Performance Challenges of Decentralised Services

Greig Paul
Department of Electronic
& Electrical Engineering
University of Strathclyde
Glasgow, UK
Email: greig.paul@strath.ac.uk

Pierre-Louis Dubouilh
Department of Electronic
& Electrical Engineering
University of Strathclyde
Glasgow, UK
Email: gfb13141@uni.strath.ac.uk

James Irvine
Department of Electronic
& Electrical Engineering
University of Strathclyde
Glasgow, UK
Email: j.m.irvine@strath.ac.uk

***Abstract*—Decentralised, peer-to-peer based services present a variety of security and privacy benefits for their users, and highly scalable to cater for a growing numbers of users, without extra servers being required of the service operator. This presents a significant advantage for newly emerging mobile applications (with high numbers of users, and limited funds for infrastructure), although performance is a challenge when accessing decentralised services. In this paper, we firstly show the performance of our implementation of a decentralised chunk-based storage platform is constrained by the network. We show the impact of network latency on the performance of this decentralised storage solution, and propose our solution to this, in the form of a federated, intermediary server, thus creating a hybrid decentralised service. This approach offers relatively constant performance as latency increases, due to the use of TCP connectivity, while ensuring the advantages of the decentralised service are not lost in the process.**

I. Introduction

Since the idea of a distributed hash table (DHT) was first presented, as Content-Addressable Network (CAN) [1], Chord [2], Pastry [3] and Tapestry [4], the potential for developers to implement services without fixed server infrastructure has arisen. The distributed nature of a DHT means that data required for the service is held in a given location at a logical (rather than physical) location on the distributed network. These DHTs are also peer-to-peer in their nature, with data held by the users of a service, for retrieval by other service users. Since there is no individual (or group) in control of every user of the service, it is by its nature therefore decentralised, since it is not possible for one user to prevent others from making use of the service.

To store a value within a DHT, the cryptographic hash of the value to be stored is taken (per Figure 1). It should be noted that a hash function ($H(x)$) is a one-way function, in that while it is easy to calculate $H(x) = A$, it is computationally infeasible to calculate $H^{-1}(A)$. Hash functions also produce a uniformly distributed output across their range. Data is stored within the hash table at an address corresponding to its hash. This means that it is possible to retrieve a large quantity of data from the DHT, knowing only its hash.

The DHT itself is decentralised, with no one entity or actor in control of the network. A DHT can be considered as a key-value pair store, where the value (the data to be stored) is held at the address of the key ($H(value)$). Keys are uniformly distributed throughout the DHT, as $H(value)$ is uniformly distributed, and the nodes required to store a given piece of data are those whose logical DHT addresses are numerically closest to the hash of the data to be stored.

Fig. 1. Operation of a cryptographic hash function

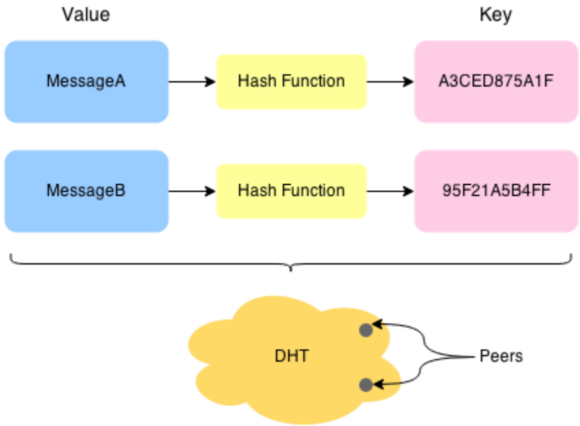
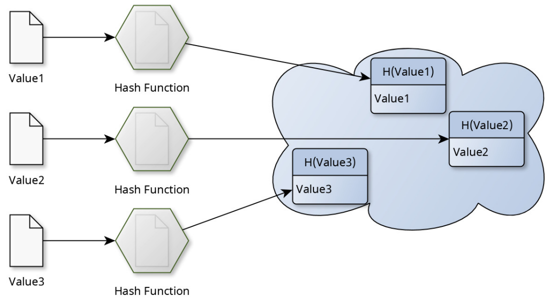


