Comparison between SLOCs and number of files as size metrics for software evolution analysis

Israel Herraiz, Gregorio Robles, Jesús M. González-Barahona,
Universidad Rey Juan Carlos
{herraiz, grex, jgb}@gsyc.escet.urjc.es

Andrea Capiluppi, University of Lincoln
acapiluppi@lincoln.ac.uk

Juan F. Ramil, The Open University
J.F.Ramil@open.ac.uk

CSMR 2006, Bari, March 23rd 2006

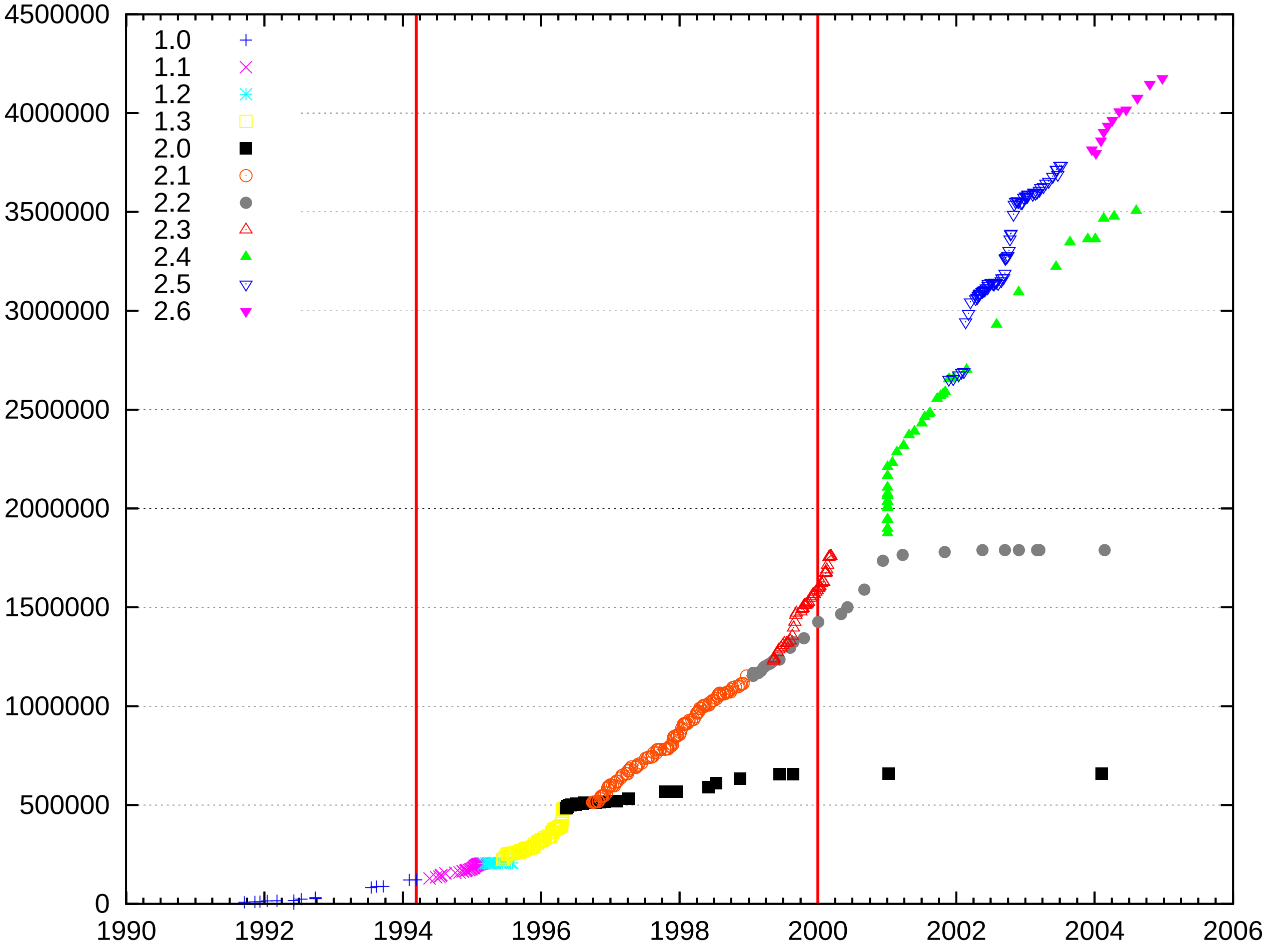
Introduction

- Libre (free / open source) software systems can grow superlinearly, not conforming to Lehman's laws.
- Classical studies use number of modules (source code files) as base.
- Recent studies on libre software, SLOCs.

