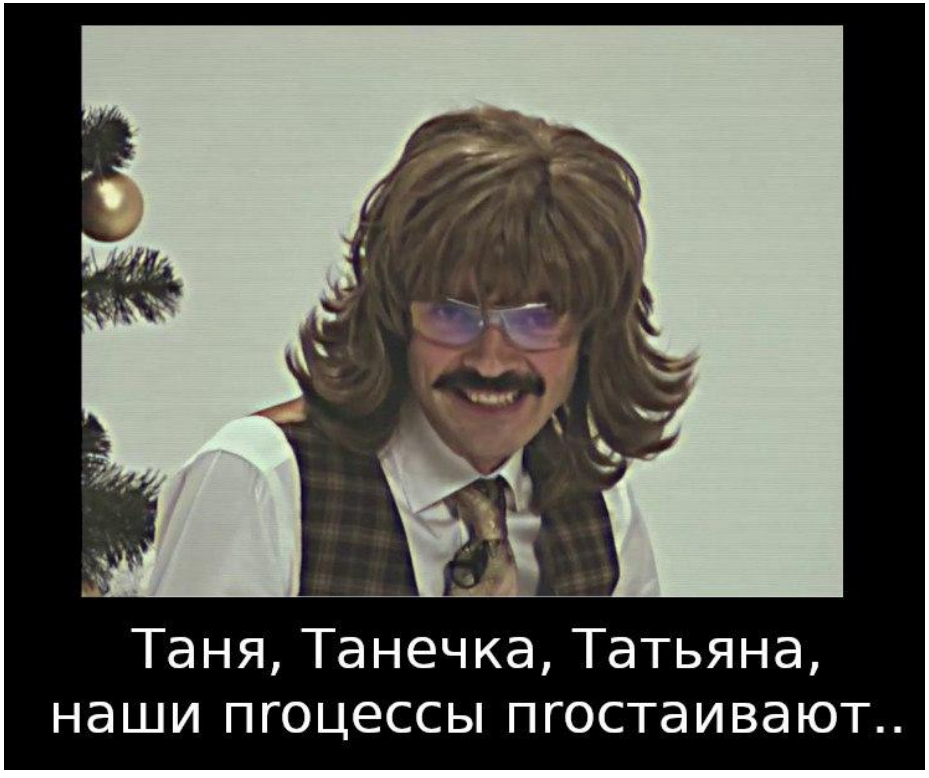
Лирическое отступление о планировщиках задач. Задачи с графами

Серегй Матвеев, Сергей Рыкованов

Plan for today

1. Difference between stand-alone server and computing cluster
2. Overview of workload schedulers and their logical setup (Torque PBS, SLURM etc).
   - General logic
   - Example: Torque PBS
   - Example: SLURM
   - Schedulers' commands
   - Practical implementation of the simplest SLURM script.
3. Pagerank algorithm
   - Ranking problem, graph of connections
   - Adjacency matrix, reminder on linear algerbra
   - Perron-Frobenius theorem, numerical method
   - Damping for generic case and some applications
   - Dynamical Pagerank
   - SimRank model
   - Hometask

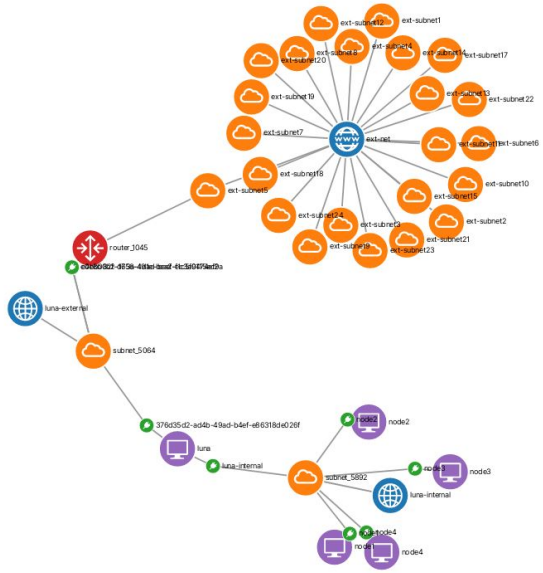

Таня, Танечка, Татьяна,
наши nгоцессы nгостаивают..

