The Social Network of Java Classes
Power-law for Dummies

Diego Puppin

Institute for Information Sciences and Technology
Pisa, Italy

March 24, 2006

Outline

Why in English?

- Anybody speaking good French or German?
- Are these slides going to be on-line somewhere/someday?
- I need to get ready for SAC 2006 :-)

Quick Check-pointing

- Document-partitioning using the query logs
- Peer-to-peer discovery systems
- Scheduling on SUN machines
- Pretty sweet languages

...and then...

Hold on!

...and then...

Hold on!
Awesome findings on power laws and Java!

Outline

Document-partitioning using the query logs

- WHY: Started as a problem related to the scalability of the discovery system
- WHAT FOR: General applicability to Web search engine, Big research potential
- WHAT: We identified a novel approach to partition documents using query logs, so to minimize the number of queried partitions
- To be presented at INFOSCALE 2006, raised the interest of Prof. Ricard Baeza-Yates

Document-partitioning using the query logs

- WHY: Started as a problem related to the scalability of the discovery system
- WHAT FOR: General applicability to Web search engine, Big research potential
- WHAT: We identified a novel approach to partition documents using query logs, so to minimize the number of queried partitions
- To be presented at INFOSCALE 2006, raised the interest of Prof. Ricard Baeza-Yates
- and of Dr. Fabrizio Silvestri... I almost forgot

Peer-to-peer discovery systems

- WHAT FOR: In the context of XtreemOS
- WHAT: Distributed solutions for an information service
- We have one “tirocinio” going on (Thx Hanien)
- Part of the syllabus of the class Topic in Grid Computing at the Indian Institute of Science, Bangalore: http://www.serc.iisc.ernet.in/~vss/courses/GC2005/

Scheduling on SUN machines

- WHY: My MS thesis at MIT does not want to die
- WHAT FOR: Maybe we can show it off to SUN
- WHAT: So far, we have one student “tirocinio” and one student “thesis”, plus Marco Pasquali

Sweet sausages

Sweet sausages

I meant... languages.

Sweet sausages

I meant... languages.

- WHY: different ways of doing “components”
- WHAT: I studied Aspect Programming and Design-by-Contract
- Along the way, Python and Ruby (pretty sweet)
- WHAT FOR:

Sweet sausages

I meant... languages.

- WHY: different ways of doing “components”
- WHAT: I studied Aspect Programming and Design-by-Contract
- Along the way, Python and Ruby (pretty sweet)
- WHAT FOR:

When all you have got is a hammer, everything looks like a nail

If you want a full serving of these appetizers... just ask!

Outline

Social Networks

- Behavior emerging from linked entities
- Ranging from biology, to computer science, to economics
- First studies in late 1930s
- Then, studies on random graph by Erdos
- Recently, good summary by Barabazi

Emerging Behavior

- *Rich* Web pages getting *richer*
- Stocks raising or falling quickly
- Ants building nests

Emerging Behavior

- *Rich* Web pages getting *richer*
- Stocks raising or falling quickly
- Ants building nests
- Traffic jams

