

SOS: SELF-ORGANIZING SUBSTRATES

THÈSE N° 3615 (2006)

PRÉSENTÉE LE 21 AOÛT 2006

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Laboratoire de système d'information répartis

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Anwitaman DATTA

Bachelor of Technology in Electrical Engineering, IIT Kanpur, Inde
de nationalité indienne

acceptée sur proposition du jury:

Prof. M. Hasler, président du jury
Prof. K. Aberer, directeur de thèse
Prof. M. A. Shokrollahi, rapporteur
Prof. P. Felber, rapporteur
Prof. E. Aurell, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL

2006

Abstract

Large-scale networked systems often, both by design or chance exhibit self-organizing properties. Understanding self-organization using tools from cybernetics, particularly modeling them as Markov processes is a first step towards a formal framework which can be used in (decentralized) systems research and design. Interesting aspects to look for include the time evolution of a system and to investigate if and when a system converges to some absorbing states or stabilizes into a dynamic (and stable) equilibrium and how it performs under such an equilibrium state. Such a formal framework brings in objectivity in systems research, helping discern facts from artefacts as well as providing tools for quantitative evaluation of such systems.

This thesis introduces such formalism in analyzing and evaluating peer-to-peer (P2P) systems in order to better understand the dynamics of such systems which in turn helps in better designs.

In particular this thesis develops and studies the fundamental building blocks for a P2P storage system. In the process the design and evaluation methodology we pursue illustrate the typical methodological approaches in studying and designing self-organizing systems, and how the analysis methodology influences the design of the algorithms themselves to meet system design goals (preferably with quantifiable guarantees). These goals include efficiency, availability and durability, load-balance, high fault-tolerance and self-maintenance even in adversarial conditions like arbitrarily skewed and dynamic load and high membership dynamics (churn), apart of-course the specific functionalities that the system is supposed to provide.

The functionalities we study here are some of the fundamental building blocks for various P2P applications and systems including P2P storage systems, and hence we call them substrates or base infrastructure. These elemental functionalities include: (i) Reliable and efficient discovery of resources distributed over the network in a decentralized manner; (ii) Communication among participants in an address independent manner, i.e., even when peers change their physical addresses; (iii) Availability and persistence of stored objects in the network, irrespective of availability or departure of individual participants from the system at any time; and (iv) Freshness of the objects/resources' (up-to-date replicas).

Internet-scale distributed index structures (often termed as structured overlays) are used for discovery and access of resources in a decentralized setting. We propose a rapid construction from scratch and maintenance of the P-Grid overlay network in a self-organized manner so as to provide efficient search of both individual keys as well as a whole range of keys, doing so providing good load-balancing characteristics for diverse kind of arbitrarily skewed loads - storage and replication, query forwarding and query answering loads. For fast overlay construction we employ recursive partitioning of the key-space so that the resulting partitions are balanced with respect to storage load and replication. The proper algorithmic parameters for such partitioning is derived from a transient analysis of the partitioning process which has Markov property. Preservation of ordering information in P-Grid such that queries other than exact queries, like range queries can be efficiently and rather trivially handled makes P-Grid suitable for data-oriented applications. Fast overlay construction is analogous to building an index on a new set of keys making P-Grid suitable as the underlying indexing mechanism for peer-to-peer information retrieval applications among other potential applications which may require frequent indexing of new attributes apart regular updates to an existing index.

In order to deal with membership dynamics, in particular changing physical address of peers across sessions, the overlay itself is used as a (self-referential) directory service for maintaining the participating peers' physical addresses across sessions. Exploiting this self-referential directory, a family of overlay maintenance scheme has been designed with lower communication overhead than other overlay maintenance strategies. The notion of dynamic equilibrium study for overlays under continuous churn and repairs, modeled as a Markov process, was introduced in order to evaluate and compare the overlay maintenance schemes.

While the self-referential directory was originally invented to realize overlay maintenance schemes with lower overheads than existing overlay maintenance schemes, the self-referential directory is generic in nature and can be used for various other purposes, e.g., as a decentralized public key infrastructure. Persistence of peer identity across sessions, in spite of changes in physical address, provides a logical independence of the overlay network from the underlying physical network. This has many other potential usages, for example, efficient maintenance mechanisms for P2P storage systems and P2P trust and reputation management. We specifically look into the dynamics of maintaining redundancy for storage systems and design a novel lazy maintenance strategy. This strategy is algorithmically a simple variant of existing maintenance strategies which adapts to the system dynamics. This randomized lazy maintenance strategy thus explores the cost-performance trade-offs of the storage maintenance operations in a self-organizing manner. We model the storage system (redundancy), under churn and maintenance, as a Markov process. We perform an equilibrium study to show that the system operates in a more stable dynamic equilibrium with our strategy than for the existing maintenance scheme for comparable overheads. Particularly, we show that our maintenance scheme provides substantial performance gains in terms of maintenance overhead and system's resilience in presence of churn and correlated failures.