Our sample cloud platform for today

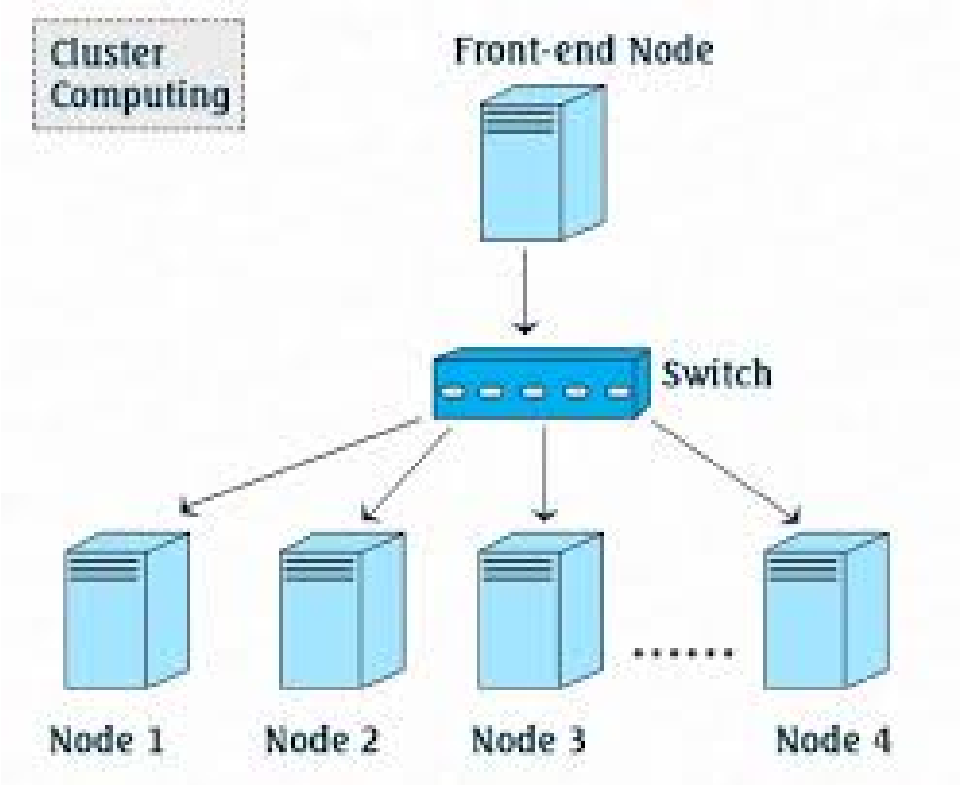
https://mcs.mail.ru

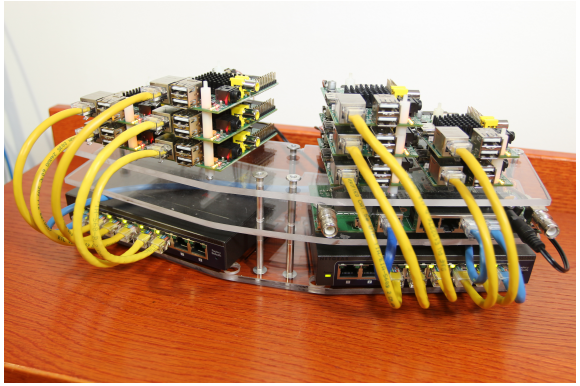
VK Cloud Solutions

Виртуальные машины	IP-адреса	Тип
node4	192.168.254.4	Standard-2-2 luna-cluster
node2	192.168.254.2	Basic-1-1 luna-cluster
node3	192.168.254.3	Standard-2-2 luna-cluster
node1	192.168.254.1	Basic-1-1 luna-cluster
luna	10.0.0.5 192.168.254.254 95.163.209.100	Basic-1-2 luna-cluster



Computing cluster organization





What may be available at your work?

- Ask your professor/collaborators/colleagues/supervisor (and ask one more time) for group resources. Quite many exploit their own servers/mini-clusters
- For example we have some at Skoltech/MSU/RAS available for collaborations...
- Submit requests to other academic places: Lomonosov MSU, Russian Academy of Science, MIPT
- Industry always have good resources, e.g. MailRu Group , Sber, MTS

Workload schedulers

In case of single server, resources are under control of OS and its scheduler. But at cluster computing system each node is server itself.
Hence it is necessary to have application controlling allocation of computing resources on higher level. What should it be able to do?

- Provide monitoring: availability and power status
- Allow user to allocate and release resources
- Avoid conflicts in process

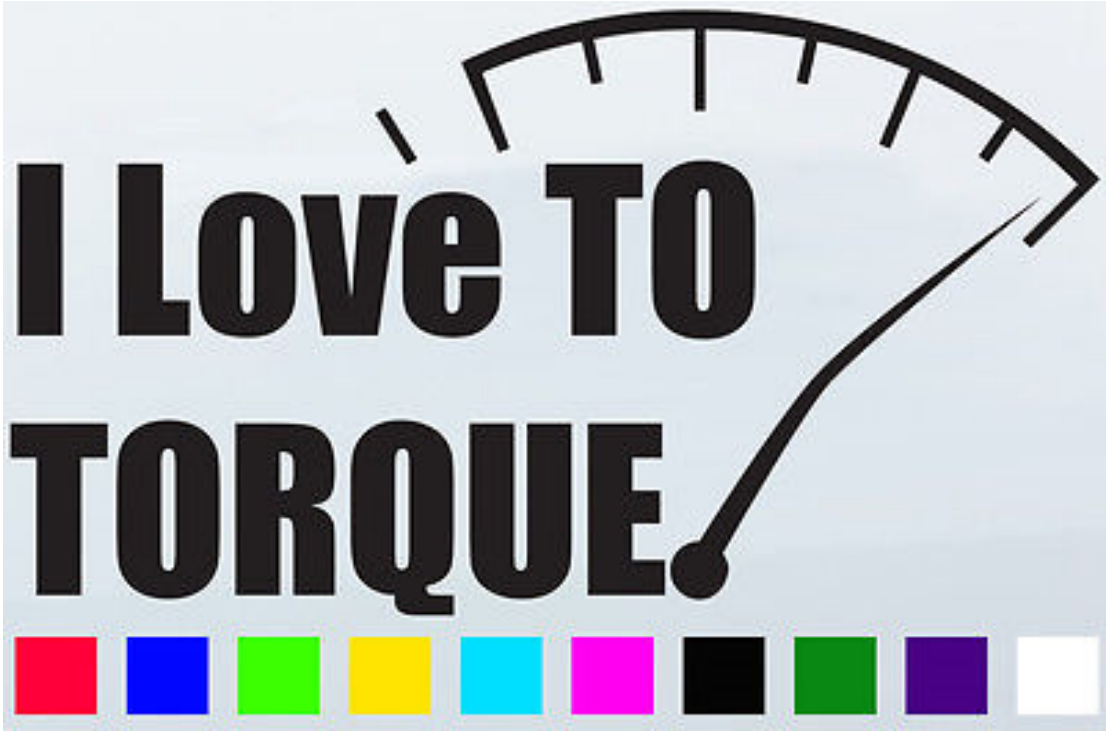
Go to
https://en.wikipedia.org/wiki/Comparison_of_cluster_software
for MORE information about schedulers. There are many...