Emerging Behavior

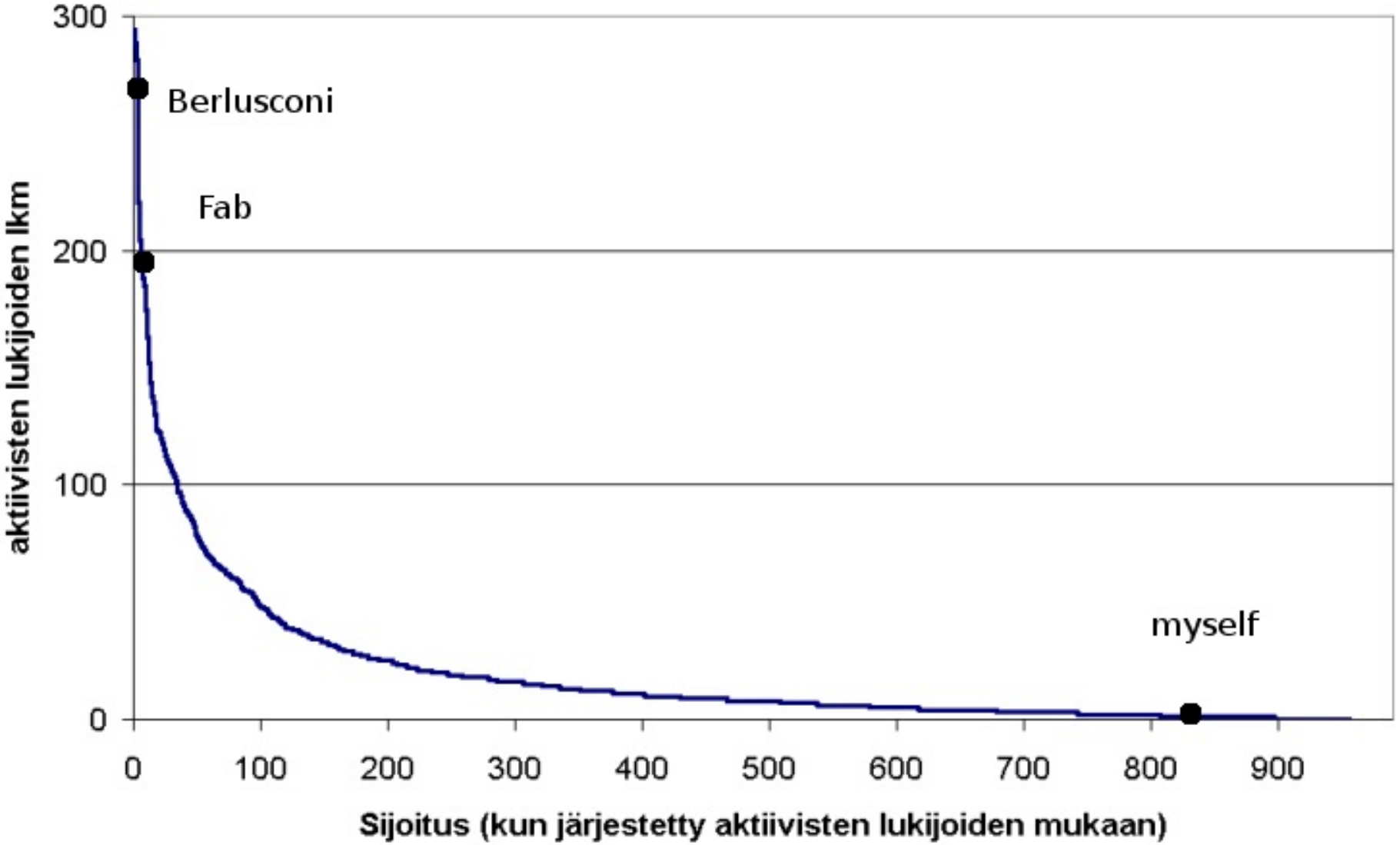
- *Rich* Web pages getting *richer*
- Stocks raising or falling quickly
- Ants building nests
- Traffic jams

When counting links, visitors, references in a natural network, large events are rare, but small ones are quite common.

Zipf, Pareto, Power Law

- Zipf's law: the 'size' y of an occurrence of an event is inversely proportional to its 'rank': $y \propto r^{-b}$, with b close to unity.
- Pareto: the number of events larger than x is an inverse power of x: $P[X > x] \propto x^{-k}$.
- Power law: the number of events equal to x: $P[X = x] \propto x^{-(k+1)} = x^{-a}$.

Also called scale-free distribution, because the distribution doesn't change with scale.