Finally, we propose a gossip mechanism which works with lower communication overhead than existing approaches for communication among a relatively large set of unreliable peers without assuming any specific structure for their mutual connectivity. We use such a communication primitive for propagating replica updates in P2P systems, facilitating management of mutable content in P2P systems. The peer population affected by a gossip can be modeled as a Markov process. Studying the transient spread of gossips help in choosing proper algorithm parameters to reduce communication overhead while guaranteeing coverage of online peers.

Each of these substrates in themselves were developed to find practical solutions for real problems. Put together, these can be used in other applications, including a P2P storage system with support for efficient lookup and inserts, membership dynamics, content mutation and updates, persistence and availability. Many of the ideas have already been implemented in real systems and several others are in the way to be integrated into the implementations.

There are two principal contributions of this dissertation. It provides design of the P2P systems which are useful for end-users as well as other application developers who can build upon these existing systems. Secondly, it adapts and introduces the methodology of analysis of a system's time-evolution (tools typically used in diverse domains including physics and cybernetics) to study the long run behavior of P2P systems, and uses this methodology to (re-)design appropriate algorithms and evaluate them.

We observed that studying P2P systems from the perspective of complex systems reveals their inner dynamics and hence ways to exploit such dynamics for suitable or better algorithms. In other words, the analysis methodology in itself strongly influences and inspires the way we design such systems. We believe that such an approach of orchestrating self-organization in internet-scale systems, where the algorithms and the analysis methodology have strong mutual influence will significantly change the way future such systems are developed and evaluated. We envision that such an approach will particularly serve as an important tool for the nascent but fast moving P2P systems research and development community.

Keywords: *Peer-to-peer (P2P), Randomized algorithms, Self-organization, Markov model.*

Version Abrégée

Les systèmes de réseaux à large échelle font souvent montre, par construction ou par hasard, de propriétés d'auto-organisation. Comprendre cette auto-organisation fait appel à des techniques provenant de la cybernétique, comme leur modélisation en tant que processus markoviens, qui est le premier pas vers un contexte formel, utilisable dans la conception et la recherche de systèmes (décentralisés). Parmi les aspects intéressants souhaitables sont compris l'évolution du système en fonction du temps, ainsi qu'étudier si et quand un système converge vers des états absorbants ou se stabilise en un équilibre dynamique (et stable), et comment il se comporte dans un tel état d'équilibre. Un tel contexte formel apporte de l'objectivité dans la recherche en systèmes, aidant ainsi à discerner les faits des artefacts, ainsi qu'à fournir des outils en vue d'une évaluation quantitative de tels systèmes.

Cette thèse introduit un tel formalisme en analysant des systèmes pair-à-pair (P2P, "Peer-to-Peer") afin de mieux comprendre leur dynamisme, ce qui à son tour aide à obtenir une meilleure conception.

En particulier, cette thèse développe et étudie les blocs fondamentaux à la construction de systèmes P2P. Ce faisant, notre méthodologie d'évaluation et de conception illustre les approches méthodologiques typiques de l'étude et de la conception de systèmes s'auto-organisant, ainsi que comment la méthodologie d'analyse influence la conception d'algorithmes afin d'atteindre les objectifs de conceptions de systèmes (de préférence avec des garanties quantifiables). Ces objectifs comprennent l'efficacité, la disponibilité, et la durée, l'équilibre des charges, la tolérance aux erreurs, et l'auto-maintenance même lors de conditions adverses, comme des charges arbitrairement déséquilibrées et dynamiques, un fort dynamisme d'adhésion ("churn"), ainsi que, clairement, les fonctionnalités spéciales que le système est sensé fournir.

Les fonctionnalités que nous étudions font partie des blocs fondamentaux de nombreuses applications P2P ainsi que de systèmes incluant des systèmes de stockage P2P, nous les appelons donc substrat ou infrastructure de base. Ces fonctionnalités essentielles comprennent: (i) découverte fiable et efficace des ressources distribuées dans le réseau de façon décentralisée; (ii) Communication parmi les pairs indépendamment de l'adressage, i.e., même lors de changement d'adresse physique; (iii) Disponibilité et persistance des objets stockés dans le réseau, indépendamment de la disponibilité ou du départ de participants individuels à un moment donné; et (iv) Fraîcheur des objets/ressources (copies mises-à-jour).

