Off-line Karma: Towards a Decentralized Currency for Peer-to-peer and Grid Applications (Brief Abstract)*

Flavio D. Garcia and Jaap-Henk Hoepman
Nijmegen Institute for Computing and Information Science,
Radboud University Nijmegen, The Netherlands.
{flaviog,jhh}@cs.ru.nl

15th September 2004

Abstract

P2P and grid systems allow their users to exchange information and share resources in a uniform and reliable manner. In an ideal world, users make roughly as much resources available as they use. In reality, this is not always the case, and some kind of currency or barter (called *karma*) is needed that can be exchanged for resources to limit abuse. P2P and grid systems are distributed systems without centralised control or hierarchical organisation. Unfortunately, all currency-like systems known require some kind of centralised control to manage security and to detect fraud (e.g. double spending).

To solve this problem, we present a completely decentralised, off-line karma implementation for P2P and grid systems, that detects double spending and other types of fraud under varying adversarial scenarios. The system is based on the tracing of the spending pattern of coins, and distributing the normally central role of a bank over a predetermined, but random, selection of nodes. The system is designed to allow nodes to join and leave the system at arbitrary times[1].

*Id: karma-smp-abstract.tex,v 1.2 2004/09/15 07:43:23 jhh Exp

[1]An extended abstract of this paper is available on www.cs.ru.nl/~jhh

1 Introduction

Peer-to-peer (*aka* P2P) networks and grid systems like BitTorrent [Coh03] and Gnutella [Kir] distributed systems without centralised control or hierarchical organisation. Given this flat structure, these systems scale very well when the number of nodes increases. Scalability is important, given the fact that the Internet is still growing exponentially and more people have permanent Internet connections. Current applications of these systems include but are not limited to: file sharing, redundant storage, distributed computations, data perpetuation, and providing anonymity.

Grid systems have been developed as a response to the fact that computer resources are usually very badly distributed in time and space, and almost all of them are wasted most of the time. CPU cycles are maybe the best example of that. Most of the computers in the world are usually idle, with occasional periods of hi-load. Then, it seems natural to make available resources when idle, and to be able to other users' resources in return when needed. In an ideal grid system, the whole Internet constitutes a huge supercomputer with practically unlimited resources, that can use as long as they contribute to it as well.

Projects like seti@home, folding@home and distributed.net have shown that a big set of common desktop computers can provide a tremendous amount of computing power. Even

though they receive no direct benefit, users participate in such projects because the associate themselves with the goals of the project, and the very odd chance of being the lucky guy that finds the solution. If such large scale computations are for an uncompelling cause, is not easy to find people willing to donate their CPU time. Also, many P2P networks suffer form the 'free-riders' problem where users only occasionally connect to the network to use the resources offered by it, but do not donate any resources themselves.

To counter both these problems, 'currencies' of some sort have been proposed to reward users contributing to the system and that can be used as payment when the resources of the network are used.

1.1 Related work

Several P2P systems like POPCORN [NLRC98] and MojoNation[2] use some kind of digital currency to enforce contribution or optimise resource distribution. All these systems use a central bank or broker to keep track of each user's balance and transactions. Micropayment schemes like PayWorld and MicroMint [RS97], Millicent [GMA+95] and Pepercoin [Riv04] appear to be especially suitable for such a task. In all of these schemes the load of the broker grows linearly with the number of transactions.

It is clear that when scalability is of primary concern, which is the case for P2P networks and grid systems, a central bank or broker constitutes a bottleneck as well as a single point of failure. The aforementioned approaches are therefore not viable to solve the problem under consideration.

There should be no bank or any kind of central server. All nodes should run the same client and there should be no difference between the relevance and responsibilities that each node possesses. At the moment, the only distributed currency we are aware of that fulfils these characteristics is KARMA [VCS03]. KARMA splits the bank functionality in different bank sets, which are sets of users of size k. Each of these bank sets is responsible for keeping the state balance of a set of users. In KARMA, every transaction between a user a and b involves communication between a and b, between a and b and their bank sets, and most importantly, it involves k to k communication between the bank set of a and the bank set of b. This incurs a big overhead, especially in case that the transaction rate is high.

Another interesting approach is PPay [YGM03]. PPay is a lightweight micropayment scheme for P2P systems, at least from the point of view of the users. The main drawback with PPay is that it uses a central server (called broker) when the issuer of a coin is off-line. This means that when a user a, who owns a coin minted by m who is off-line, wants to spend it at the user b, a should make the transaction via a central broker. In some frameworks, were the ratio of users off-line is high or in very dynamic systems were users join at some point and never reconnect again, the probability of finding the original issuer of the coin on-line is very low. In this situation PPay converges to a system with a centralised accounting bank.