Goals

- Is it superlinear (or at least, non-sublinear) a common pattern in the libre software world?
- Does the results change using different metrics?
- In the long term, why are they growing faster than expected?

An example: the case of the Linux kernel



Selected population

Debian GNU/Linux distribution

- There are some studies on the size and evolution of this distribution.
- There are a list of packages sorted by size.
- Map from packages to projects.

Sample selection criteria

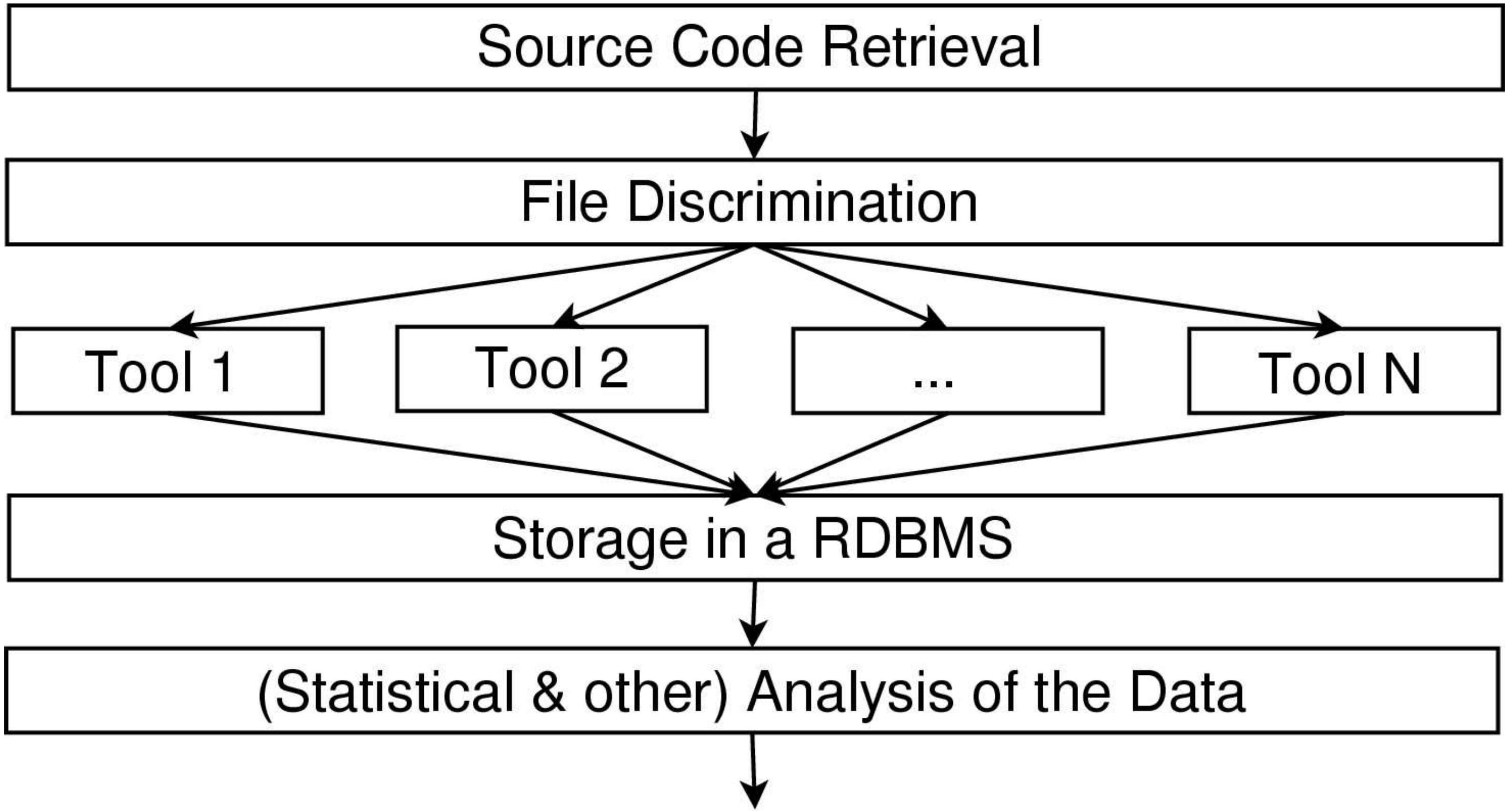
- 30 months old projects.
- Control version system publicly available.
- The largest projects of Debian.