Workload schedulers

Thus, logically we need commands which can be executed **at master node** and control state of **worker nodes**. What are they?

- Check system state and details about it: list of jobs, state of nodes, jobs distribution (**user level**)
- Submit and close jobs (**user level**)
- Priviledged commands for fix of collisions in system (**admin level**)

Example 1. Torque PBS



Example 1. Torque PBS

Terascale Open-Source Resource and QUEue Manager

- Check system state and details about it: **qstat**
- Submit and close jobs: **qsub**, **qdel**

Let's start from PBS.
Job submission can be done via **submission of bash-script** calling your program for execution.
You also have to give details about required resources for allocation.

Example 1. Torque PBS

- Trivial example of interactive PBS job submission:
  echo "sleep 10; echo helloworld"| qsub -N Helloworld -l nodes=1:ppn=1:pmem=2Gb
  As a result you will get files:
  Helloworld.e**JOB-NUMBER** – standard error stream
  Helloworld.o**JOB-NUMBER** – standard output stream
- Trivial example of PBS-script
  submit.qs
  #!/bin/bash
  #PBS -l nodes=8:ppn=4:mem=2Gb
  #PBS -l walltime=40:00
  echo helloworld
- qsub submit.qs

Example 2. SLURM

Example 2. SLURM

Simple Linux Utility for Resource Management

- Monitoring: **sinfo**, **squeue**
- Submit job/script: **srun**, **sbatch**. Close job: **scancel**
- Trivial example of interactive SLURM job submission: **srun echo Helloworld**
- Trivial example of SLURM-script
  submit.sl
  #!/bin/bash
  echo helloworld
- **sbatch -N 1 -c 16 -p MMM submit.sl**

Time to try hands-on and collect some measurements...

Environment modules

For typical choice of user environment definitions it is possible to get ready-made environment modules from system administrators

- module avail
- module load
- module list
- module unload
- module purge

Pagerank intro

A **ranking** is a relationship between a set of items such that, for any two items, the first is either 'ranked higher than', 'ranked lower than' or 'ranked equal to' the second. E.g.

- Credit ranking/rating
- FIFA ranking
- Ranking of students
- H-index
- **ranking of the websites**
- and other cases

One possible approach solving ranking problem is a famous Pagerank method co-authored by S. Brin and L. Page. Once upon a time those guys were PhD-students.[1]

[1] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Stanford InfoLab.

Simplified Pagerank

Naive ranking: rank with number of incoming links.
Better way is:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

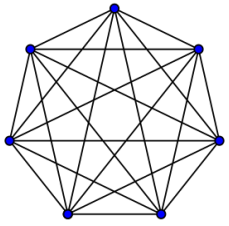
where

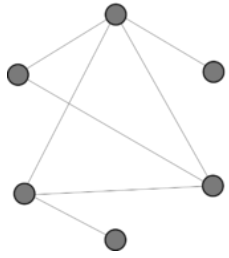
- B_u is the set containing all pages linking to page u
- $L(v)$ number of links from page v. Ok... also damping (suddenly user also can randomly go to any page):

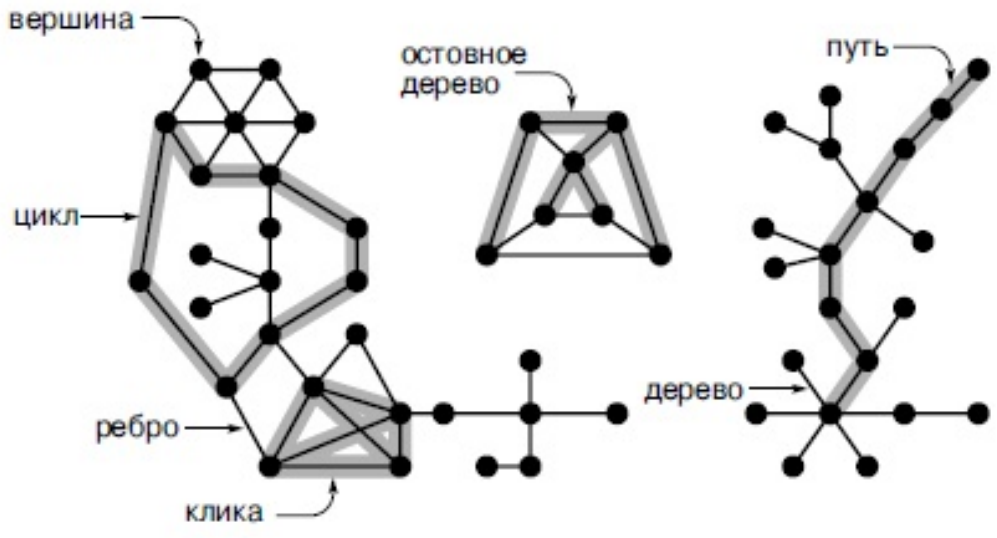
$$PR(u) = \frac{1-d}{N} + \frac{d}{N} \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

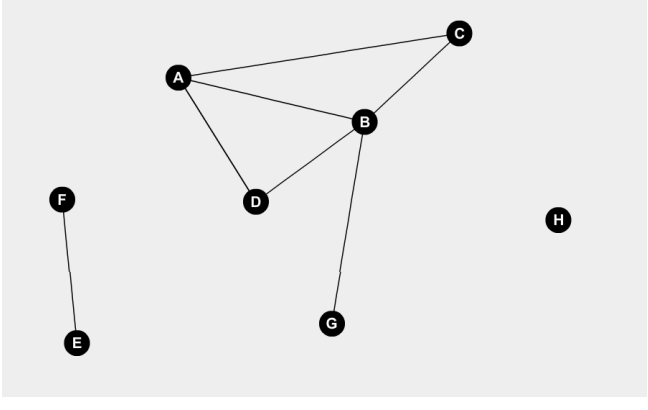
in WiKi-example $d = 0.85$ is said to be typical.

