

---

# Applying Social Network Analysis to the Information in CVS Repositories

*Luis López-Fernández, Gregorio Robles,  
Jesús M. González Barahona,  
GSyC, Universidad Rey Juan Carlos, Madrid, Spain  
{llopez,grex,jgb}@gsync.escet.urjc.es*



*MSR 2004 (Edinburgh, UK)  
25th May 2004*

---

## Background

- There is a lot of (too much?) information about libre software projects out there
- We're starting to streamline the extraction of raw data (e.g., from CVS repositories)
- We have to apply data mining and data interpretation techniques to get meaningful information
- Let's explore approaches which were productive in other fields

## Main aims of the study

- To advance in the understanding of the social structure of libre software projects
- To characterize projects according to this structure
- To relate the evolution of a project to the evolution of its social structure
- To explore self-organization in the social structure of libre software projects

## Methodology

- Download CVS history information from the repository for a libre software project
- Extract the information related to who committed what
- Build with it the commiter and module networks
- Analyze the resulting networks using social network analysis
- Extract some conclusions

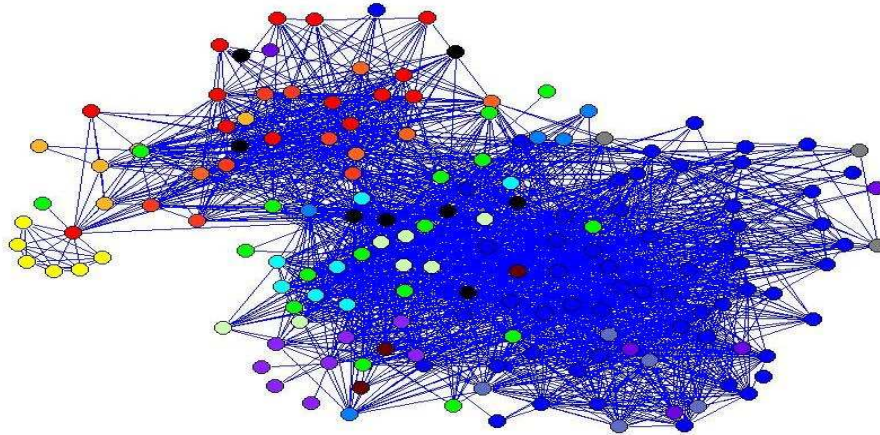
## The commiter network

- One side of affiliation network
- Each vertex, a commiter (usually a developer)
- Edge: when there is contribution to at least one common module
- Weight of edges: commits by both committers to all common modules

## The module network

- Other side of the same affiliation network
- Each vertex, a module (usually a top-level directory)
- Edge: when there is at least one common commiter
- Weight of edges: commits by common committers to both modules

