M. Troyer, P. Koumoutsakos
ETH Zürich, HIT G 31.8
CH-8093 Zürich

Set 3 - Sparse Linear Algebra

Issued: March 2, 2015
Hand in: March 9, 2015

Question 1: Page Rank

The order of results generated by a search engine depends on the relative popularity between pages within a network and is determined by the page rank. In this exercise we will make use of the page rank matrix for two different applications. Compute the 20 most visited sites within the ETH network and find out which were the most important US patents from 1975 to 1999.

A network with N nodes can be represented by a so called transition matrix A of size $(N \times N)$ containing probabilities to jump between each individual node. Each column j of A represents a node with $M_j \leq N$ links and $A_{ij} = 1/M_j$ if there is a link from site j to site i. All other elements of A are zero.

The page rank vector v is the steady state of this matrix, i.e. it is unchanged by a traversal through the network and therefore satisfies

$$Av = v \tag{1}$$

There is a problem. If a node has no outgoing links the corresponding column in A will be zero. This will create a sink at that node and complicate finding the page rank vector. Likewise it would be problematic if a node does not contain any incoming links. One solution is the Google matrix B

$$B = (1 - p)A + pR \tag{2}$$

where p[1] is the chance of jumping to a random page in the network and $R_{ij} = 1/N \; \forall \; i, j \in [0, N)$. We can now find v by solving $Bv = v$ instead.

The transition matrices are provided in the files `ETH_network.mtx`[2] and `US_patents.mtx`[2,3] in the format [row, column, value]. The third line in the file defines the dimensions of A and the number of non-zero elements.

a) Parse the file (*hint: only non-zero values are listed*) and write a container for matrix A which allows efficient computation. You can build on the skeleton code for the matrix parser in the file `page_rank/matrix_parser.hpp`

[1] for the ETH network use $p \sim 0.3$ and for the US patents $p \sim 0.1$

[2] can be found in `/cluster/scratch/hahna/hpcse_lecture_fs15/` on Euler

[3] the first row (index $= 0$) of the patents matrix represents all patents before 1975

The matrix is stored in a list-of-lists (LIL) format and we can parse line by line the corresponding row and column indices together with the matrix elements. The LIL format is convenient for writing/reading, but for storage and linear algebra it is inefficient. We will therefore load the matrix into a CSR format, which uses less memory than LIL and is more suitable for computation. Additionally, note that the transition matrices for US patents citations and the ETH network represent directed graphs and are not symmetric.

b) Write a serial code to calculate the page rank vector using the power method.

The power method is not the most efficient, but is simple to implement and relatively stable, requiring ~ 30 iterations to converge. We should normalise v every iteration and choose an optimal p for the current problem.

We can now sort the vector v according to the page rank and find the entries pointed by the corresponding index. Lookup lists of ETH pages and patent numbers are provided in `ETH_network.lst`[2] and `US_patents.lst`[2]

c) Sort the page rank vector and match it to the corresponding lookup entries. A prototype for the lookup parser can be found in `page_rank/lookup_parser.hpp`

A simple way is to make a vector of a data type containing the lookup definition and page rank according to which we can sort.

d) List the 20 most visited sites in the ETH network and the most important US patents [4].

The top 20 ETH sites and US patents are listed below in Table 1 and Table 2.

e) Parallelize you code and make a strong-scaling plot up to 24 cores.

If we store the matrix in CSR format, a straightforward way is to parallelize the outer loop over the rows of the matrix A corresponding to segments of the page rank vector v and divide the number of rows equally between each process. However, the transition matrix is unstructured and the number of non-zero elements per row can vary, which can result in different amount of work given to each thread. For example older US patents have more incoming citations (non-zero row elements) than newer ones. As we can see in fig. 1 the scaling is better for larger matrices (US patents), but still sub-optimal compared to applying some structured stencil in the diffusion equations (see previous semester). A more complex, but less general solution would be to optimise the multiplication for the specific structure of a matrix.

The matrix vector multiplication is a bandwidth limited problem – each value in the matrix is used only for one operation. This effect is visible in the last part of the scaling, where the speedup is flattening.