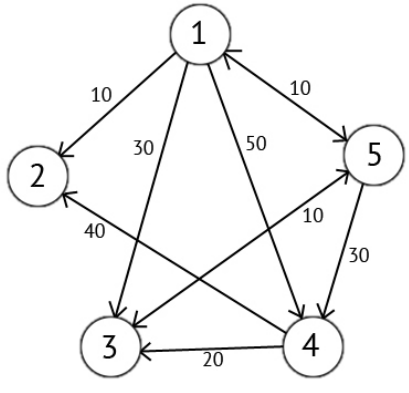
Graphs reminder











Graph and its adjacency matrix

Let G be a graph with n nodes (u_i) and some edges$(v_{i,j})$. Then one can generate $n \times n$ matrix A such that:

$$A_{i,j} = \begin{cases} w_{i,j} & \text{if } v_{i,j} \text{ exists} \\ 0 & \text{else} \end{cases}$$

Interesting facts:

- G_1 and G_2 equivalent if there exists permutation matrix P:

$$A_1 = P^{-1} \; A_2 \; P$$

- $(A^m)_{i,j} \neq 0$ equals number of paths with m edges from u_i to u_j (try matmul for some sample graphs).

Example for dummies (directed graph)

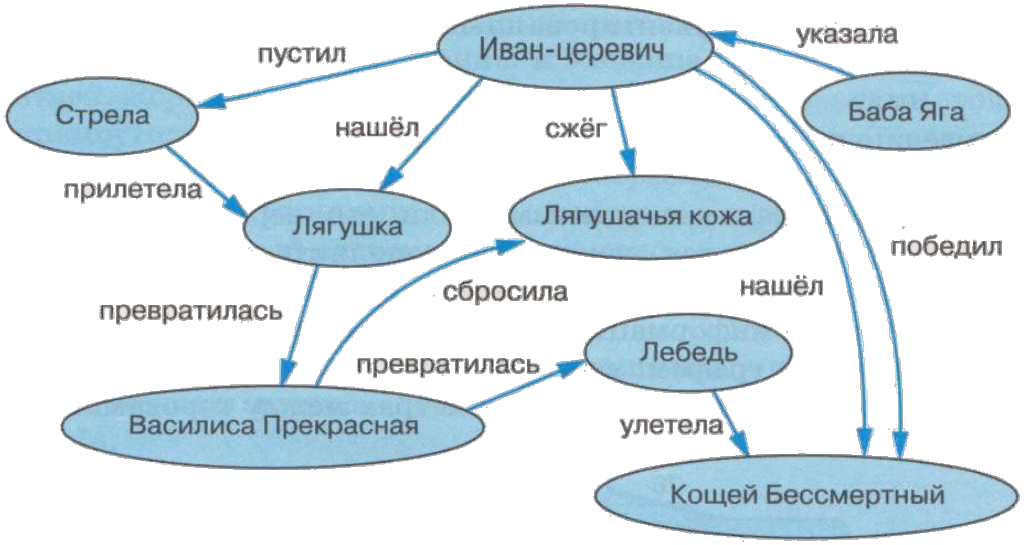


Рис. 47

$$A = \begin{array}{c|cccccccc} \text{Ivan} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \text{Strela} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{Lyagushka} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{Kozha} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \text{Yaga} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{Vasilisa} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \text{Lebed'} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \text{Koshey} & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

Example for dummies (directed graph)