## Both are a complex mesh

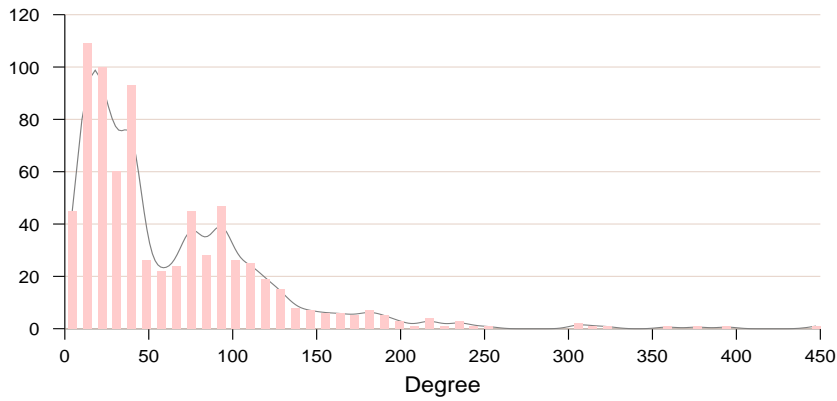


Module network for the Apache project, ca. February 2004

## But they can be characterized

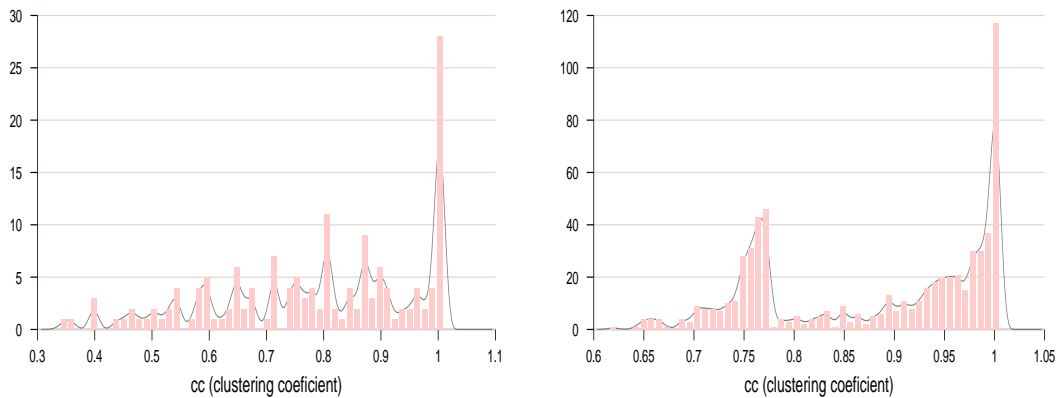
- Degree (number of connections per vertex)
- Weighted degree (in our case, by commits)
- Distance centrality (proximity to the rest of the network)
- Betweenness centrality (shortest paths traversing a vertex)
- Clustering coefficient (connectivity to the neighborhood)
- Weighted clustering coefficient (in our case, by commits)
- Community analysis (Girvan-Newman algorithm)