Les structures d'indexage distribué à l'échelle d'internet (souvent appelées "overlays" structurés) sont utilisées pour la découverte et l'accès à des ressources décentralisées. Nous proposons une construction rapide (partant de zéro) et une maintenance auto-organisées du réseau P-Grid pour une recherche efficace des clefs individuelles et d'un ensemble de clefs, fournissant ainsi de bonnes caractéristiques d'équilibre des charges pour différentes sortes de charges arbitrairement déséquilibrées - stockage et copies, retransmission de requêtes et charges de réponse aux requêtes. Pour une construction rapide d'overlay, nous utilisons une partition récursive de l'espace des clefs, afin que les partitions résultantes soient équilibrées par rapport à la charge de stockage et de copies. Les paramètres algorithmiques propres à de telles partitions sont dérivés d'une analyse transiente du processus de partition, qui est markovien. La préservation d'un ordre de l'information dans P-Grid telle que des requêtes autres qu'exactes, comme des requêtes sur des ensembles, puissent être gérées efficacement et presque trivialement, rend P-Grid adapté aux applications orientées données ("data-oriented"). La construction rapide d'overlay est analogue à construire un index sur un nouvel ensemble de clefs, faisant de P-Grid un mécanisme potentiel d'indexage sous-jacent pour des applications P2P de récupération d'information parmi d'autres applications potentielles demandant un indexage fréquent de nouveaux attributs, et des mises-à-jour d'un index existant.

Pour gérer la dynamique d'adhésion lors des sessions, en particulier changer l'adresse physique des pairs, l'overlay lui-même est utilisé comme un service d'annuaire (auto-référentiel) pour maintenir l'adresse physique des participants. Exploitant cet annuaire auto-référentiel, une famille de schémas pour la mainte-

nance d'overlays est conçue, à coût de communication moindre que les stratégies existantes. La notion d'étude d'équilibre dynamique, sous "churn" continu et réparation, modélisé comme processus markovien, est introduite pour évaluer et comparer les schémas de maintenance d'overlays.

Alors qu'il était originalement inventé pour réaliser des schémas de maintenance d'overlays à coût moindre que les schémas existants, l'annuaire auto-référentiel est générique de nature et peut être utilisé pour différentes autres applications, e.g., comme une infrastructure à clé publique décentralisée. La persistance de l'identité des clefs lors des sessions, en dépit des changements dans l'adresse physique, fournit une indépendance logique du réseau overlay par rapport au réseau physique sous-jacent. Cela offre de nombreuses autres utilisations, comme des mécanismes efficaces de maintenance pour des systèmes de stockage, et un moyen de gérer la confiance et la réputation dans les réseaux P2P. Nous étudions en particulier la dynamique de maintenance de redondance pour les systèmes de stockage, et avons conçu une nouvelle stratégie "paresseuse" de maintenance. Celle-ci est algorithmiquement une simple variante de stratégies de maintenance existantes, qui s'adapte à la dynamique du système. Cette stratégie "paresseuse" randomisée explore donc les compromis coût-performance des opérations de maintenance nécessaires au stockage d'une manière auto-organisée. Nous modélisons le système de stockage (redondance), sous "churn" et maintenance, comme un processus markovien. Nous faisons une étude d'équilibre afin de montrer que le système opère dans un équilibre dynamique plus stable avec notre stratégie qu'avec les schémas de maintenance existants pour des coûts comparables. En particulier, nous montrons que notre schéma de maintenance fournit des gains substantiels de performance en terme de coût de maintenance et de résilience du système en présence de "churn" et de défaillances corrélées.

Finalement, nous proposons un mécanisme de bavardage (gossip), qui fonctionne avec un coût de communication inférieur à ceux existants, pour établir une communication parmi un ensemble relativement large de pairs non-fiables, sans supposer de structure spécifique sur leur connectivité mutuelle. Nous utilisons cette primitive de communication pour propager des mises-à-jour de copies dans des systèmes P2P, rendant plus facile de gérer les contenus changeants. La population de pairs affectée par le bavardage est modélisée comme un processus markovien. Etudiant la propagation transiente du bavardage aide à choisir les paramètres de l'algorithme afin de réduire le coût de communication tout en garantissant la couverture des pairs online. Chacun de ces substrats ont été développés pour trouver des solutions pratiques à des problèmes réels. Mis ensemble, ceux-ci peuvent être utilisés dans d'autres applications incluant un système de stockage P2P supportant des recherches et des insertions efficaces, de la dynamique d'adhésion, de la mutation dans les contenus, des mises-à-jour, de la persistance et de la disponibilité. Plusieurs de ces idées ont déjà été implémentées dans des systèmes réels.

