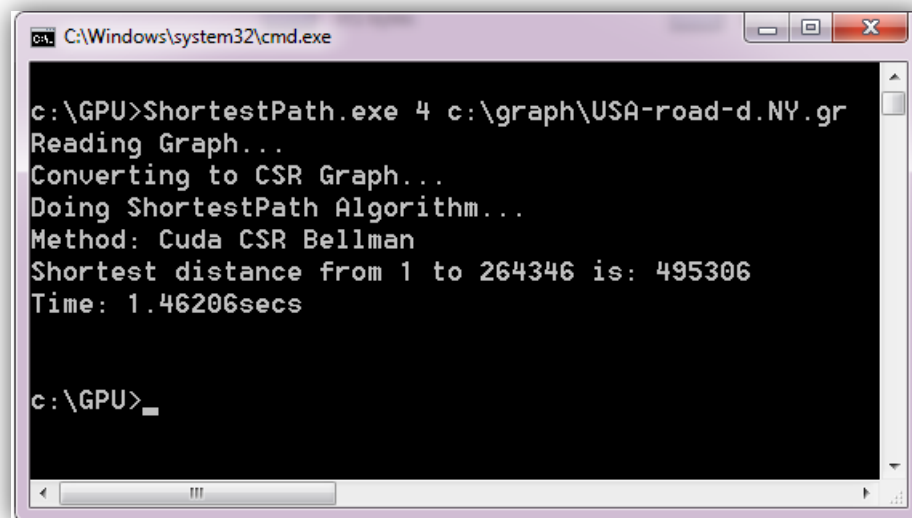
4)cuda-csr-bellman-scalar: CUDA CSR Bellman Ford (Scalar) 算法。

5)cuda-csr-bellman-vector: CUDA CSR Bellman Ford (Vector) 算法。

6)cuda-ell-bellman: CUDA ELL Bellman Ford 算法。

printPath 选项（可选）：0 为不输出路径，1 为输出路径。如果没有指定，默认为 0。

其余命令参数和 CPU 版本相同。



```
C:\Windows\system32\cmd.exe

c:\GPU>ShortestPath.exe 4 c:\graph\USA-road-d.NY.gr
Reading Graph...
Converting to CSR Graph...
Doing ShortestPath Algorithm...
Method: Cuda CSR Bellman
Shortest distance from 1 to 264346 is: 495306
Time: 1.46206secs

c:\GPU>
```

3) UI 界面

为了便于程序使用，我用 C#实现了一个 UI 界面调用上述的 cmd 文件。由于结果的显示非实时，等 cmd 全部计算完成才能显示。所以如需了解程序执行进度，请使用命令行界面。

Shortest Path Problem

图名称 : USA-road-d.NY.gr

选择

CPU Program: ShortestPath.exe

选择

GPU Program: 未选

选择

可选节点范围 : 1 到 264346

源节点 : 1

目标节点 : 264346

开始

计算方法

☒ CPU ☐ GPU

Dijkstra

☐ 输出路径

结果 :

Converting To METIS Graph File...
Reading Graph...
Constrcting Graph...
Doing Shortest Alogrithm...
Method: Dijkstra
Shortest distance from 1 to 264346 is: 495306
Time: 0.09secs