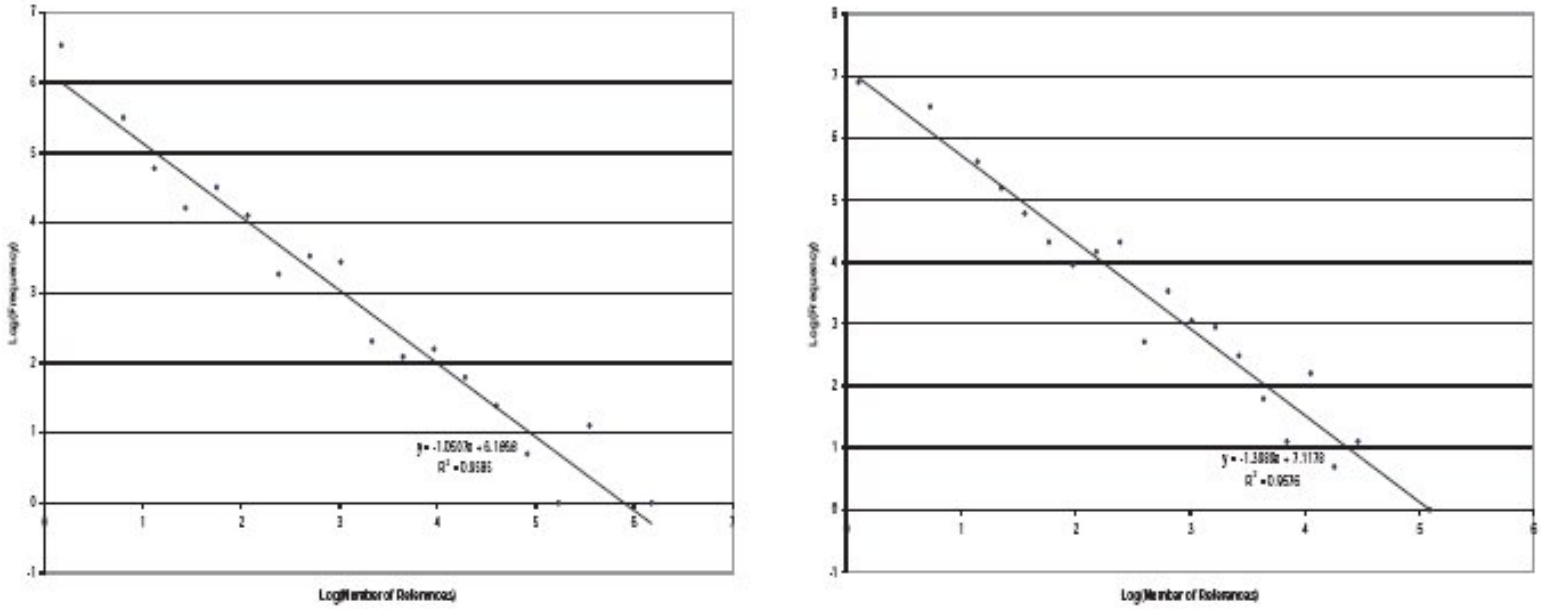

300
200
100
0
aktiivisten lukijoiden lkm
Berlusconi
Fab
myself
0
100
200
300
400
500
600
700
800
900
Sijoitus (kun järjestetty aktiivisten lukijoiden mukaan)

...may I have some more?

- Read Barabasi's "Linked" (in our library).
- Seminal paper by Faloutsos (x3) (1999).

Power-Law from Engineering Choices

From “Power Law Distribution in Class Relationships”. Interesting power-laws were found in the design of the Java Development Kit. Coherent SW engineering choices (modularity, orthogonality, interface contracts) lead to a scale-free network with power-law distributions: number of sub-classes, number of fields, number of methods, number of constructors.
Also in “Scale-free Networks from Optimal Design”.



(a) Field members

(b) Containing classes

Figure 7. Log-Log plots showing power law distributions in (a) the number of classes referenced as field variables and (b) in the number of classes which contain references to classes as field variables.