Details of the selected projects

Name	Modules	SLOCs	Num. of files	First date	Last Date
Amaya	`Amaya`	443295	1092	1997-06-30	2005-05-19
Evolution	`evolution`	359148	1489	1997-12-27	2005-05-19
FreeBSD	`src`	1756198	5652	1993-06-19	2005-05-17
Kaffe	`kaffe`	598387	5294	1998-04-30	2005-05-23
NetBSD	`src`	2535613	13175	1992-07-19	2005-06-11
OpenBSD	`src`	1606210	6254	1997-10-26	2005-06-16
Prc-Tools	`prc-tools`	19176	152	2000-05-19	2005-05-25
Python	`python`	615113	2261	1997-06-30	2005-05-19
Wine	`wine`	927415	2098	1998-12-27	2005-05-24
wxWidgets	`wxwidgets`	1781581	4175	2002-11-04	2005-05-23
XEmacs	`XEmacs`	2210705	5266	1997-12-27	2005-05-19
XFree86	`xc`	2200501	6756	1997-06-30	2005-05-19
Linux	–	4176875	16583	1991-09-17	2004-12-24

Data extraction and analysis

By means of the GlueTheos and SlocCount tools, we obtained a snapshot of the source code each six months.



Results

- High correlation between SLOCs and number of files.
- 10 out of 13 projects show non-sublinear growth.