Fig. 2. Operation of a distributed hash table



Despite decentralised services being well-established and researched [5], they have failed to gain significant traction and deployment. Indeed, services which were originally developed and operating using decentralised services, such as Spotify [6] have even begun to retreat from decentralised peer-to-peer technologies back to conventional server-based architectures, often as a result of the decreasing costs of operating cloud-based servers [7]. Despite this, there are significant cost savings involved for a start-up using decentralised technologies such as peer-to-peer, compared to conventional techniques. Conventional service architectures would require significant server resources (and thus considerably investment), either in hardware or as payments to the owner of such hardware.

II. Motivation

Decentralised services present many advantages for users, such as those discussed in [8]. In particular, decentralised services offer users reassurance that access to their own data cannot be terminated by the operator (as the operator no longer has the ability to prevent users from accessing their own data). Compared with centrally hosted *cloud-based* services, users have an extra level of reassurance that their data is not accessible to the operator of the service (for data mining or analysis, perhaps without their knowledge or consent). The ability to implement such decentralised services on a DHT-based network is apparent, with the emergence of platforms such as MaidSafe [9], Storj [10] and Box2Box [11]. Despite this, there remain unresolved questions as to the practicality of such platforms for user-facing services.

In recent years, smartphones have become the primary means of consuming internet-based content for many people. Indeed, in January 2014, mobile apps (not including mobile-based web browsers) were used to access the internet more than desktop or laptop computers in the USA [12]. In developing countries, mobile phones are proving particularly popular, with 1.9 billion worldwide high-speed smartphone internet subscriptions in 2014 [13]. With 6.7 billion total worldwide mobile phone subscriptions [13], it is clear that while mobile phones are wide-reaching, yet are not always able to offer high speed connectivity. As such, it is important when designing services to consider usability of those who may not have the most reliable connection.

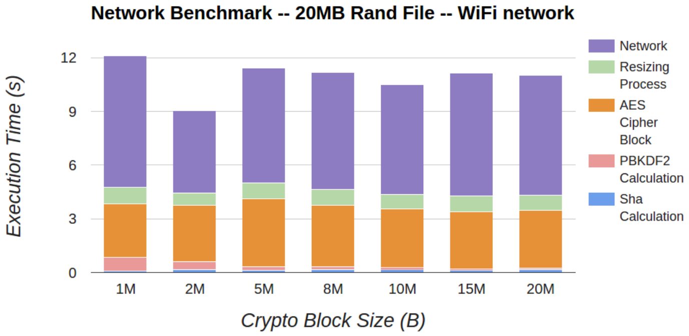
Given the growth of the use of mobile devices to access the internet, often exclusively, in the case of users in developing countries, consideration should be made as to the performance of DHT-based solutions on mobile devices (and networks). Additionally, the often-unreliable performance of mobile-based networks poses a concern, and past research has attempted to address the problems of mobile access to DHT-based networks [14], [15].

III. Performance of DHT-Based Storage

As described in [8], it is possible to create a wide variety of services upon a decentralised, storage-based infrastructure, removing the need for reliance upon a single central service provider, and giving users greater assurance as to the security of their own data. In order to assess the viability of such a service architecture for use by users on real-world networks, we created an implementation of a storage system, built using our model described in [8].

Typically, DHT-based applications use a UDP-based transport protocol, on account of the significant overhead needed when establishing TCP connections with new nodes, which are used only briefly before being closed. As such, UDP was used as the transport protocol for all communications within our storage service. Inkeeping with typical packet-size limitations on UDP, we constrained all UDP packets to 8100 bytes, which was the default maximum UDP packet size in the Python Twisted networking library, and implemented fragmentation/defragmentation support in our client.