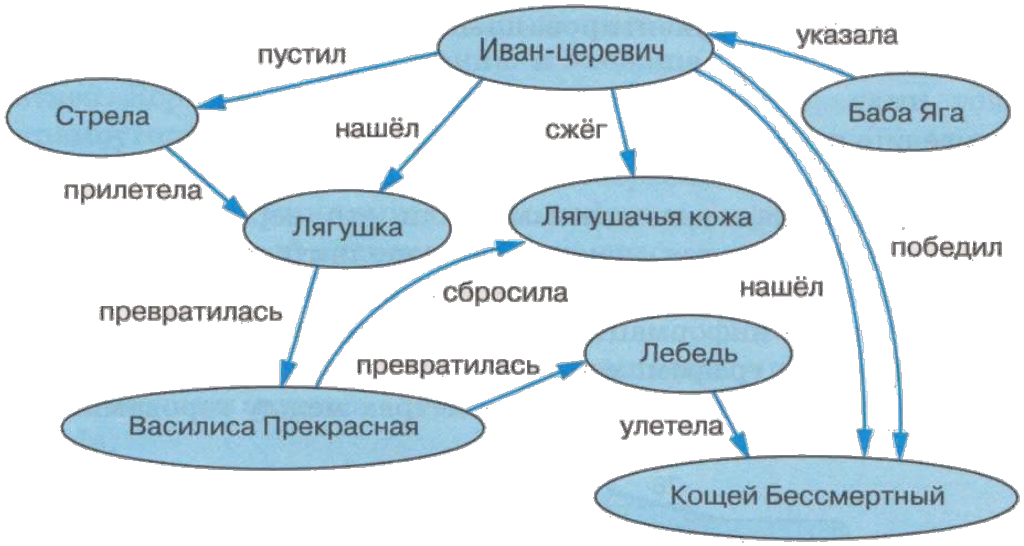


Рис. 47

$$A = \begin{array}{c|cccccccc} \text{Ivan} & 0.0 & 0 & 0 & 0 & 1 & 0.0 & 0 & 0 \\ \text{Strela} & 0.2 & 0 & 0 & 0 & 0 & 0.0 & 0 & 0 \\ \text{Lyagushka} & 0.2 & 1 & 0 & 0 & 0 & 0.0 & 0 & 0 \\ \text{Kozha} & 0.2 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ \text{Yaga} & 0.0 & 0 & 0 & 0 & 0 & 0.0 & 0 & 0 \\ \text{Vasilisa} & 0.0 & 0 & 1 & 0 & 0 & 0.0 & 0 & 0 \\ \text{Lebed'} & 0.0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ \text{Koshey} & 0.4 & 0 & 0 & 0 & 0 & 0.0 & 1 & 0 \end{array}$$

Some linear algebra...

v is an eigenvector of A with eigenvalue λ if

$$A \cdot v = \lambda \cdot v.$$

Perron-Frobenius theorem:
Let A be a matrix with all-positive real elements, then

- The largest eigenvalue r is real and positive
- It is a prime root of characteristic polynomial
- Corresponding eigenvector has all-positive entries (!)
- $\min \sum_j a_{i,j} \leq r \leq \max \sum_j a_{i,j}$

Very important theorem!
Let us normalize our matrix columnwise...

Irreducible matrix

- Matrix A $n \times n$ is irreducible if for some permutation matrix P:

$$P^T A P$$

  is block upper-triangular. It is equivalent to fact that
- the digraph associated to A is strongly connected.
- The Perron-Frobenius theory gives more information about irreducible matrices.

Some more information

We seek to have a funny property:

$$\mathbf{u} = [1, 1, \ldots 1]$$
$$u \cdot A = u$$

Why our sample matrix is bad???

Some more information

After normalization of columns of A we seek to have funny property:

$$\mathbf{u} = [1, 1, \ldots 1]$$
$$u \cdot A = u$$

Why our sample matrix is bad???
We need damping!
Remember:

$$PR(u) = \frac{1-d}{N} + \frac{d}{N} \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

Some more information

After normalization of columns of A we have funny property:

$$\mathbf{u} = [1, 1, \ldots 1]$$
$$u \cdot A = u$$

Why our sample matrix is bad???
We need damping [2]!
For matrices:

$$A \to \hat{A} = \frac{1-d}{N} \cdot \hat{E} + d \cdot A,$$

matrix $\hat{E}$ – elementwise ones.

[2] still does not help if our matrix has pure zero column/row

Fast matvec for low-rank matrix

$$\hat{E} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Fast matvec for low-rank matrix

Rank-1 matrix

$$\hat{E} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 1\ 1]^T [1\ 1\ 1\ 1\ 1]$$

Fast matvec for low-rank matrix

$$\hat{T} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

What is rank of this matrix?

$$\hat{T} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

What is rank of this matrix?

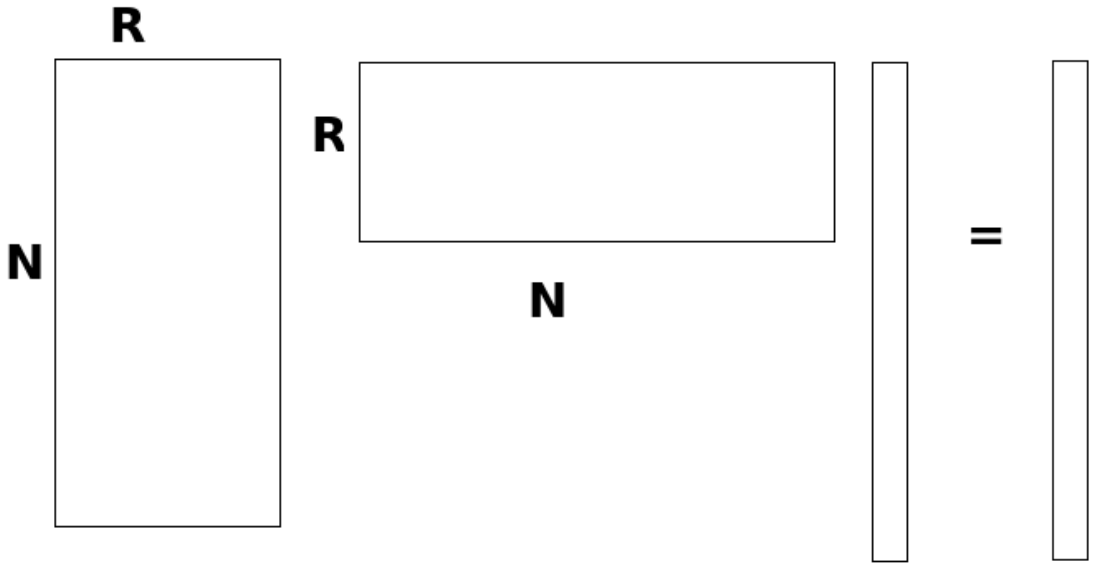
$$rank(T) = 2$$

. Rank-R matrix:

$$A_{i,j} = \sum_{\alpha=1}^{R} u_i \cdot v_j = U \cdot V$$

Fast matvec for low-rank matrix

$$A = U \cdot V^T.$$



- Singular Value Decomposition
- MaxVol principle and cross approximation (!)