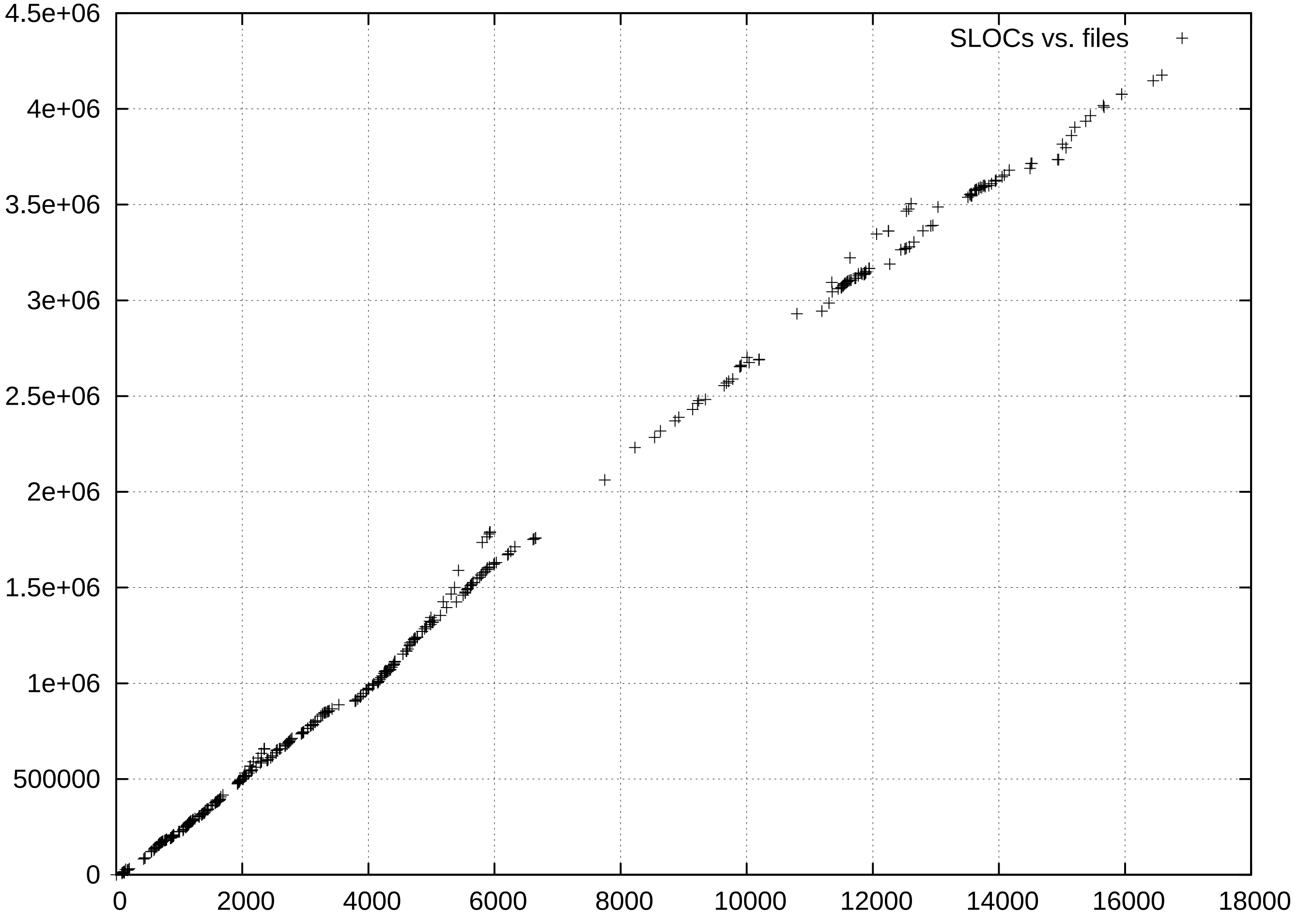
Correlation coefficients

Project	Equation	n	r^2
Amaya	$S = 563.635 \cdot f - 153662.2706$	14	.8893
Evolution	$S = 237.8748 \cdot f - 1397.7322$	11	.9945
FreeBSD	$S = 312.2457 \cdot f + 108.9p28$	46	.9975
Kaffe	$S = 105.7720 \cdot f - 5316.4131$	35	.9842
NetBSD	$S = 189.8916 \cdot f + 101416.3262$	43	.9889
OpenBSD	$S = 219.8261 \cdot f + 230075.6100$	12	.9797
Prc tools	$S = 101.2863 \cdot f + 2347.8471$	122	.9709
Python	$S = 335.8650 \cdot f - 189998.4555$	13	.9552
Wine	$S = 514.9345 \cdot f - 142849.9740$	11	.9927
wxWidgets	$S = 465.9170 \cdot f - 233395.2761$	29	.8950
XEmacs	$S = 424.4406 \cdot f - 26183.4515$	15	.9956
XFree86	$S = 326.6894 \cdot f - 70754.4803$	12	.9861
Linux	$S = 265.3134 \cdot f - 26979.9934$	533	.9977

