

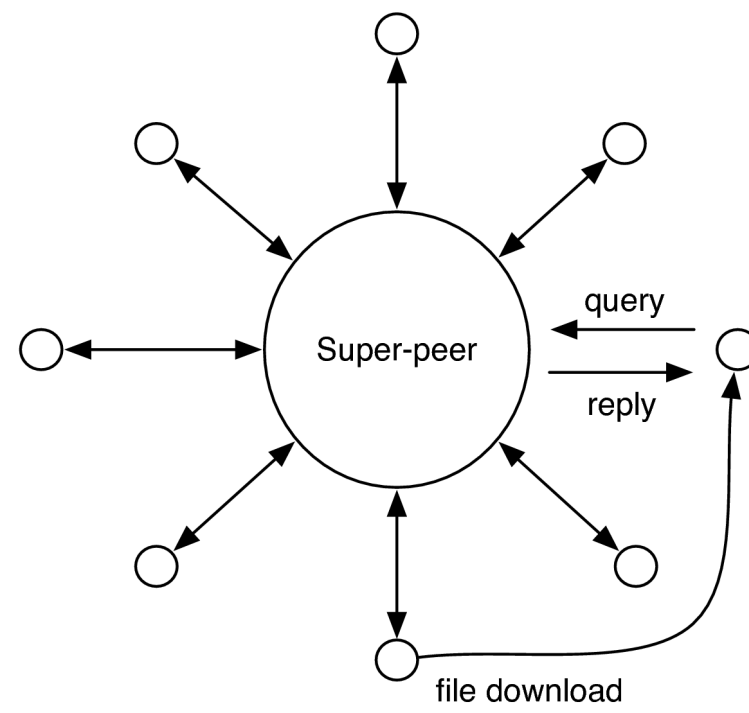
Testing Peers' Volatility

Eduardo Cunha de Almeida^a, Gerson Sunyé^a, Yves Le Traon^b and Patrick Valduriez^c
<http://peerunit.gforge.inria.fr/>

P2P systems

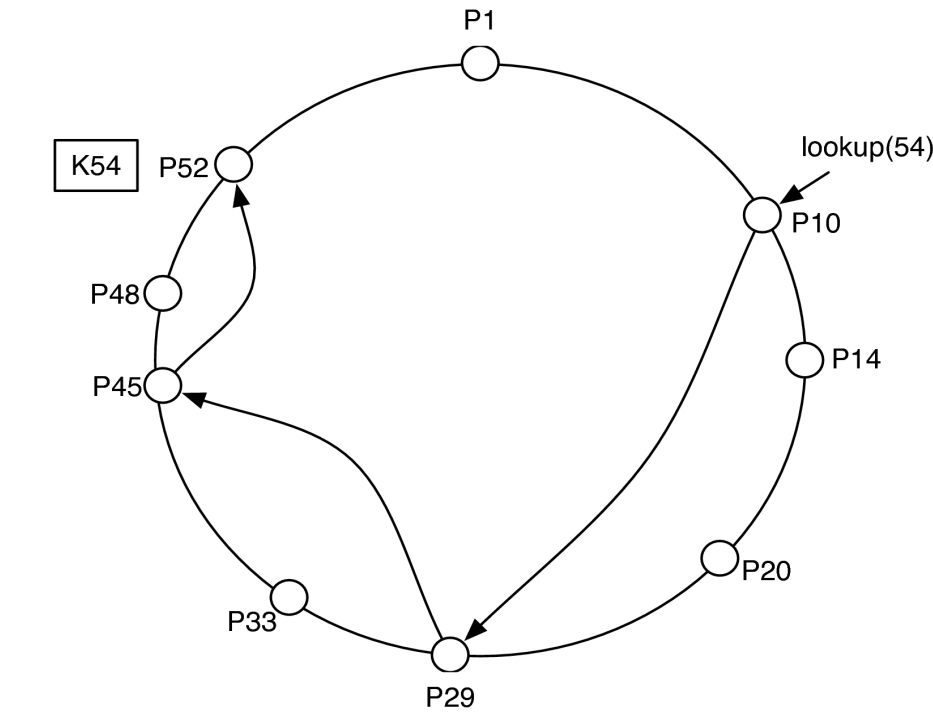
- Resource sharing
- Volatility of peers
- No single point of failure

Super-peer¹



¹JXTA

Unstructured², Structured³



²Gnutella, ³Chord, Pastry

Problem

How to validate correctness in P2P systems?