Norms and distanses

- $\rho(\cdot, \cdot)$ – distanse is correspongs to some properties,
- $|| \cdot ||$ – norm, if corresponds to some more properies,
- $||x||_p = (\sum^i |x_i|^p)^{1/p}$ – Hölder norms, in particular
- $||x||_p$ – Euclidian norm (`cblas_dnrm2`)

Low-rank matrices and least squares problem

$$||Ax - b||_2 = ||U\Sigma V^T x - b||_2 = ||\Sigma y - U^T b||_2$$
$$y = \Sigma^\dagger U^T b$$
$$x = V\Sigma^\dagger U^T b.$$

More in numerical linear algebra courses...

Fixed-point iteration

$$A \cdot x_i = y_i,$$
$$x_{i+1} = \frac{y_i}{||y_i||}.$$

Examples of norms:

$$||x||_1 = \sum_i |x_i|,$$

and

$$||x||_2 = \sum_i |x_i|^2,$$

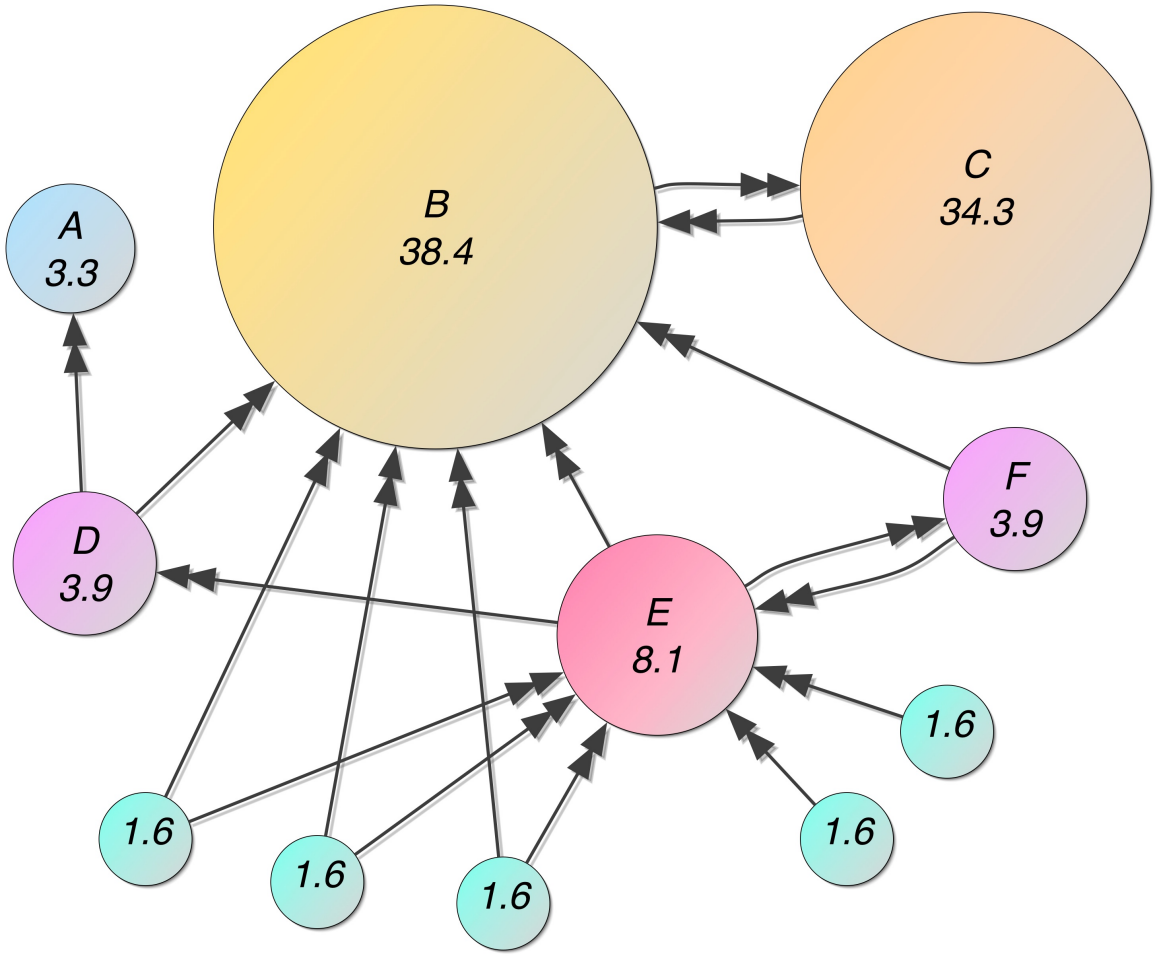
Stopping criteria:

$$||x_k - x_{k-1}|| \leq \varepsilon$$

Such sequence of iterations leads to desired result [3]

[3] See any book on Numerical Linear Algebra

Test case and discussion (see WiKi)



Page	B	E	A	D	C	F	G	H	I	J	K
Inputs	7	6	1	1	1	1	0	0	0	0	0

Ok, what about our example?