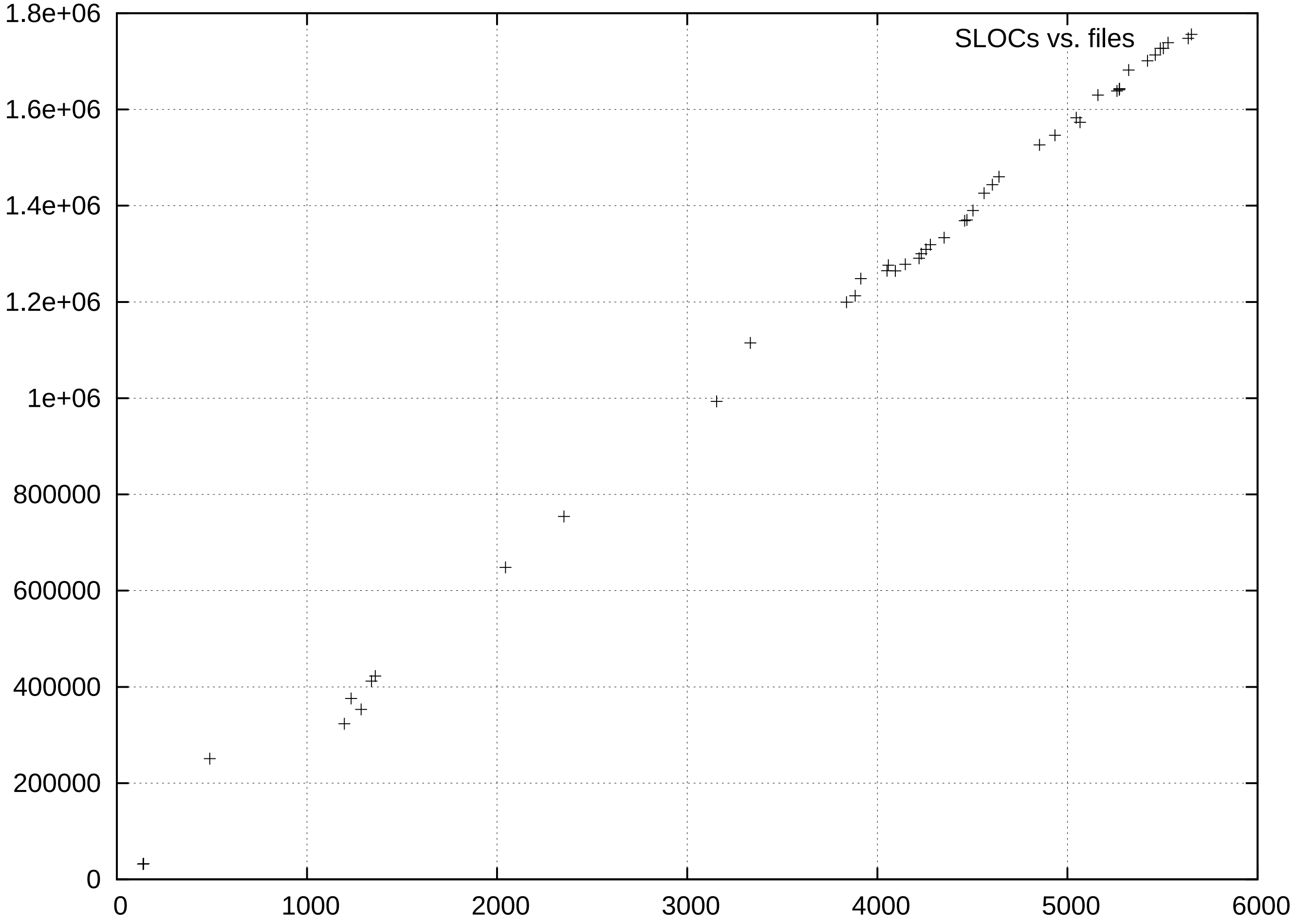
Growth patterns

Project	Growth rate (SLOCs)	Growth rate (# files)	Category
Amaya	1.45	-0.0055	Linear
Evolution	-31.89	-0.17	Sublinear
FreeBSD*	15.16	0.056	Linear
Kaffe	77.13	0.71	Superlinear
NetBSD*	152.74	1.04	Superlinear
OpenBSD*	401.20	2.01	Superlinear
Prc tools	4.31	0.044	Superlinear
Python	18.43	-0.062	Linear
Wine	50.06	0.064	Linear
wxWidgets*	587.56	0.29	Superlinear
XEmacs	-259.44	-0.60	Sublinear
XFree86	-412.28	-1.47	Sublinear
Linux*	186.21	0.71	Superlinear