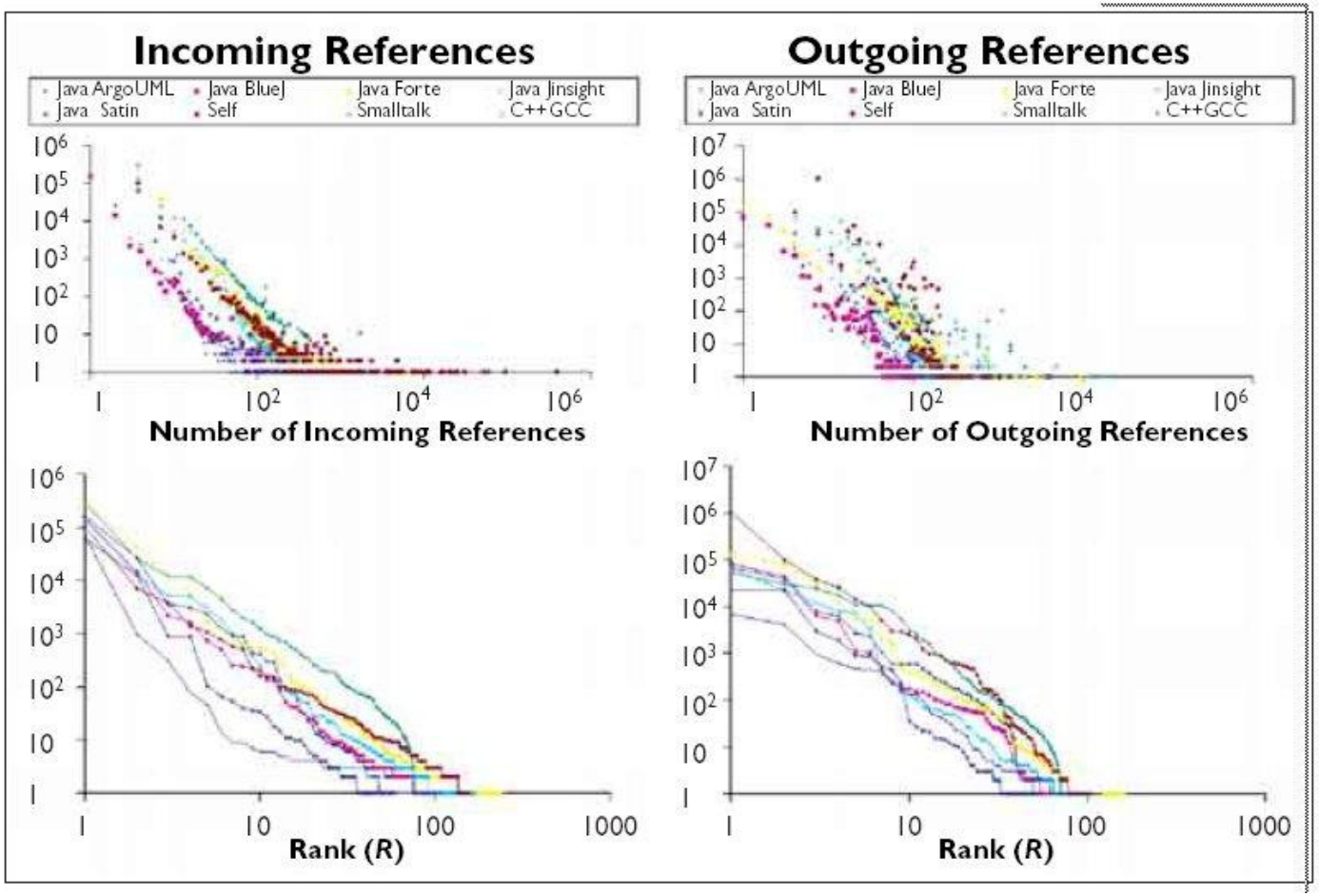
Power-Law from Social Behavior

We want to study the behavior that is *naturally emerging* from the social network, at the level of *static* references.

Scale Free Geometry in OO Programs

- Authors get a snapshot of the memory at run time
- They counts the number of references (link) among objects
- Number of inlinks and outlinks are distributed with power law

Scale Free Geometry in OO Programs



Scale Free Geometry in OO Programs

- No typical size of an object
- Few objects are very important
- Not CLASS, but object
- Interesting, but not useful for developers
- Dynamic analysis of code:
  - Interesting results from dynamic analysis (profiling, optimization) [GGMS03, vdAB03]
  - BUT dynamic data can be commercial secrets

Good for ACM

”Clustering Object-Oriented Software Systems using Spectral Graph Partitioning”, by Spiros Xanthos, second place at the 2005 ACM Student Research Competition.

Outline

How to Rank Software

- Similar to Google PageRank
- Static analysis of code links
- INTERFACES ONLY
- Every time a class is used, there is a rank boost
- No source code is needed
- No runtime information is needed
- Only public interfaces

Why only interfaces?