- Volatile
- Spread across the net

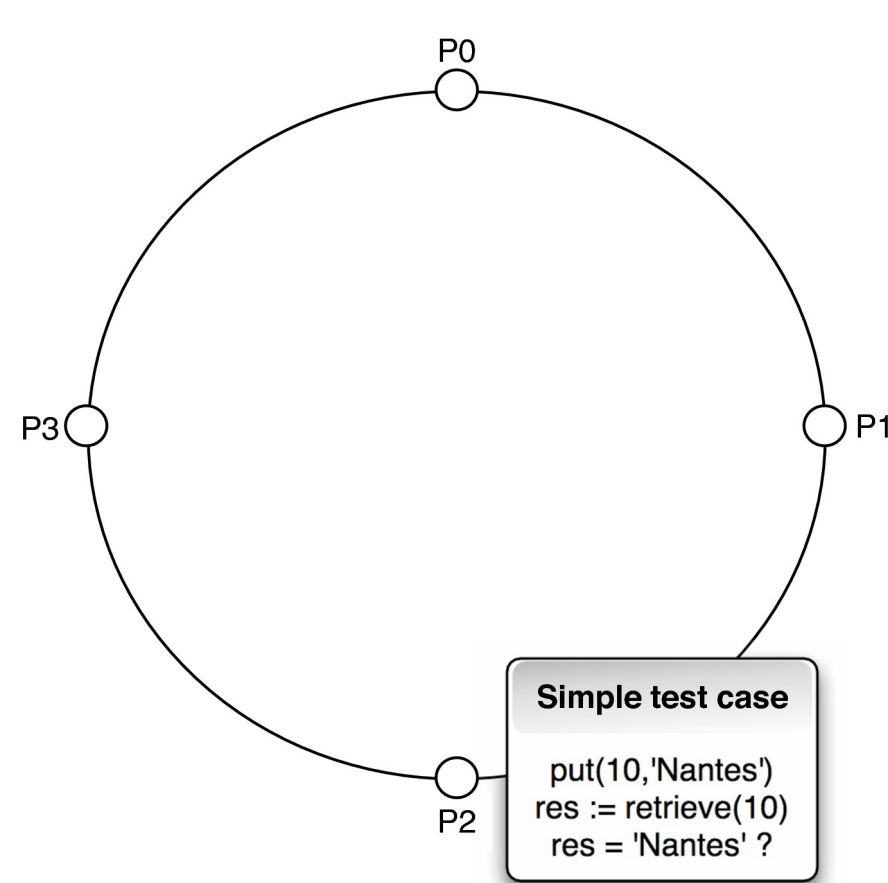
Motivation

- Increasing number of P2P systems vs.
- Very few and inadequate approaches to test them⁴.

⁴Theotokis et al. (2004)