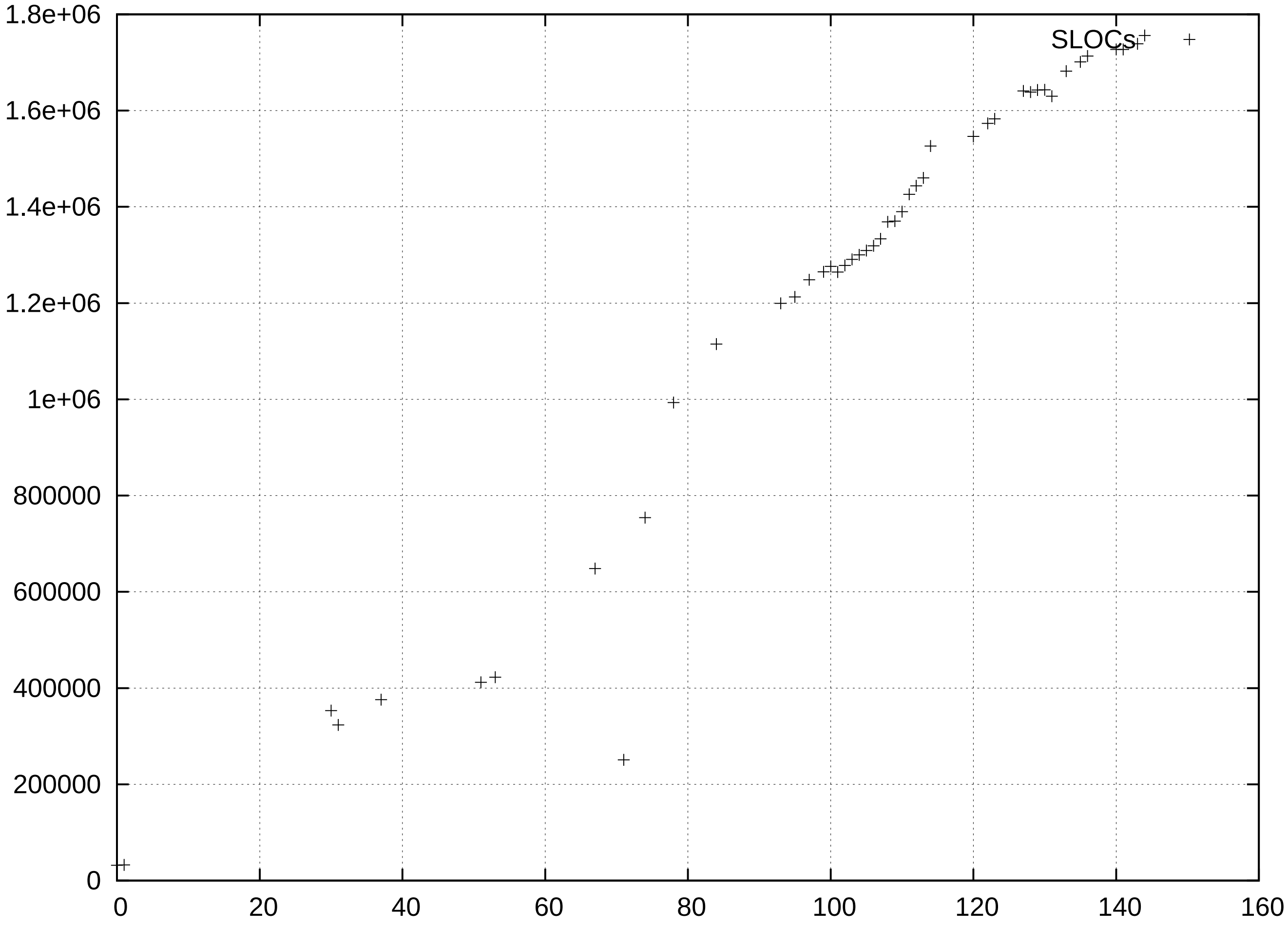
Correlation SLOCs vs Files in the case of Linux