- Commercial component will hide source code
- Will also hide runtime profile information
- Interfaces must be public, in order to use a component
- We suggest to base ranking on this
- A composed application should also make public the composition structure
- use/provide ports

Why composition should be public?

- To improve trust
- To become more popular
- To support a standard
- Open Source...
  - Compare with digital libraries
  - bib. references VS full-text

Class Graph: Static vs Dynamic

- Abstract static information $\rightarrow$ Progr. Interface:Class Rank
- Executable information $\rightarrow$ Program Code:Comp. Rank
- Dynamic information $\rightarrow$ Process:Scale Free

A Model for Ranking

Assumptions:

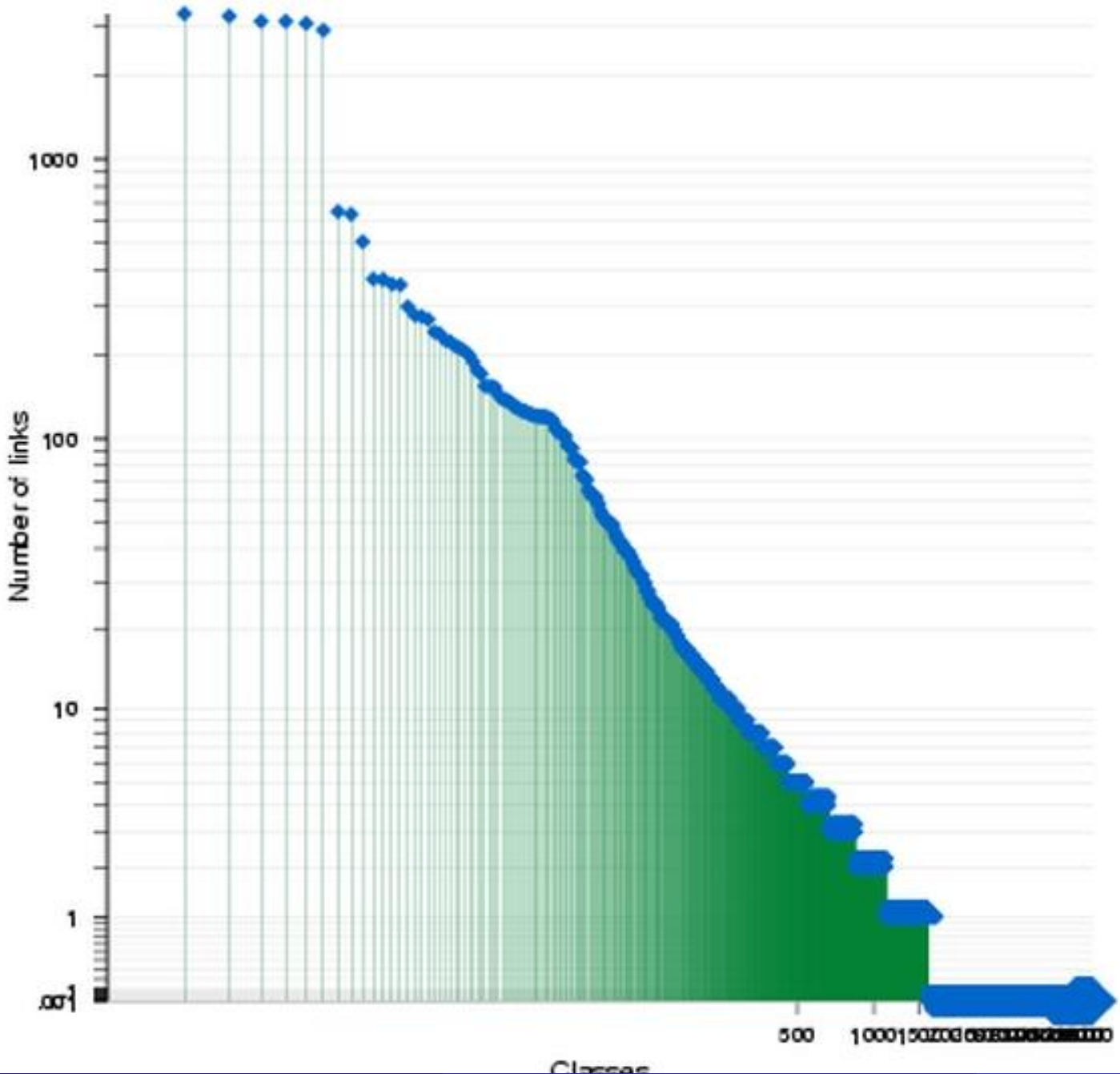
- Avoid ontologies
  - An ontology for the whole Grid?
- Heavy emphasis on LINKS
  - Positive experience on the Web
- No use of source code and dynamic data
  - Hidden in commercial apps
- Use only public interfaces
  - Maybe semantic info can be used
    - Sub/super-types...