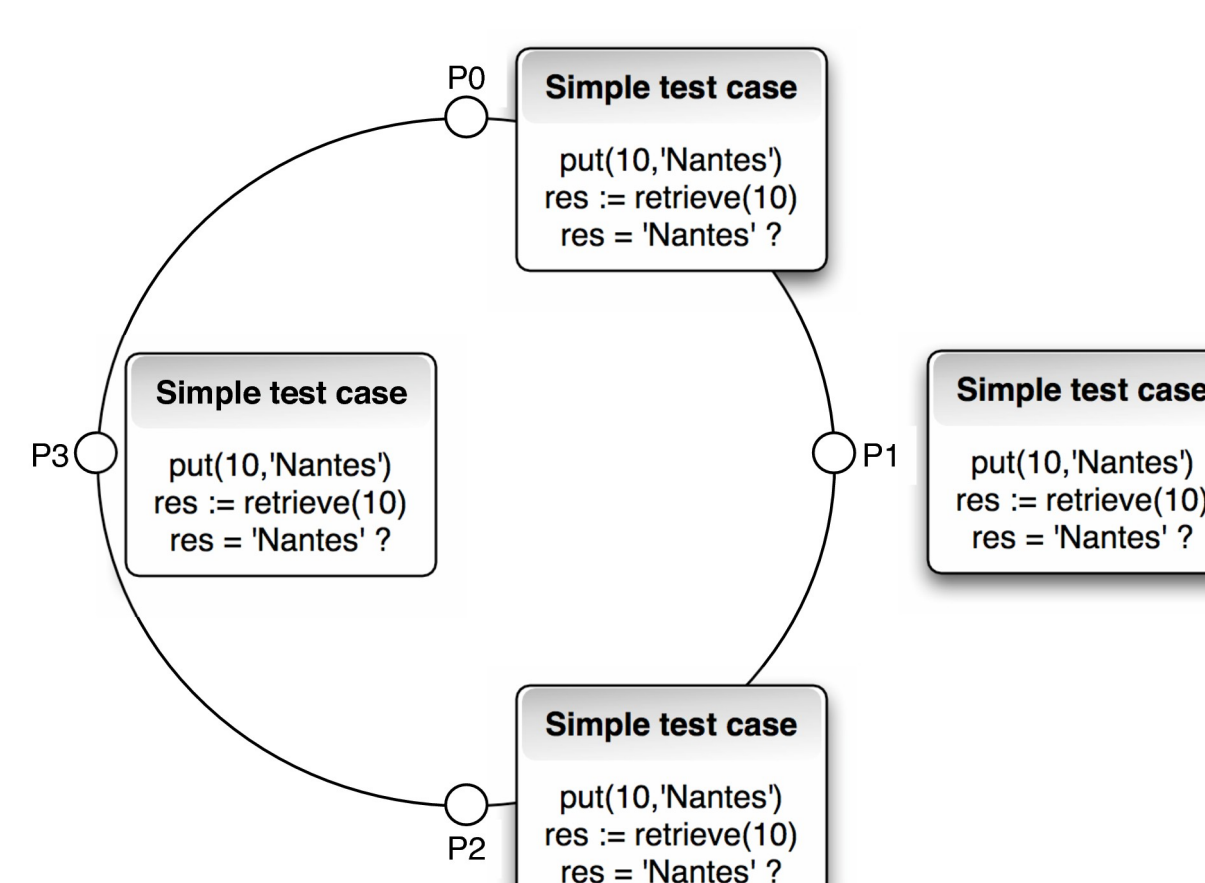
Approaches

Assume a single peer represents the entire P2P system.



