Experience with a physics-style approach for the study of self properties in structured overlay networks

Sameh El-Ansary\textsuperscript{1}, Erik Aurell\textsuperscript{1,2}, Per Brand\textsuperscript{1} and Seif Haridi\textsuperscript{1,3,∗}

\textsuperscript{1} Distributed Systems Lab, SICS-Swedish Institute of Computer Science, Sweden
\textsuperscript{2} Department of Physics, KTH-Royal Institute of Technology, Sweden
\textsuperscript{3} IMIT, KTH-Royal Institute of Technology, Sweden
\{sameh,eaurell,perbrand,seif\}@sics.se

Abstract

This paper gives a brief summary of our experience in applying a physics-style approach for analyzing the behavior of structured overlay networks that deploy self-organization and self-repair policies. Such systems are not always simple to model analytically and simulation of scales of interest can be prohibitive. Physicists deal with scale by characterizing a system using intensive variables, i.e. variables that are size independent. The approach proved its substantial usefulness when applied to satisfiability theory and it is the hope that it can be as useful in the field of large-scale distributed systems. We report here our finding of one simple self-organization-related intensive variable, and a more complex self-repair-related intensive variable.

1 Introduction

A number of structured P2P overlays [6, 5, 7, 1], aka Distributed Hash tables (DHTs)\textsuperscript{1}, were recently designed relying on self-organization and self-repair strategies.

**Self-organization:** In a structured overlay network, nodes are supposed to self organize in a graph with a diameter and outgoing arity of nodes that are both of a logarithmic order of the number of nodes\textsuperscript{2}. Moreover, every node in the graph is responsible for the storage of data items and storage is uniformly distributed among nodes in a self-organizing fashion as well. Nodes balance storage load between themselves without any central coordination.

**Self-repair:** i) Repair of routing tables. To maintain the graph in an optimal state despite change of membership (joins/failures), each node needs to follow some maintenance policy for keeping its routing table (the outgoing edges) up-to-date. ii) Repair of storage. Upon a join, a node might need to transfer some data items to the new node. Upon a graceful departure, a node needs to hand-over its stored items to another node. Upon ungraceful failures, data items are lost and thus a replication policy is applied.

**Dominance of self-organization in DHT literature.** Our general observation on the literature of structured overlay networks is that the self-organization aspect is dominant compared to self-repair. The typical case for a paper introducing a DHT system would be to show the structure of the routing table, how the overlay graph will be constructed, protocols for joins and leaves, when it comes to self-repair, the discussion gets relatively superficial. We find arguments on the level of: “Periodic maintenance of routing tables will ensure its correctness”, “Data items have to be republished periodically by the upper layer”, etc. In the best case, a simulation is given showing that under particular stabilization rates, the network can operate.

We attribute this phenomena to two factors: i) The novelty of such systems and the requirement to establish the new concepts first before deep analysis is performed. ii) Once we have a system exhibiting a self-repair property, an analytical model that describes the behavior of the system can range in degree of difficulty from not-so-clear to very-complex-to-analyze. Therefore, we see the compelling need for more studies of analytical nature that can tell us whether those novel approaches are really useful or over-hyped.

2 The physics-style approach

Motivation. Having observed that analytical models are not always trivial to formulate given a
system applying a given self-repair policy, therefore, using simulation seemed to be the practical tool for analyzing such systems. However, scales of interest could be prohibitive for simulation purposes. At this point, a physics-style approach started to be of interest since physicists are accustomed to reasoning about large natural systems.

**How do physicists deal with scale?** The first level of analysis in a physical system of many components, is to try to separate intensive and extensive variables. Extensive variables are those that eventually become proportional to the size of the system, such as total energy. Intensive variables, such as density, temperature and pressure, on the other hand, become independent of system size. A description in terms of intensive variables only is a great step forward, as it holds regardless of the size of the system, if sufficiently large. Further steps in a physics-style analysis may include identifying phases, in each of which all intensive variables vary smoothly, and where the characteristics of the system remain the same. **Was the approach useful in the computer science arena?** A physics-style approach was carried over to satisfiability theory more than ten years ago. KSAT is the problem to determine if a conjunction of \( M \) clauses, each one a disjunction of \( K \) literals out of \( N \) variables can be satisfied. Both \( M \) and \( N \) are extensive variables, while \( \alpha = M/N \), the average number of clauses per variable, is an intensive variable. For large \( N \), instances of KSAT fall into either the SAT or the UNSAT phase depending on whether \( \alpha \) is larger or smaller than a threshold \( \alpha_c(K) \) [2]. Without question, statistical mechanics have been proved to be very useful on very challenging problems in theoretical computer science, and it can be hoped that this will also be the case in the analysis and design of distributed systems.