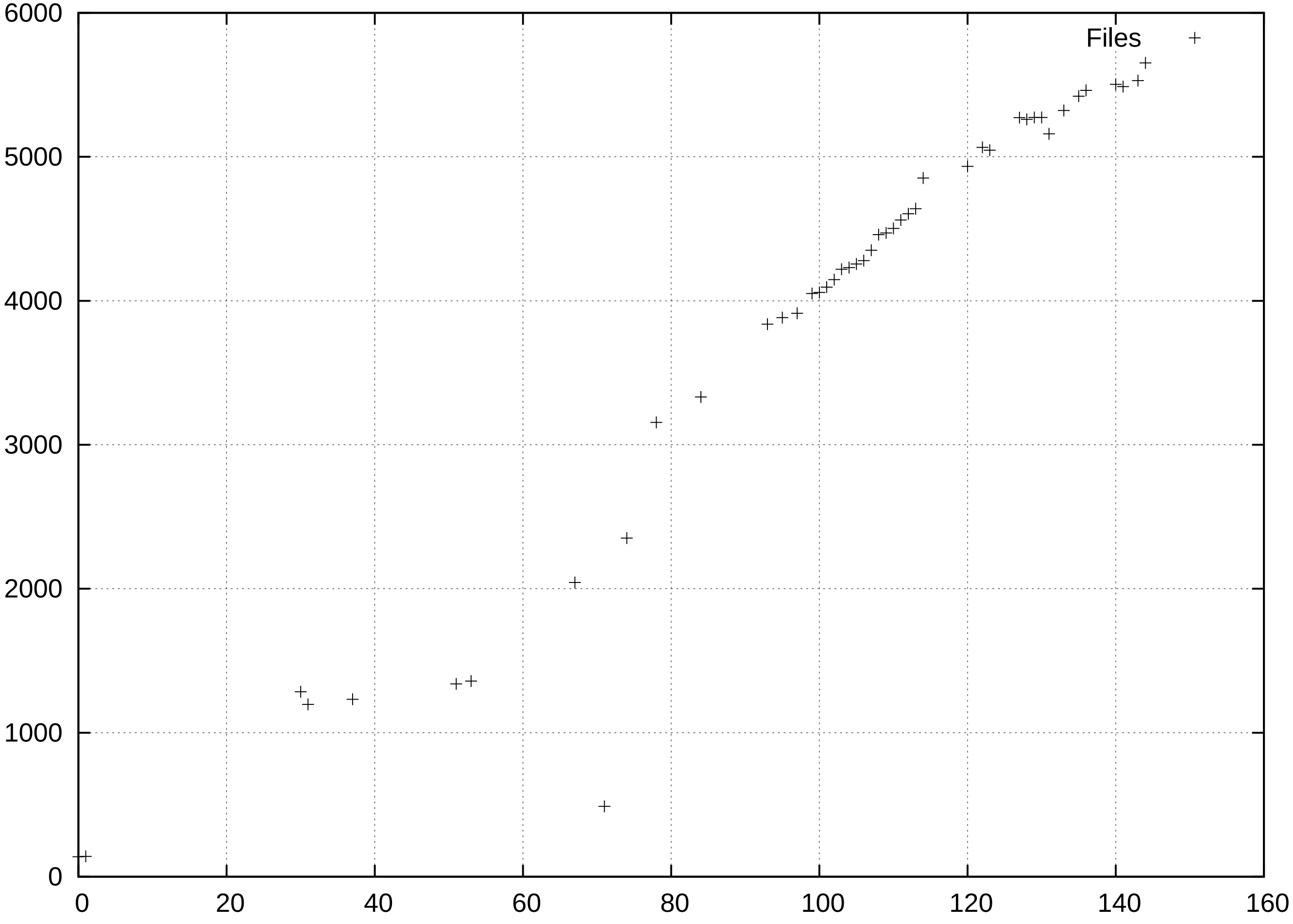
Correlation SLOCs vs Files in the case of FreeBSD