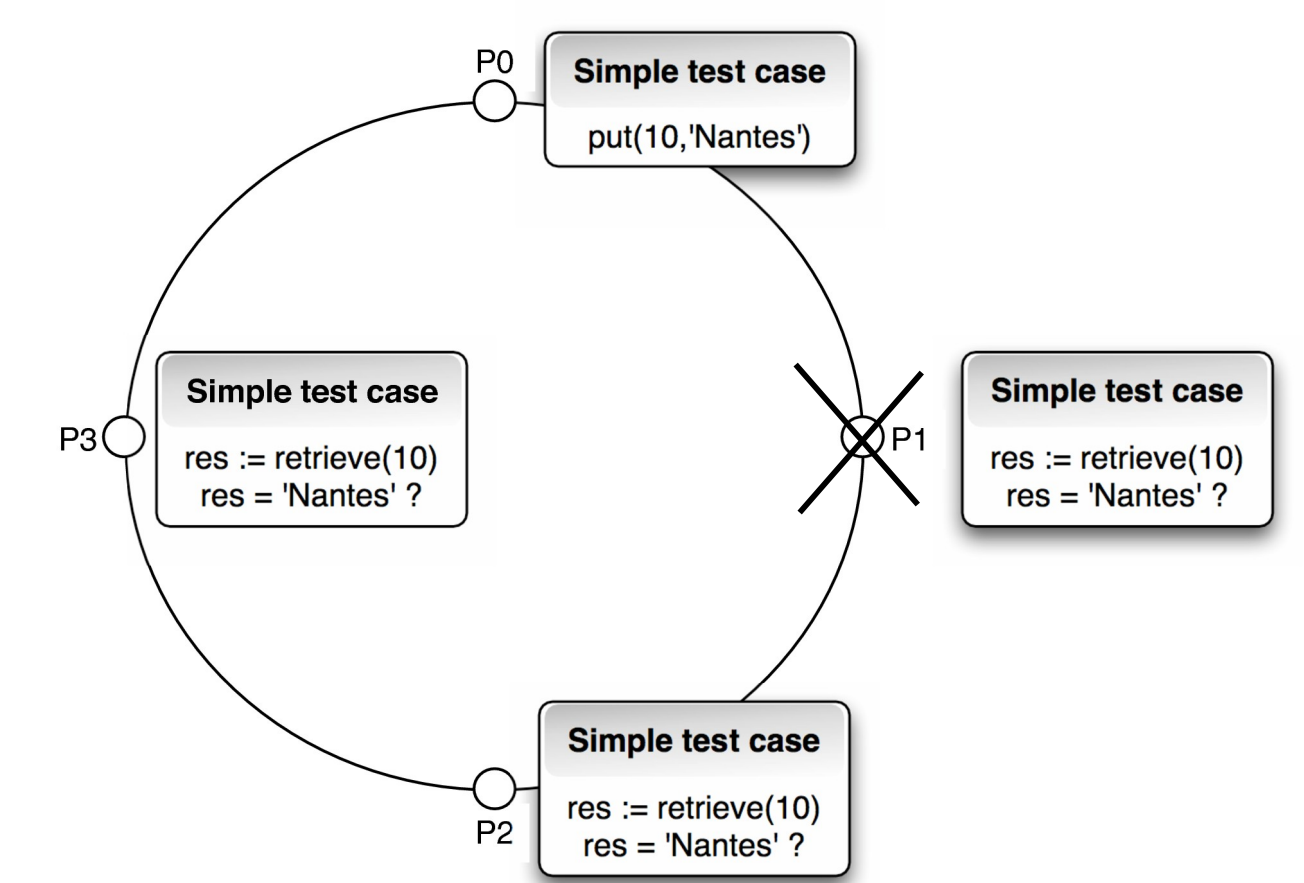
JUnit

Deploy test cases to all the peers in the system.



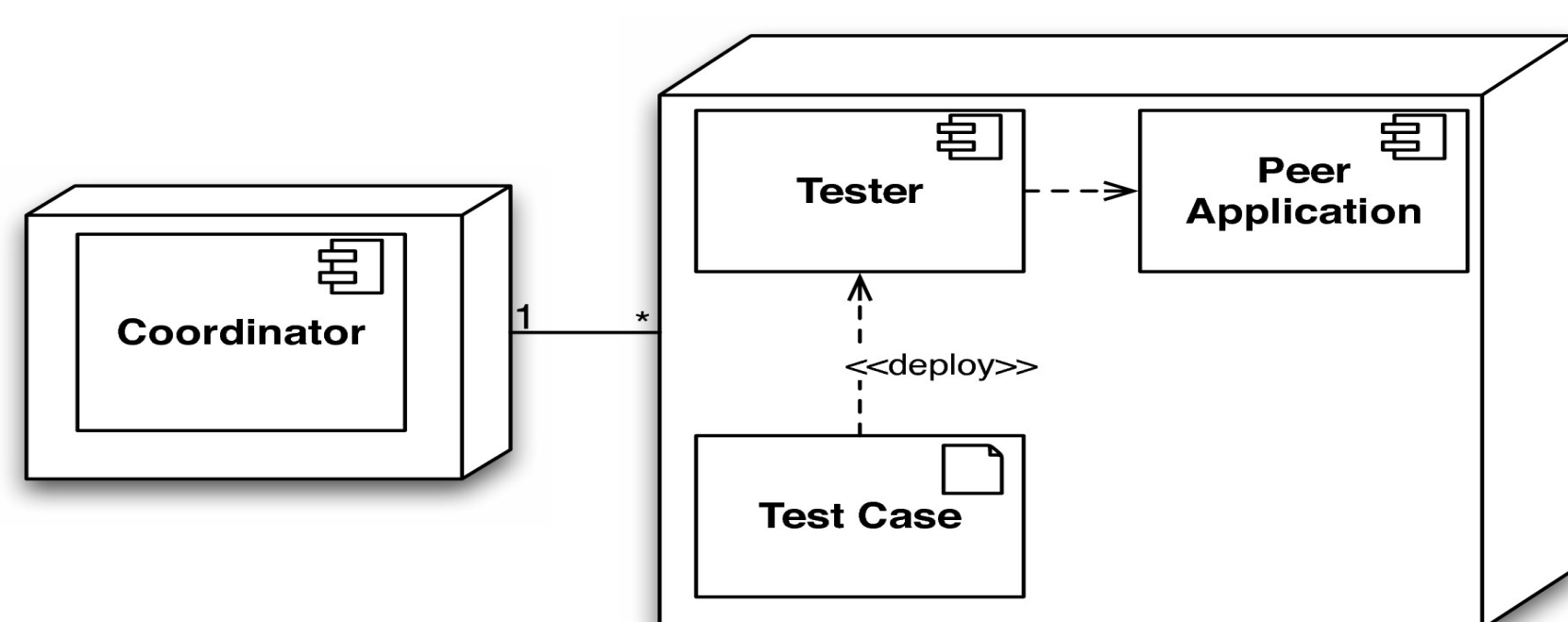
GridUnit

Decompose/deploy actions and control volatility.