![Figure 1: The average lookup length as a function of \( \rho \) and \( N \).](image1)

![Figure 2: Data collapse of the average lookup length as a function of \( \rho \) and \( N \) compared to \( 0.5 \log_2 \rho \).](image2)

## 3 Intensive variables of structured overlays networks

Having explained the importance of identifying intensive variables in describing characteristics that hold irrespective of size. The question that we are trying to investigate is: “Is it possible to find intensive variables to describe the characteristics of structured overlay networks that deploy self-organization and self-repair policies?”

### 3.1 A self-repair-related intensive variable

To establish our methodology we started first with analyzing a self-organization aspect of a DHT, namely, the effect of the density of nodes in the identifier space. We used Chord as an example system on which we conduct our experiments.

The analysis followed the three following methodological steps:

**Step 1: Nomination of intensive variables.** Let \( N \) be the size of the identifier space and \( P \) be the population, i.e. the number of nodes that are uniformly distributed in the identifier space. We define the density \( (\rho) \) to be the ratio \( P/N \) with a maximum value of 1 for a fully populated system. Our question is: “Is \( \rho \) an intensive variable?”

**Step 2: Looking for characteristic behavior.** A key quantity of interest in a DHT system is the average lookup path length. Therefore, studying the effect of the density on the average lookup path length should represent a characteristic behavior.

**Step 3: Simulation.**

**Experiments set-up** Let CHORD\((P,N)\) be an optimal Chord graph, where all the fingers of all nodes are correctly assigned. For all \( N \in \{2^7, 2^8, .., 2^{14}\} \).
for all \( P \in \{0.1 \times N, 0.2 \times N, \ldots, 1.0 \times N\} \), we generate \( \text{CHORD}(P, N) \), inject uniformly distributed \( P^2 \) lookups, and record the average lookup length over the \( P^2 \) lookups denoted \( \langle L(P, N) \rangle \) or equivalently \( \langle L(\rho, N) \rangle \). This procedure is repeated 10 times, with different random seeds, and the results are averaged.

**Results.** Figure 1 shows the behavior of the path length as a function of the density and the size of the identifier space. The curves are, to first approximation, vertically shifted by the same distance, while the values of \( N \) used are exponentially spaced. This means that the dependence on \( N \) alone (constant \( P \)) is logarithmic. Indeed, it was noted in the Chord papers that the average path length is \( 0.5 \times \log P \). However, we can see an additional observation by looking at the data collapse obtained in figure 2 by subtracting \( \langle L(1, N) \rangle \) from every respective curve \( \langle L(\rho, N) \rangle \) compared to \( 0.5 \log_2 \rho \). From the data collapse, we can clearly see that \( \langle L(\rho, N) \rangle = 0.5 \log_2 \rho + f(\rho) \) where the function \( f \) is a decreasing function. That is for any given number of nodes, the average lookup length increases when they are placed in a smaller identifier space and the relative effect is the same irrespective of the system size, therefore \( \rho \) is an intensive variable.

It is a curious fact that the function \( f(\rho) \) alluded to above is decreasing because if the \( P \) populated nodes are regularly spaced in the circular address space, the average path length is exactly \( 0.5 \log_2 P \), in other words larger. Hence, we have as a result that randomization improves the performance of P2P system built on DHT, even in a static situation, with no peers leaving or joining the system.