1. Koshey: 61.99 %
2. kozha: 14.32%
3. lebed': 14.01%
4. Vasilisa: 6.94%
5. Lyagushka: 1.85%
6. Ivan: 0.41%
7. Strela: 0.4%
8. Yaga: 0.08%

$$\varepsilon = 0.01$$

- Koshey: 59.86%
- kozha: 15.17%
- lebed: 14.87%
- Vasilisa: 7.43%
- Lyagushka: 1.85%
- Ivan: 0.37%
- Strela: 0.37%
- Yaga: 0.07%

$$\varepsilon = 10^{-5}$$

Sparse matrix

Dense Matrix

1	2	31	2	9	7	34	22	11	5
11	82	4	3	2	2	3	3	2	1
3	9	13	8	21	17	4	2	1	4
8	32	1	2	34	16	7	78	10	7
9	22	3	9	8	71	12	22	17	3
13	21	21	9	2	47	1	81	21	9
21	12	53	12	91	24	61	8	91	2
61	8	33	82	19	87	16	3	1	55
54	4	78	24	18	11	4	2	99	5
13	22	32	42	9	15	9	22	1	21

Sparse Matrix

1	.	3	.	9	.	3	.	.	.
11	.	4	.	.	.	.	.	2	1
.	.	1	.	.	.	4	.	1	.
8	.	.	.	3	1	.	.	.	.
.	.	.	9	.	.	1	.	17	.
13	21	.	9	2	47	1	81	21	9
.	.	.	.	.	.	.	.	.	.
.	.	.	.	19	8	16	.	.	55
54	4	.	.	.	11	.	.	.	.
.	.	2	.	.	.	.	22	.	21

- *ijk*-format
- CSR-format
- CSC-format

Compressed row format

$$A = \begin{pmatrix} 7.5 & 2.9 & 2.8 & 2.7 & 0 & 0 \\ 6.8 & 5.7 & 3.8 & 0 & 0 & 0 \\ 2.4 & 6.2 & 3.2 & 0 & 0 & 0 \\ 9.7 & 0 & 0 & 2.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.8 & 5.0 \\ 0 & 0 & 0 & 0 & 6.6 & 8.1 \end{pmatrix}$$

```
rowptr:  ( 0  4  7  10 12 14 16 )
colind:  ( 0  1  2  3  0  1  2  0  1  2  0  3  4  5  4  5 )
val:     ( 7.5 2.9 2.8 2.7 6.8 5.7 3.8 2.4 6.2 3.2 9.7 2.3 5.8 5.0 6.6 8.1 )
```

SpMV

```
// loop over rows
for (i=0; i<N; i++) {
  y[i] = 0;
  for (k=rowptr[i]; k<rowptr[i+1]; k++) {
    y[i] += val[k]*x[col[k]]; }
}
```

Dense Matrix

	0	1	2	3
0	a			
1		b		
2	c		d	
3	e		f	g

Sparse Matrix in CSR

rowptr	0	1	2	4	7		
col	0	1	0	2	0	2	3
val	a	b	c	d	e	f	g

Some libraries and soft...

- lapack and lapacke (see also intel MKL)
- umfpack, sparskit, eigen
- python-numpy, python-scipy

Extension: Pagerank with trends

Instead of usual PR, with random uniform teleportation weight ε for each vertex we assume that there is some distribution v_i (trends hypothesis).
Hence,

$$x^{k+1} = (1-\varepsilon)Px^k + \varepsilon v = \tilde{P}x^k,$$

where $\tilde{P} = (1-\varepsilon)P + \varepsilon v\mathbf{1}^T$.
Stationary weighted Pagerank: The x^* is stationary if $x^* = Px^*$. It is an eigenvector of P for the random motion. In case of PageRank x^* it satisfies an equation:

$$(I - (1-\varepsilon)P)x = \varepsilon v.$$

Extension: time-dependent Pagerank

Пусть $\Delta t = 1$:

$$x^{k+1} - x^k = (1-\varepsilon)Px^k + \varepsilon v - x^k = \varepsilon v - (I - (1-\varepsilon)P)x^k.$$

получается системка дифференциальных уравнений:

$$dx(t)/dt = \varepsilon v - (I - (1-\varepsilon)P)x(t)$$

с вектором "телепортаций", имеем более общее выражение:

$$x'(t) = \varepsilon v(t) - (I - (1-\varepsilon)P)x(t).$$

Аналитически:

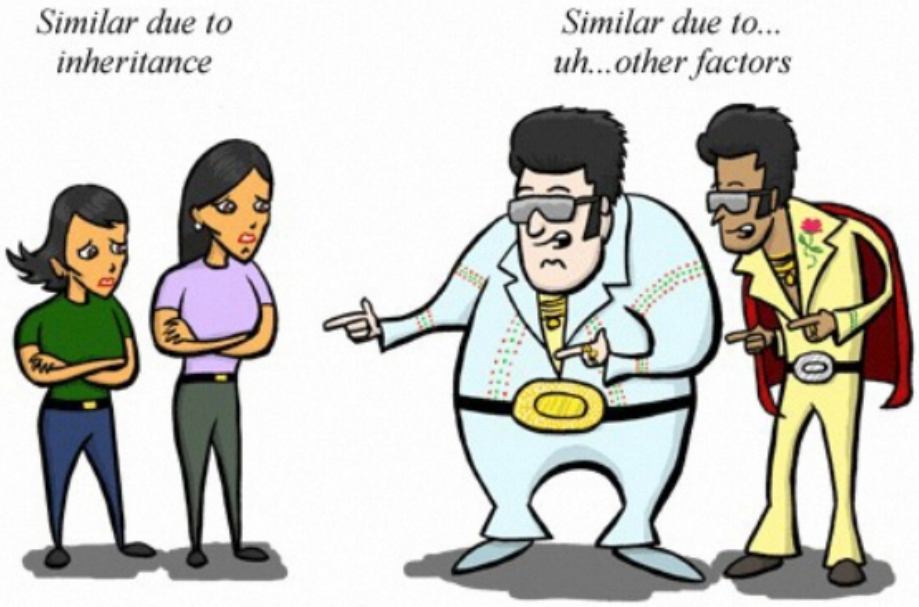
$$x(t) = \exp(-(I-(1-\varepsilon)P)t)x(0) + \varepsilon \int_0^t \exp(-(I-\alpha P)(t-\tau))v(\tau)d\tau.$$