1.2 Our contribution

We present a completely decentralised, off-line karma implementation for P2P and grid systems, that detects double spending and other types of fraud under varying adversarial scenarios. The system is based on the tracing of the spending pattern of coins, and distributing the normally central role of a bank over a predetermined, but random, selection of nodes. Transactions between users do not require the cooperation of this distributed bank. Instead, karma coins need to be occasionally reminted to detect fraud. The system is designed to allow nodes to join and leave the system at arbitrary times.

[2] MojoNation no longer exists, but the website has been archived. See web.archive.org/web/*/mojonation.net/*.

2 Model and Assumptions

We assume a distributed system where nodes join and leave an overlay network like CAN [RFH+01], Chord [SMK+01], and many others. In the context of this paper we want to stay abstracted from the underlying overlay network. We are going to model common characteristics that apply to routing overlays as in [CDG+02].

In this abstract model, every node is assigned a uniform random identifiers. We assume that the overlay network provides reliable and secure primitives for user look-up, message routing and storing data. Furthermore, each node maintains a *neighbour set* which consist of a set nodes *close* to this node in the node identifier space.

In our threat model, we assume that at most t nodes of the n nodes participating in the overlay are controlled by the adversary. We distinguish between nodes that have been taken over by adversary *after* they joined the overlay (in effect allowing the adversary to subvert a *virtual* node of his choice), and nodes that were taken over by the adversary *before* they joined the overlay (in which case the adversary gets to control a random virtual node whose identifier is not under the adversaries control). We assume at most c of the faulty nodes can been subverted by the adversary after they joined the overlay.

Finally, the requirements on a usable karma system for P2P and grid applications are the following.

Scalability Transaction cost should be independent of the size of the network.

No centralised control The system should not rely on one or several central, special, nodes (e.g., banks or brokers) and should not require any predetermined hierarchy.

Load Balance The load on any node should be proportional to the number of transactions it engages in

Availability Transactions among users can be processed uninterrupted even even when users are joining or leaving the system, and even when a set of users suddenly loose their connection.

Double-spending detection The system must detect double spending, and for every double spent coin, a fraudulent user should be blacklisted[3].

3 Protocol Sketch

The system manages the minting and transfer of karma coins. Coins can be minted by a user by finding collisions on a hash function (a la hashcash [Bac97]). A minted coin contains the name of the minting user as well as a sequence number (limiting the number of coins a single user can mint). User identity and sequence number together constitute the unique coin identity.

The coins are transferable [CP93]. A user can pay for resources by transferring a coin to another user. The sender signs the coin, and the receiver verifies this signature and stores the coin (with signature) for further use. With every transfer, a coin is extended with another signature. Thus, the sequence of signatures on a coin record the payment history of that coin. Double spending is detected by comparing the history of two coins with the same coin identity, and the culprit (or his accomplice) will be found at the node where both histories fork. This check is performed whenever a coin is reminted.

Every once in a while (but at least before the validity period of the coin expires), coins must be reminted. Reminting is used to detect double spending, and at the same time to reduce the size of the coin by removing its history. The reminting user asks a set of *remin-*

[3] We note that for any system offering off-line currency, double-spending *prevention* is generally speaking not possible, unless extra assumptions (e.g., special tamper proof hardware) are made.

ters to do so. The set of reminters is constructed in such a way that

- at least one of the reminters is a non-corrupted node, and
- all honest reminters possess the history of previously reminted coins with the same identity.

Together, these requirements ensure that any double spending activity will be detected. Timestamps are protected such that the adversary cannot escape reminting. If the reminters do not detect any fraudulent activity regarding the coin to be reminted, they strip the history from the coin and store it in their database. Then they sign the reminted coin using a multisignature [OO99]. Another way to view this is that each coin is protected by a unique, random looking, distributed bank.

The set of reminters is chosen solely based on the identity of the coin to be reminted, using a hash function to ensure that the set of reminters is random to the adversary. In fact, the hash of the coin identity is used as a reference to a virtual node in the overlay network, and the overlay network is queried for the current neighbours of that virtual node.