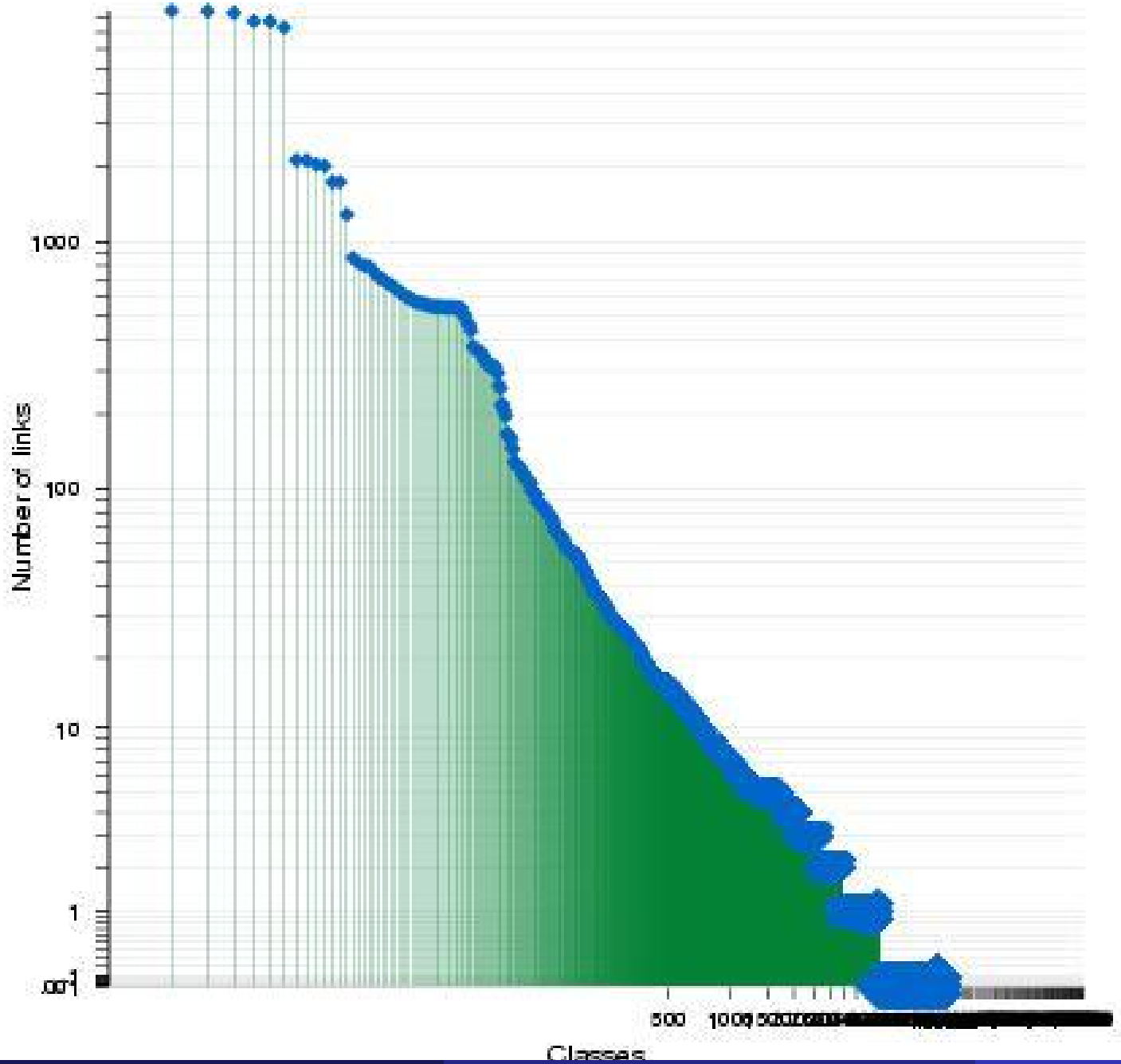
Initial Experiments

- Java classes, simple composition model
- Strong documentation (Java Docs)
- Unique ID (Package names)
- Class links are very easy to see
- We collected 49000 classes
- We parsed and built a *social graph*

Social Network in Java Classes

- Java programmer $\leftrightarrow$ component user
  - S/he chooses most general and useful classes
- Power-law behavior
  - Web pages, blogs, social networks sociali etc
  - see On Power-Law Relationships of the Internet Topology, by Faloutsos et al.
- component usage $\leftrightarrow$ Web linking!!!
- Component search $\leftrightarrow$ Web Search


1000
100
10
1
Number of links
500


Number of links
1000
100
10
1
500

Class Rank

To determine the rank of a class C, we iterate the following formula:

$$rank_C = \lambda + (1 - \lambda) \sum_{i \in inlinks_C} \frac{rank_i}{\#outlinks_i}$$

where $inlinks_C$ is the set of classes that use C (with a link into C), $\#outlinks_i$ is the number of classes used by i (number of links out of i), and λ a small factor, usually around 0.15.

Top-ranking classes

- String, Object, Class, Exception
- #7: Apache MessageResources
- #11: Tomcat CharChunk
- #14: DBXML Value
- #73: JXTA ID

GRIDLE 0.1

- Ranking using two metrics:
  - TF.IDF (term frequency times inverted document frequency)
  - GRIDLE Rank
- Bells and whistles:
  - Snippets, Links and Reverse Links
- http://gridle.isti.cnr.it

A search engine for SW components

GRIDLE: Google-like Ranking, Indexing and Discovery service for a Link-based Eco-system of software components

http://gridle.isti.cnr.it/?query=file+buffer&numero=10&sort=pagerank&submit=submit

Find high-relevance Java classes out of a repository of 7700 elements!!!