Но можно и попроще через метод Эйлера:

$$x(t+h) = x(t) + h[\varepsilon v(t) - (I - (1-\varepsilon)P)x(t)]$$

Он устойчив при $h < \dfrac{2}{1+(1-\varepsilon)}$. Соответственно, при $h < 1$

Simrank: similar vertices



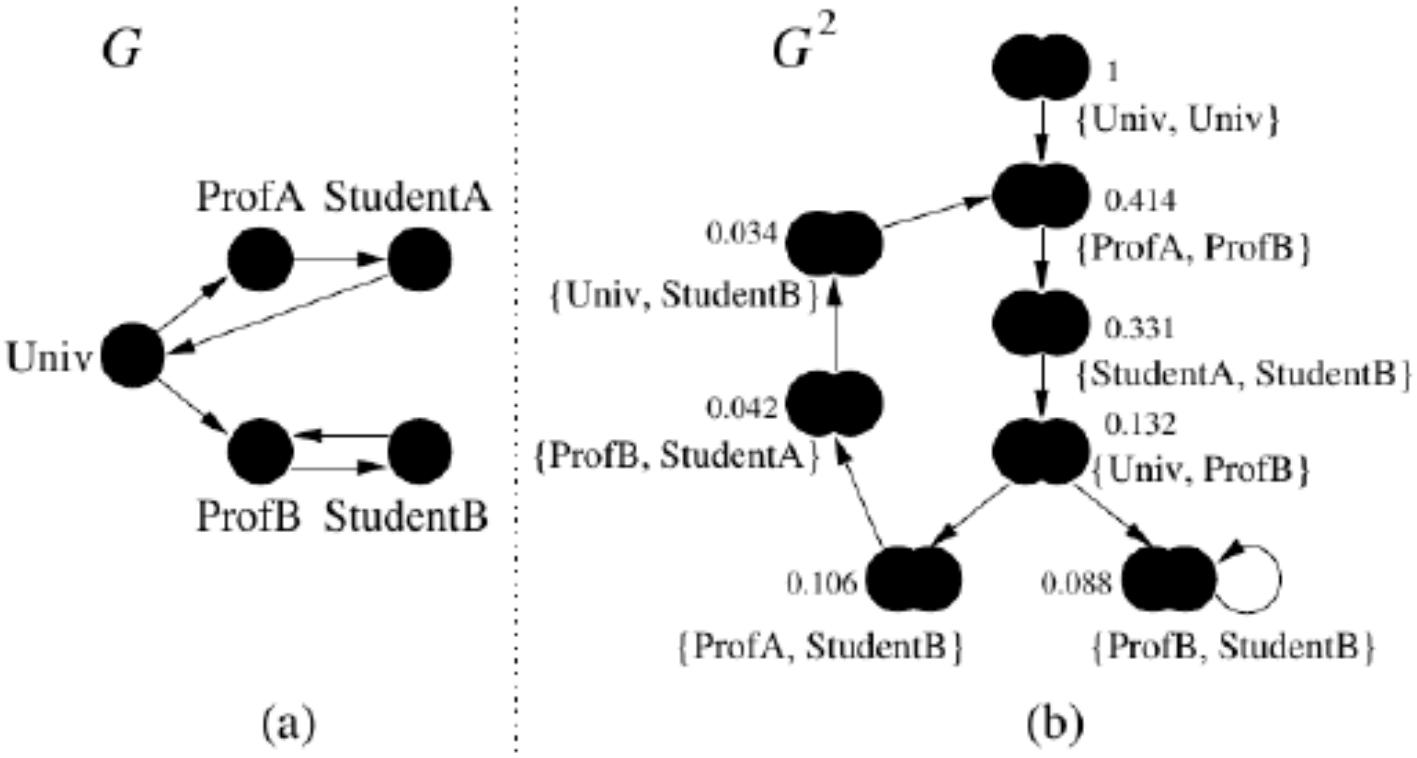
Guys are similar if they like the similar girl!
... and vice-versa!

Simrank: more equations

$$s(V_i, V_j) = \begin{cases} 1, & \text{если } V_i = V_j, \\ 0, & \text{если } |I_i| = 0 \text{ или } |I_j| = 0, \\ \dfrac{c}{|I_i| \cdot |I_j|} \cdot \sum\limits_{k \in I_i} \sum\limits_{l \in I_j} s(V_k, V_l), & \text{иначе,} \end{cases}$$

Формально это система линейных уравнений с N^2 неизвестными $s(V_i, V_j)$.
В лоб решать через метод Гаусса придётся за $O((N^2)^3) = O(N^6)$ действий. ДОЛГО!

Simrank: how it works



Simrank: matrix form

$$S = cA^\top SA - c \cdot diag\left(A^\top SA\right) + I.$$

Задача в матричных обозначениях сама приводит к итерациям

$$S_{k+1} = \mathbf{F}(S_k) = \underbrace{cA^\top S_k A - c \cdot diag\left(A^\top S_k A\right) + I}_{\mathbf{F(S)}}, S_0 = I.$$

Сложность итерации можно оценить в худшем случае через $O(N^3)$, но матрица связности – разреженная, этим тоже нужно пользоваться. Через CSR-формат!

Hometask

- Скачать и приготовить датасет Московского или Лондонского метро для решения стандартной задачи Pagerank. Вывести топ-5 станций подземки по такому ранжированию. (50 %)
- Применить метод Simrank к графу Московской подземки (30 %)
- Сгенерировать случайные разреженные графы и сравнить производительность реализаций Pagerank с CSR-форматом и без него. (20 %)