**3.2 A self-repair-related intensive variable**

**Step 1: Nomination of intensive variables.** Let \( \tau \) be the time between two stabilization actions of a certain node. Let \( \mu \) be the average time between two perturbation events (joins or failures) while the network is in a stable state. That is, the number of nodes is varying around a certain population \( P_0 \). We are interested in understanding the interaction between perturbation and stabilization as two opposite forces, the former pulling the network towards suboptimal performance and the later bringing it back to optimal state. However, the behavior of that interaction is not known. Taking \( \mu \) as the magnitude of perturbation and \( \tau \) as the magnitude of stabilization, we need to answer the following question: “Is \( \beta = \frac{\mu}{\tau} \) an intensive variable?”

**Step 2: Looking for characteristic behavior.** In this investigation, we did not start with the average lookup length as the indicator for a characteristic behavior, we needed a more descriptive metric and thus we used what we call the “distance from optimal network \( \delta \)” which is computed as follows:

\[
\delta = \frac{\sum_{i \in P} \sum_{j = 1}^{\log N} \text{Edge}_i^j \neq \text{OptimalEdge}_i^j}{P \log N}
\]

(1)

Where \( \text{Edge}_i^j \) is the \( j^{th} (1 \leq j \leq \log N) \) outgoing edge of a node \( i \in P \) and \( \text{OptimalEdge}_i^j \) is the optimal value for that edge. \( P \log N \) is the total number of edges \( (P \text{ nodes, } \log(N) \text{ edges per node, where } N \text{ is the size of the identifier space}) \). Informally, \( \delta \) is the number of “wrong/outdated” edges over the total number of edges.

**Step 3: Simulation.**
Experiments set-up. We let $P_0$ nodes form a network and we wait until $\delta$ is equal to 0.0 i.e. the network graph is optimal. We then let the network operate under specified values of $\mu$ and $\tau$ for 50 turnovers (A turnover is the replacement of $P$ nodes with another $P$). During this experiment we record $\delta$ frequently and average it over the whole experiment, denoted $\langle \delta \rangle$.

Effect of $\beta$ on a fixed $P_0$. For one value of $P_0$, we examine various values of $\beta$ by fixing $\tau$ and varying $\mu$. We then, repeat the whole experiment with a different $\tau$ and vary $\mu$ such that the same values of $\beta$ are conserved. Effect of beta under different values of $P_0$. We repeat the above procedure under different values of $P_0$.

Results. As shown in figure 3, all the curves of a given $P_0$ are superimposed. This means that for a network undergoing changes around an average size $P_0$, irrespective of how fast is the stabilization ($\tau$) or how fast is the perturbation ($\mu$), as long as their ratio ($\beta$) is the same, the average distance from optimal network ($\langle \delta \rangle$) is the same. To see the behavior irrespective of the size, we need to perceive the obtained results differently. The stabilization as defined above is a “node-level” event while the perturbation is a “whole-graph-level” event. That is, $\beta$ is defined as the Perturbation of the system ($\mu$) / Stabilization of each node ($\tau$). Therefore, if we were to compare the behavior of two network sizes under the same values of $\tau$ and $\mu$, the network with larger size will have the same perturbation but higher stabilization since the number of nodes is larger. Therefore, we define $\beta'$ to be the Perturbation of the system ($\mu$) / Stabilization of the system ($\tau$), to have a more fair comparison. The $\beta'$ re-plot of figure 3 is shown in figure 4 where the data collapse shows that all the system sizes behave the same and therefore $\beta'$ is an intensive variable.

4 Conclusion and future work

We have reported in this paper our progress in investigating whether a physics-style analytical approach can give more understanding to the performance of structured overlay networks. The approach mainly necessitates the description of the characteristics of the system using variables that do not depend on the size, known as intensive variables.

Using this approach, we have shown that: i) The density of nodes in an identifier space, a self-organization-related variable is an intensive variable that describes a characteristic behavior of a network irrespective of its size. ii) The ratio of perturbation to stabilization $\beta$, a self-repair-related variable governs the absolute number of wrong pointers in an overlay graph, irrespective of its size.

In the continuation of this work, we intend to do the following: Perform the same experiment with a wider spectrum of numbers to have more statistically-accurate results. Use the characteristic behaviors in providing a more adaptive nature to the current DHT algorithms. Search for more intensive variables and possible phase transitions.

References