## Apache: connection degree (committers network)



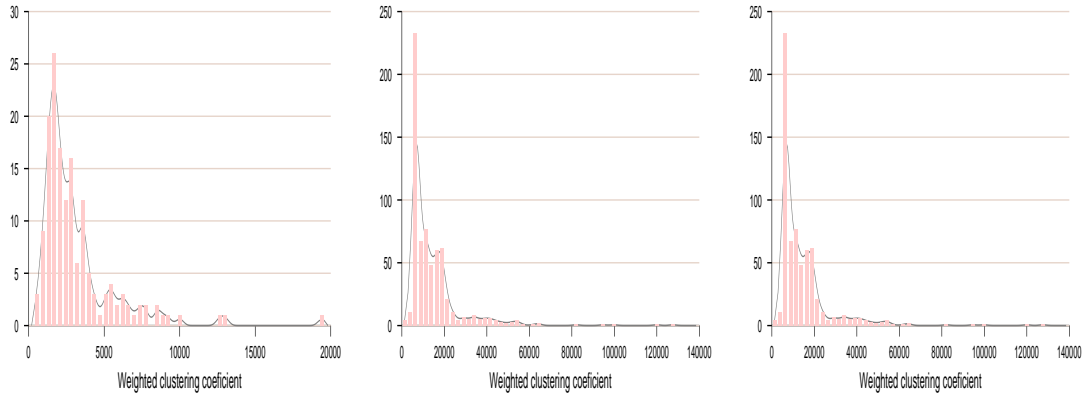
Apache, circa February 2004

## Apache and GNOME clustering coefficient (modules network)



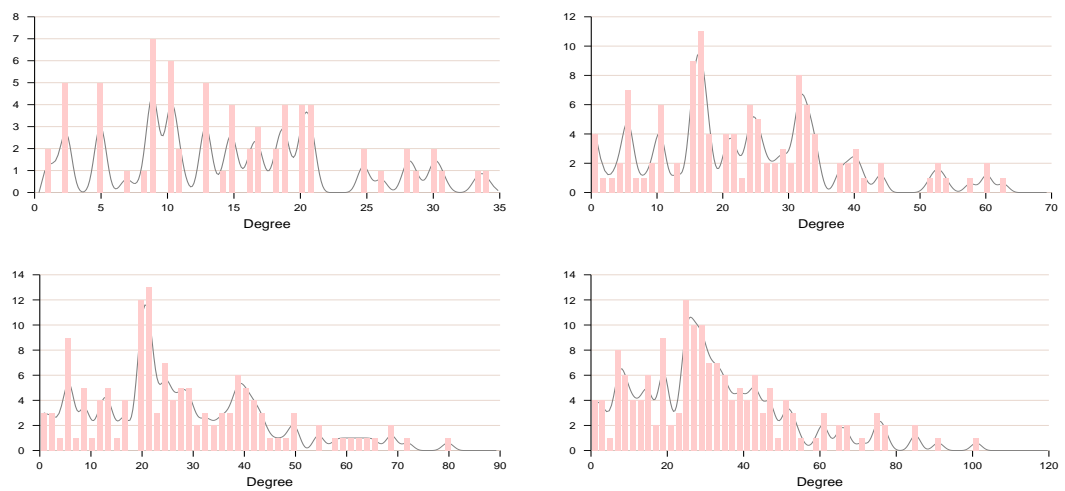
Apache (left), GNOME (right) circa February 2004

## Apache, GNOME, KDE weighted clustering coefficient (modules network)



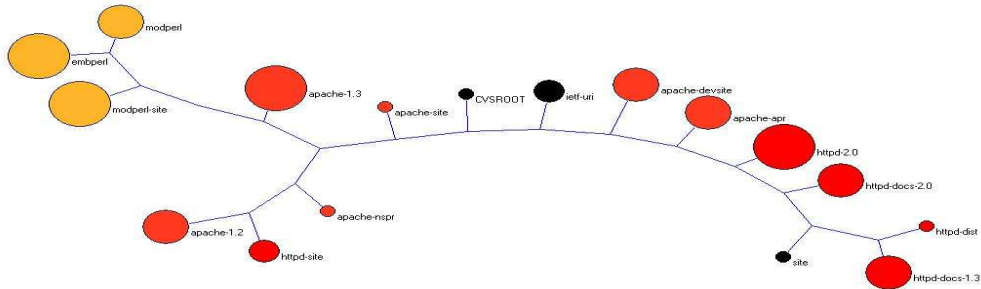
Apache (left), GNOME (center), KDE (right) circa February 2004

## Apache connection degree (modules network)

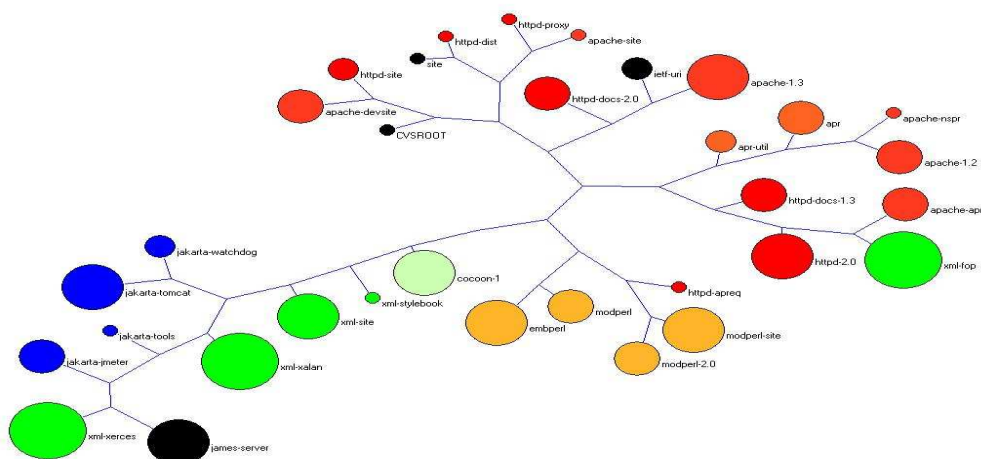


2001 (top left) to 2004 (bottom right)

