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`

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

[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

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`

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

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

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.

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