Parallel code: `PageRank_OpenMP.cpp`

[4] the patents can be looked up according to their patent number on `http://patft.uspto.gov/netahtml/PTO/srchnum.htm`

Table 1: 20 most linked website in the ETH network.

http://www.ethz.ch	0.59741
http://www.cd.ethz.ch/services/web	0.434388
http://www.hk.ethz.ch	0.240311
http://www.ethz.ch/index_EN	0.236807
http://www.math.ethz.ch/rss/bulletin	0.217827
http://www.hk.ethz.ch/index_EN	0.214507
http://blogs.ethz.ch/klimablog/feed	0.134142
https://blogs.ethz.ch/klimablog/feed	0.08772
http://www.ethz.ch/index	0.0744939
http://www.ethz.ch/people/index	0.0694293
http://www.ethz.ch/libraries/index	0.0685078
http://www.ethz.ch/research/index	0.068397
http://www.ethz.ch/help/index	0.0682438
http://blogs.ethz.ch/klimablog/comments/feed	0.068241
http://www.ethz.ch/imprint/index	0.0682313
http://www.ethz.ch/disclaimer/index_DE	0.0680948
http://www.ethz.ch/studies/index	0.0677354
http://www.ethz.ch/continuing/index	0.0677054
http://www.ethz.ch/news/index	0.0676338
http://www.ethz.ch/about/index	0.0675803

Table 2: 20 most cited patents among the US patents from 1975 to 1999.

	old patents	0.254558
3935707	Hydraulic control system	0.000158918
4247191	Projection colour copier	0.000151492
3992961	Reversible gear system	$9.10075 \cdot 10^{-5}$
4309918	Retarding means for motor vehicles	$7.84671 \cdot 10^{-5}$
4027055	Tin plating by immersion	$6.91466 \cdot 10^{-5}$
4173204	Control valve of exhaust	$5.89012 \cdot 10^{-5}$
4408626	Valve seat for a steam trap	$5.86834 \cdot 10^{-5}$
3949624	Lifting linkage for roof vent panels of automobiles	$5.75856 \cdot 10^{-5}$
3950607	Bandwidth compression system	$5.5945 \cdot 10^{-5}$
3953665	Bushings, grommets or like devices	$5.39684 \cdot 10^{-5}$
4070809	Automatic sugar cane harvesting machine	$5.21188 \cdot 10^{-5}$
4020383	Pulsing incandescent lamp filaments	$5.03404 \cdot 10^{-5}$
4080735	Scraper blade	$4.98989 \cdot 10^{-5}$
3877340	Concrete penetrating pin	$4.76484 \cdot 10^{-5}$
3890102	Catalytic action gas generator	$4.62873 \cdot 10^{-5}$
4123107	Sulphuric acid production system	$4.54874 \cdot 10^{-5}$
4321222	Manufacturing anisotropic permanent magnets	$4.48696 \cdot 10^{-5}$
4017895	Detecting defects in read out signals	$4.2777 \cdot 10^{-5}$
3988926	Determination energy transferred by a flowing fluid	$4.2582 \cdot 10^{-5}$

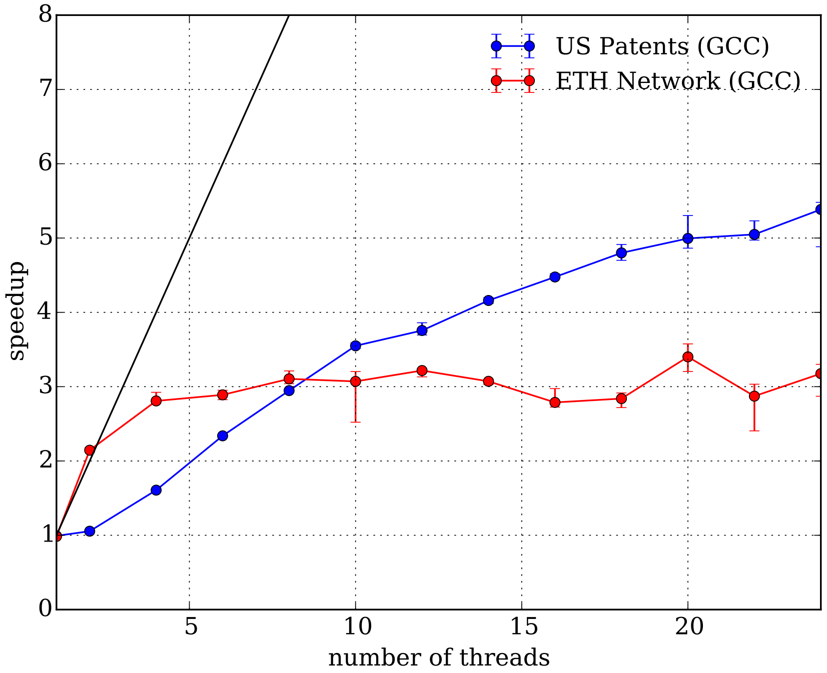

Figure 1: Strong scaling for ETH network and US patents. Data obtained always using a full exclusive node with GCC 4.8.2 (median and 20-40 percentiles for 10 runs)

Summary

Summarize your answers, results and plots into a short PDF document. Furthermore, elucidate the main structure of the code and report possible code details that are relevant in terms of accuracy or performance. Send the PDF document and source code to your assigned teaching assistant.