QUERY: file buffer RESULTS: 10

Sort by Class Rank or TF.IDF GO

SORTED BY CLASS RANK

PrintWriter (Java 2 Platform SE 5.0)
Class Path: java.io.PrintWriter

... The output will be written to the **file** and is **buffer**ed.csn - The name of a supportedHREF="../../java/nio/charset/Charset.html" title="class in java.nio.charset">charsetThrows:HREF="../../java/io/ **file**NotFoundException.html" title="class in java.io"> **file**NotFoundException - If the given string does not denote an existing, writable regular **file** and a new regular **file** of that name cannot be created, or if some ...
http://java.sun.com/j2se/1.4.2/docs/api/java/io/PrintWriter.html
Score: 35.75 - Cached copy - Class Graph

PrintStream (Java 2 Platform SE 5.0)
Class Path: java.io.PrintStream

... The output will be written to the **file** and is **buffer**ed.csn - The name of a supportedHREF="../../java/nio/charset/Charset.html" title="class in java.nio.charset">charsetThrows:HREF="../../java/io/ **file**NotFoundException.html" title="class in java.io"> **file**NotFoundException - If the given **file** object does not denote an existing, writable regular **file** and a new regular **file** of that name cannot be created, or if ...
http://java.sun.com/j2se/1.4.2/docs/api/java/io/PrintStream.html
Score: 35.50 - Cached copy - Class Graph

File (Java 2 Platform SE 5.0)

Outline

Conclusions

- Usage of Java classes follows the pattern of links among Web pages
- We can rank classes according to this finding
- ...towards a search engine for software

...going to SAC 2006!

Please, give me feedback!!!
Thanks for coming!!

Acknowledgments

- MIUR CNR Strategic Project L 499/97-2000 (5%)
- NextGrid
- CoreGRID
- Università degli Studi di Pisa
- ISTI-CNR