PeerUnit

Architecture



- Control peer's volatility
- Decompose test cases into actions
- Deploy and manage these actions across the P2P SUT.
- Define and use variables on-the-fly

Components

Tester

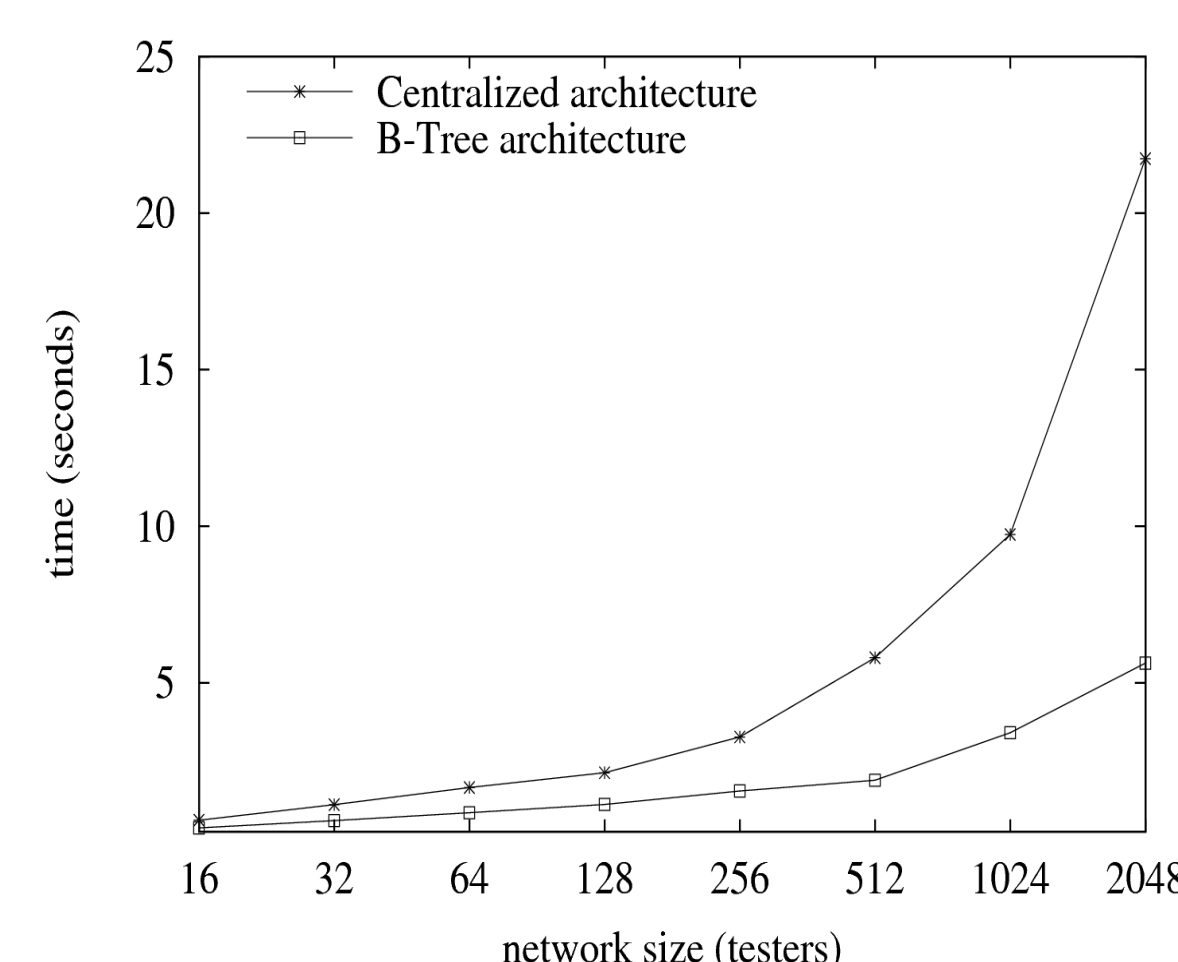
- execute actions
- control volatility

Coordinator