Cette dissertation a deux principales contributions. Elle fournit une conception de systèmes P2P, qui sont utiles à des utilisateurs comme à d'autres développeurs, qui peuvent construire d'autres systèmes sur ceux existants. Elle adapte et introduit la méthodologie d'analyse de l'évolution du temps d'un système (outil typiquement utilisés en physique et cybernétique) afin d'étudier le comportement à long terme des systèmes P2P, et utilise cette méthodologie pour (re-)concevoir des algorithmes appropriés et les évaluer.

Nous avons observé qu'étudier les systèmes P2P du point de vue des systèmes complexes révèle leur dynamique intérieure et donc des moyens d'exploiter celle-ci pour des algorithmes adaptés ou meilleurs. Autrement dit, la méthodologie d'analyse en elle-même influence fortement et inspire la façon dont nous concevons de tels systèmes. Nous croyons qu'une telle approche dans l'orchestration de l'auto-organisation dans les systèmes à l'échelle d'internet, où les algorithmes et les méthodologies d'analyse ont une forte influence mutuelle, va changer significativement la façon dont de tels systèmes seront développés et évalués dans le futur. Nous pensons qu'une telle approche va servir d'outil important pour la communauté, naissante mais déjà grandissante, de recherche et développement dans les systèmes P2P.

Mots-clés: *Pair-à-pair (P2P), Algorithmes randomisés, Auto-organisation, Modèle markovien.*

Table of Contents

1. Preamble	1
1.1 The peer-to-peer (P2P) paradigm	1
1.2 Self-organizing Substrates	3
1.2.1 Structured overlay networks	4
1.2.2 Managing peers' identity and logical mobility	5
1.2.3 Persistent and available storage	5
1.2.4 A gossiping primitive	6
1.3 The philosophy and practice of self-organization	6
1.3.1 Probabilistic systems	7
1.3.2 Markov model for self-organization	7
1.4 Thesis organization and main contributions	8

Part I. Background

2. Peering into peer-to-peer systems	15
2.1 Back to the future	15
2.1.1 Rise of the servers	16
2.1.2 P2P: A born again networking paradigm	16
2.2 Overlay networks	18
2.2.1 Unstructured overlays	18
2.3 Structured overlays	19
2.4 A taxonomy of structured overlay topologies	21
2.4.1 Ring	21
2.4.2 Tree	24
2.4.3 Hypercube	25
2.4.4 Others	25
2.5 What is stored in structured overlays? Where is it stored?	27
2.5.1 Index vs. Storage: Separation of concerns	27
2.5.2 A taxonomy of replication in structured overlays	28
2.6 Conclusion	29

Part II. Self-organizing overlay substrate

3. The P-Grid overlay network	33
3.1 Beyond DHTs	33
3.2 The P-Grid overlay network	35
3.2.1 Average Search Cost Analysis	37
3.3 Range queries: Algorithms and complexity	39
3.3.1 Min-max traversal algorithm	39
3.3.2 Shower algorithm	42
3.4 Complementary contemporary contributions	43
3.4.1 Locality	43
3.4.2 Look-ahead routing	44
3.4.3 Abstracting k-ary trees	45
3.4.4 Iterative vs. Recursive processing of an isolated query	45
3.5 Conclusion	45
4. Multi-faceted load-balanced overlay	47
4.1 Gamuts of load-balancing in structured overlays	47
4.1.1 Sources of load-skew	47
4.1.2 Alleviating load-skew	48
4.2 Need for speed in overlay construction	50
4.3 Fast construction of load-balanced overlay	53
4.3.1 Decentralized Partitioning	55
4.3.2 Adaptive eager partitioning	56
4.4 Algorithmic issues and heuristics	59
4.4.1 Initiating the indexing process	59
4.4.2 Synchronizing and terminating the indexing process	60
4.4.3 Coalescing partitions (path retraction)	61
4.4.4 Complexity	61
4.5 Peers joining a (partially) existing P-Grid network	62
4.5.1 Local view of the global structure	63
4.5.2 A new peer joining an existing P-Grid network	64
4.6 Re-balancing structural replication	65
4.6.1 Collecting statistical information at a peer	67
4.6.2 Choosing migration path for a peer	67
4.6.3 Migrating a peer	68
4.7 P-Grid as a DHT	68
4.7.1 Balanced tree construction with controlled replication	69
4.8 Evaluation results	69
4.8.1 Parallelized load-balanced overlay construction	69
4.8.2 New peers joining an existing network	72
4.8.3 Replication load balancing	74
4.8.4 Simultaneous balancing of storage and replication load in a dynamic setting	79
4.9 Related work	80