Fig. 3. Time to carry out each stage of the data upload process



To upload a file to the storage network (modelled around the MaidSafe network, as described in [16]), The process consisted of 5 discrete steps:

- Input file is split into chunks
- Each chunk is hashed with SHA512
- Securely derive a key from this chunk hash (PBKDF2)
- Symmetrically encrypt each chunk using that key
- Fragment chunks for transmission over UDP
- Transmit each chunk to the DHT for storage

Figure 3 shows the time taken to upload a 20 MB file (containing uncompressable pseudo-random data), based on different file chunk sizes (i.e. for the first step of the above process) over a standard home WiFi router, to a server on the same network. Note that each chunk underwent fragmentation for transmission over UDP, irrespective of the chunk size. From these results, it is clear that the most time-consuming stage of this process is the network transmission, on account of the UDP-fragmented data. By transmitting around 2600 UDP packets (at 8100 bytes each), any loss would result in delays while recovering (by re-transmitting, on account of no acknowledgement being received).

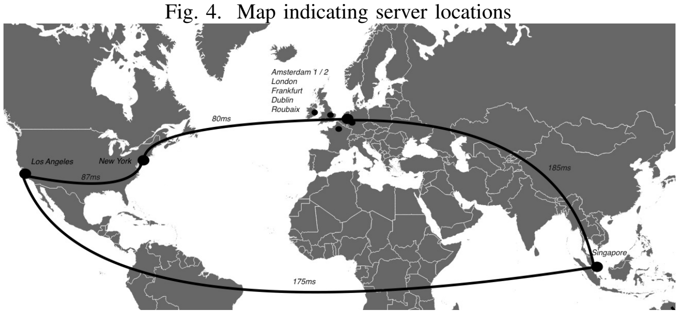
IV. Impact of Latency

Packet latency is an unavoidable and undesirable property of networks, although one which is highly relevant in decentralised networks. Since users are storing their data in a logically distributed manner, where users in close proximity geographically will statistically be uniformly distributed throughout the network, latency between users is significantly higher than that experienced connecting to a nearby centralised server. By creating a test network, featuring several nodes located in different data centres around the world, it was possible to investigate the impact of network latency, based on real-world values which would be achieved by users of a commercially deployed decentralised service. In order to reduce as much as possible the influence of network interface speed, each DHT node had an interface speed of 1 Gigabit per second.

For the purpose of more readily reproducible and comparable results, these latencies (varying from 8 ms between London and Amsterdam, and 300 ms between London and Singapore) were used to select a suitable range of synthetic latencies (which were imposed via the Linux iptables firewall).

The server locations used for calculating these latencies are shown in Figure 4. We found that the highest typical latency was encountered between Dublin and Singapore, at around 330 ms. Bearing in mind that these figures are for latencies between high-capacity data centres, rather than users' domestic internet connections, it is clear that latency is, and will be, a significant concern when implementing decentralised services.

Fig. 4. Map indicating server locations

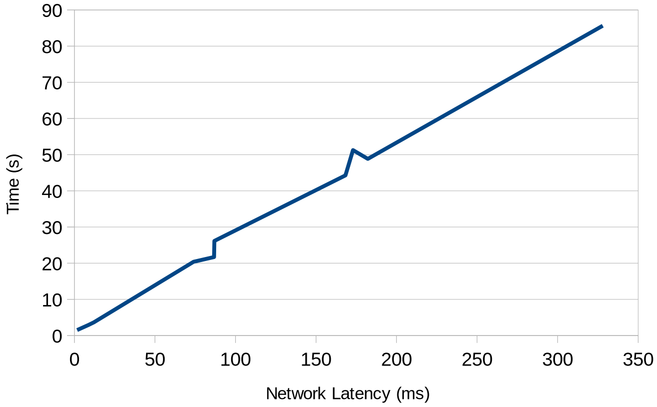


To establish the impact of latency on performance, while taking into account the size limitations of UDP packets (as discussed in Section III, we carried out a number of experiments to transfer 1 MB over a variety of channels with varying latency. Figure 5 shows clearly that (as one would expect), the time to upload 1 MB of data increases linearly with latency, on account of the increase in waiting time between chunks being sent, in order to ensure they were correctly received, prior to proceeding with later chunks.

V. Techniques to Reduce Impact of Latency

In order to reduce the impact of latency, it is possible to consider a hybrid approach between centralised and decentralised services. By allowing a mobile device to directly communicate with an intermediary server, which carries out operations on the DHT itself, on behalf of a mobile user, it is possible to significantly improve the performance of the operation of our storage network. It is possible to achieve this, while still preserving complete confidentiality of user data, by using delegated instructions, such that the intermediary server need never have access to signing or decryption keys of the client.