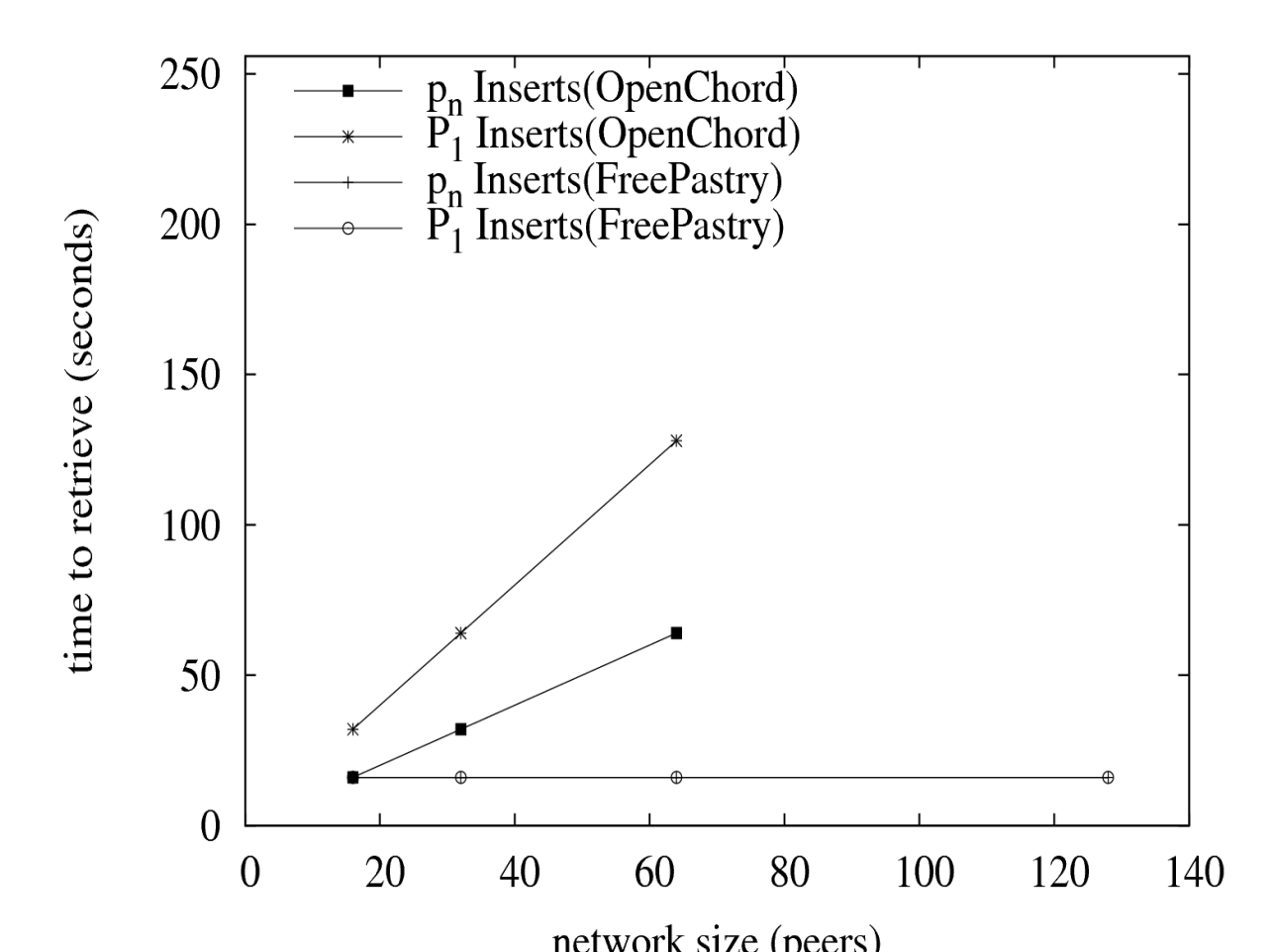
- coordinate the test sequence
- store variables
- control volatility

Results

- Evaluation of the overhead of our testing coordination algorithm
- Test of two open source P2P systems: FreePastry and OpenChord



Overhead evaluation



Insert/retrieve testing