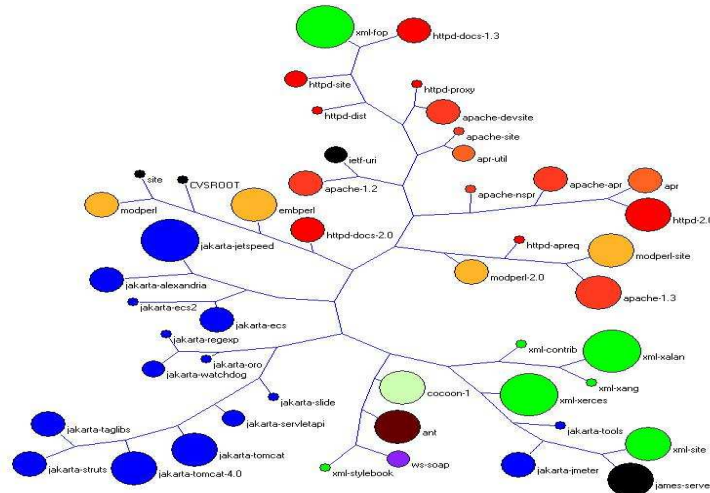
## Apache modules community analysis (1999.01)



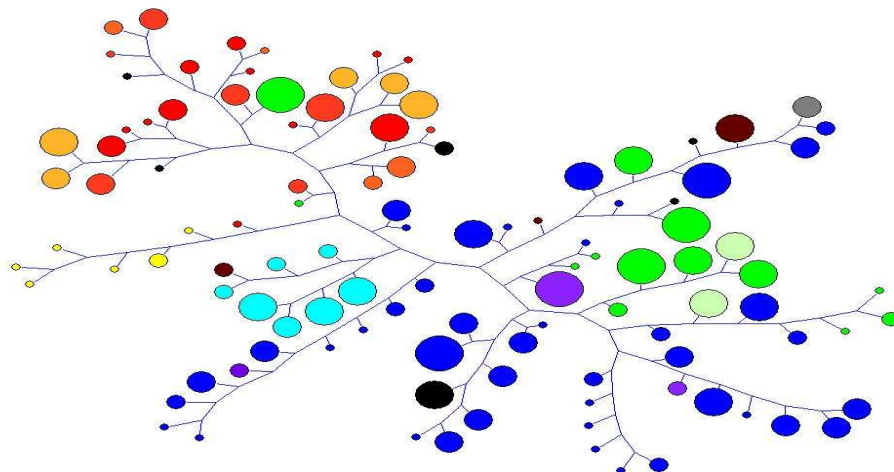
## Apache modules community analysis (2000.01)



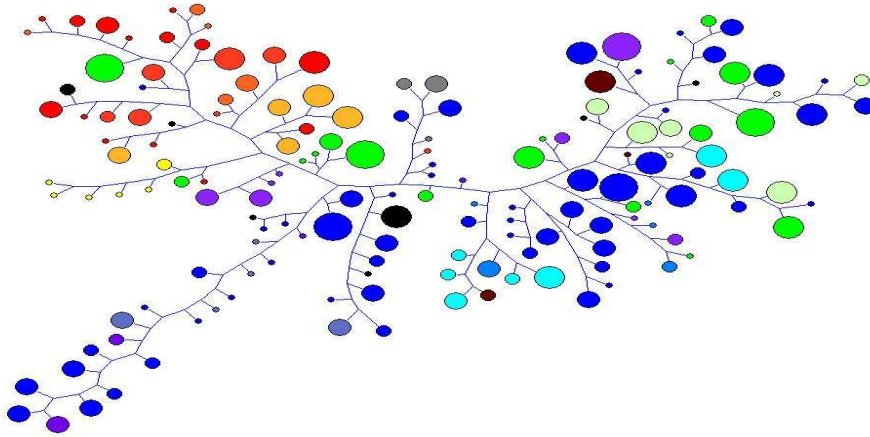
## Apache modules community analysis (2000.09)



## Apache modules community analysis (2002.01)



## Apache modules community analysis (2004.02)



## Conclusions

- Methodology for studying the structure of libre software projects
- Captures both relationships between modules and committers
- First step to community analysis
- Access to traditional social network analysis tools
- Further work: characterization of projects