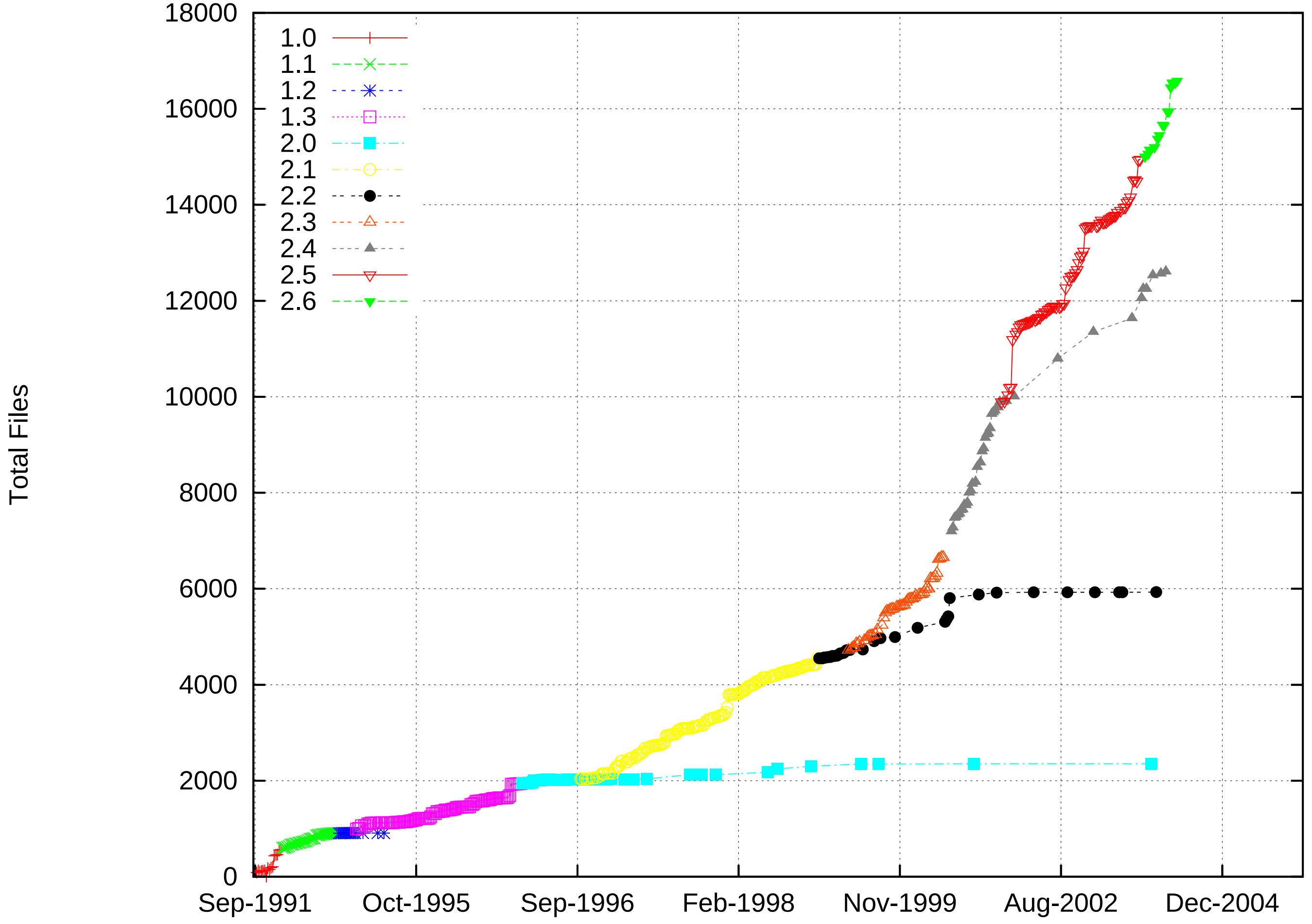
Evolution of the size of FreeBSD (SLOCs)



Evolution of the size of FreeBSD (Num. of files)



Evolution of the size of Linux (Num. of files)



Conclusions

- Most of the projects grow linear or superlinearly, not conforming to Lehman's laws.
- All the studied projects show the same pattern (sublinear, linear or superlinear), independently of the selected metric
- So, classical studies and more recent ones are comparable.
- Studies on the case of the Linux kernel that used SLOCs, are also validated using number of source code files.

Next steps

- To compare more metrics, in particular, syntactic with semantic metrics. Is it worthy to measure semantic metrics for the analysis of evolution of libre software?
- Effort and complexity measurements can help to explain superlinearity.

Many thanks for your attention

Reach me at `israel.herraiz@urjc.es`**.**

This presentation, some tools, and more papers on this topic available on `http://libresoft.urjc.es`**.**

Questions? Comments?