Of course, nodes join and leave the network, possibly changing the remint set for a particular coin. Therefore, with every join or leave, nodes update their database to ensure that it contains the history of all coins for which that node is in the remint set. Sufficiently old histories (for those coins for a which a remint request can no longer be received because their validity has expired) are automatically pruned from the database. This keeps the database bounded.

With such a dynamic method of assigning a remint set to a coin, there is a risk that the adversary is free to select a remint set of his own choosing (containing only corrupted nodes). This is avoided by requiring that the remint set is large enough, and that it does not contain nodes with a virtual overlay identity too far from the hashed coin identity. In fact we prove that if the remint set is larger than $\alpha s + c$ (for some constant α and c being the number of nodes that can be subverted by the adversary after they join the overlay), than the probability of undetected fraud is negligible in s.

4 Conclusions

We have outlined the principles underlying the first system for truly off-line karma coins, that can be used in highly dynamic peer-to-peer networks and grid-systems. Several interesting research questions remain. For instance the length of a coin increases with every transaction, and involves several public-key cryptographic operations. This is quite heavyweight, in contrast with micropayment schemes that are usually associated with the kinds of value transfers we consider here. The exposition has necessarily been sketchy, but details on the protocol, as well as an elaboration on open areas of research can be found in [GH04].

References

[Bac97] BACK, A. Hashcash. www.cypherspace.org/hashcash, 1997.

[CDG+02] CASTRO, M., DRUSCHEL, P., GANESH, A. J., ROWSTRON, A. I. T., AND WALLACH, D. S. Secure routing for structured Peer-to-Peer overlay networks. In *Proceedings of the 5th ACM Symposium on Operating System Design and Implementation (OSDI-02)* (New York, 2002), Operating Systems Review, ACM Press, pp. 299–314.

[CP93] CHAUM, D., AND PEDERSON, T. Transferred cash grows in size. In *EUROCRYPT '92* (Balatonfüred, Hungary, 1993), R. A. Rueppel (Ed.), LNCS 658, Springer, pp. 390–407.

[Coh03] COHEN, B. Incentives build robustness in bittorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems* (Berkeley, CA, USA, 2003).

[GH04] GARCIA, F. D., AND HOEPMAN, J.-H. Off-line karma: Towards a decentralized currency for peer-to-peer and grid applications. (to appear), 2004.

[GMA+95] GLASSMAN, S., MANASSE, M., ABADI, M., GAUTHIER, P., AND SOBALVARRO, P. The MilliCent protocol for inexpensive electronic commerce. In *Fourth International Conference on the World-Wide-Web* (MIT, Boston, 1995).

[Kir] KIRK, P. Gnutella. rfc-gnutella.sourceforge.net.

[NLRC98] NISAN, N., LONDON, S., REGEV, O., AND CAMIEL, N. Globally distributed computation over the internet – the POPCORN project. In *18th International Conference on Distributed Computing Systems (18th ICDCS'98)* (Amsterdam, The Netherlands, 1998), IEEE, pp. 592–601.

[OO99] OHTA, K., AND OKAMOTO, T. Multisignature scheme secure against active insider attacks. In *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* (1999), pp. E82–A(1): 21–31.

[RFH+01] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. A scalable Content-Addressable network. In *Proceedings of the ACM SIGCOMM 2001 Conference (SIGCOMM-01)* (New York, 2001), R. Guerin (Ed.), vol. 31, 4 of *Computer Communication Review*, ACM Press, pp. 161–172.

[Riv04] RIVEST, R. L. Peppercoin micropayments. In *Proceedings Financial Cryptography '04* (2004), A. Juels (Ed.), vol. 3110 of *Lecture Notes in Computer Science*, Springer, pp. 2–8.

[RS97] RIVEST, R. L., AND SHAMIR, A. PayWord and MicroMint: Two simple micropayment schemes. In *Proceedings 1996 International Workshop on Security Protocols* (Cambridge, United Kingdom, 1997), M. Lomas (Ed.), vol. 1189 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Germany, pp. 69–87.

[SMK+01] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications* (2001), ACM Press, pp. 149–160.

[VCS03] VISHNUMURTHY, V., CHANDRAKUMAR, S., AND SIRER, E. G. KARMA: a secure economic framework for peer-to-peer resource sharing. In *Proceedings of the Workshop on the Economics of Peer-to-Peer Systems* (Berkeley, California, 2003).

[YGM03] YANG, B., AND GARCIA-MOLINA, H. PPay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS-03)* (New York, 2003), V. Atluri and P. Liu (Eds.), ACM Press, pp. 300–310.