5. A first-order balancing of query-load	85
5.1 Introduction	85
5.2 Route in-degree in randomized overlay topologies	87
5.3 Replication and search cost	88
5.4 Optimal query-adaptive replication strategy for structured overlays	89
5.5 Optimal replica placement	91
5.6 Results	92
5.6.1 Balancing in-degree in randomized routing networks	92
5.6.2 Numerical evaluation: Square-root vs. Proportional replication	93
5.6.3 Simulations: Optimal query-adaptive replication	95
5.7 Conclusion and future work	97
6. A self-referential directory	101
6.1 Introduction	101
6.1.1 A separation of concern from the underlying physical network	103
6.1.2 What are some of the other overlays doing?	104
6.1.3 Motivation for a new approach	104
6.1.4 Problem statement and overview of the approach	105
6.2 Self-referential directory service protocols	108
6.3 Processing queries using self-healing routing: An example	110
6.4 Self-healing routing algorithm for a query (search)	113
6.5 Analysis of the algorithms	114
6.5.1 Models for analyzing the overlay under churn	114
6.5.2 Analysis of an isolated search/query (static resilience of the overlay)	115
6.5.3 Recursive queries and dynamic equilibrium	116
6.6 Analytical and simulation results	118
6.7 Related Work	121
6.7.1 Identity management	121
6.7.2 Security issues	122
6.7.3 Overlay route maintenance	123
6.7.4 Analysis of overlays under churn	125
6.8 Conclusions	127
7. Experimental evaluation on PlanetLab	129
7.1 PlanetLab as an experiment testbed	129
7.2 Objectives and scope of the experiments	130
7.3 Experimental setup for overlay construction by recursive re-partitioning	130
7.3.1 Experimental evaluation	131
7.4 Experimental setup for evaluation of the range query algorithms	132
7.4.1 Experimental evaluation	135
7.5 Conclusion	140

8. Efficient redundancy maintenance in storage systems	145
8.1 Introduction	145
8.2 Redundancy mechanisms: Replication, Erasures and Digital Fountains	147
8.3 Maintenance strategies	149
8.4 Markovian time-evolution analysis	151
8.5 Churn model	152
8.6 Analysis: Erasure code based redundancy, lazy maintenance	154
8.6.1 Effect of churn	155
8.6.2 Lazy Maintenance Strategy-A: Deterministic Procrastination	155
8.6.3 Lazy Maintenance Strategy-B: Sampling Random Subsets	156
8.6.4 Correlated failures	157
8.7 Results	158
8.7.1 Validation of the analytical model	158
8.7.2 Static resilience versus steady state analysis	160
8.7.3 Overheads of lazy maintenance mechanisms	160
8.7.4 Surviving correlated failures while using lazy repairs	161
8.7.5 Convergence, uniqueness and stability (of the system) and validity of the model	163
8.8 Ongoing and future work	165
8.9 Conclusion	166
9. A push/pull gossiping primitive for unstructured sub-networks	167
9.1 Introduction	167
9.2 Motivation and problem statement	168
9.3 System model	169
9.4 Analysis	171
9.4.1 Setup and notation for the analysis	171
9.4.2 Analysis of the push phase	173
9.4.3 Analysis of the pull phase	176
9.4.4 Query (request)	177
9.5 Analytical results	177
9.5.1 Impact of the initial online population size	178
9.5.2 Impact of varying fanout (f_r)	178
9.5.3 Impact of departing peers (σ)	178
9.5.4 Impact of probability of forwarding ($P_F(t)$)	178
9.5.5 Scalability	181
9.5.6 Comparison with simple flooding (like in Gnutella) and variants	181
9.6 Potential optimizations and self-tuning	183
9.7 Related work	184
9.7.1 Replication and updates in databases	184
9.7.2 Group communication and lazy epidemic schemes	185
9.7.3 Peer-to-peer systems	186
9.8 Future work	187
9.9 Conclusions	187

10. Conclusion	191
10.1 Interplay of peer-to-peer systems	191
10.2 We build upon our tools	193
10.3 Analyzing self-organization	194
10.3.1 Transient versus steady-state analysis	194
10.3.2 Mean value versus density (distribution) function	195
10.4 Future directions	196
10.5 A blend of systems and theory for peer-to-peer research	197
A. Decentralized partitioning (further analysis)	213
A.1 Evolution of (probability) distribution function	213
A.2 Error Analysis	215
A.2.1 Numerical Simulation of the Markov Model	216