Fig. 5. Time to upload 1 MB of data as 8 KB UDP fragments



To demonstrate the performance improvements of this approach, we created an implementation of such an intermediary server, which presented a standard web-based API to a mobile device, and carried out DHT-based requests on behalf of the client. The client provided the intermediary server with a signed request, which was simply relayed by the server to the DHT. In return, the intermediary server queues (and then delivers, when the client next connects) the signed acknowledgement responses from the DHT nodes holding the data. Since the intermediary server is not required to reliably hold any data, and any failure to correctly relay requests is easily detected, it is not necessary for a user to trust an intermediary server. Indeed, our hybrid service supports federation, such that a client may connect to any known intermediary server it is aware of, and these servers will share any necessary messages, such that they reach clients.

By federating these intermediary servers outwith the DHT (we implemented a basic automated discovery mechanism), it is possible for a user to connect to any intermediary server. We suggest that in future, if widely deployed, a competitive marketplace would develop for the provision of relaying services, with reliable and high-speed relays available for those willing to pay slightly more to a service operator.

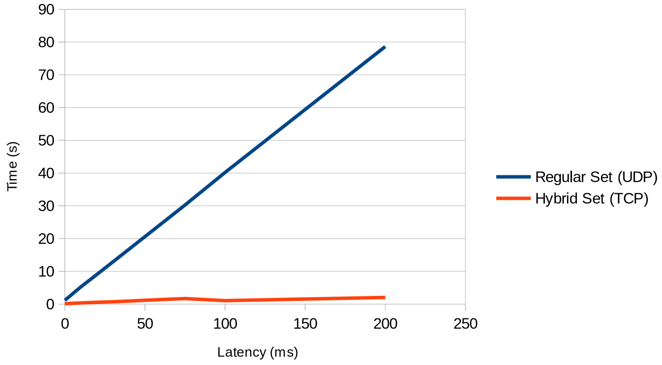
Figure 6 shows the performance gain experienced by the client, as a result of offloading the DHT processing to the intermediary server, using our hybrid approach. Our hybrid solution, using an intermediary server to offload DHT operations, gave relatively constant performance with increases in latency, due to the performance benefits of TCP windowing.

Using a link with a 10 ms latency, the overall time to transfer 1 MB (in packets of 8 KB) was 5.24 seconds, using a regular set operation over UDP. In contrast, our hybrid approach over TCP took took 0.36 seconds to achieve the same task. At higher latencies, the hybrid solution offers a more significant performance improvement. At a 200 ms latency, the UDP-based approach took 78 seconds, while the hybrid approach took 2 seconds. This performance improvement is particularly significant for users of mobile devices, where the time taken to transmit data directly impacts the battery life of the device. By offering a TCP relay, these users will experience significantly better performance, as well as reduced network activity, and thus improved battery life.

VI. Conclusion

We have found that decentralised services are highly sensitive to network latency, on account of the need to send large numbers of UDP packets from client devices to DHT

Fig. 6. Time to upload 1 MB of data (8 kB fragments)



nodes, which may well be on the opposite side of the world, experiencing significant packet latencies. We have shown that the time for a mobile device to upload data to the DHT increases linearly with network latency, and that the time to upload the data is high, even for small quantities of data, due to the very limited packet sizes of UDP. By utilising a hybrid decentralised approach, where an intermediary server acts on behalf of a user (without any ability to decrypt user data), we have shown that it is possible to achieve relatively constant performance, even in the presence of high network latency.

We implemented an intermediary server which had no access to the underlying data being accessed by a client, and was not able to forge requests on behalf of the client. In the event of the intermediary being unreliable or untrustworthy, and not properly relaying requests to the DHT, the client can detect this on account of the missing (or forged) acknowledgement receipts for the data which was to be stored, allowing the client to select another intermediary server. Since this intermediary server was designed to operate in a federated manner, it is possible for a travelling user to select a geographically close intermediary server for optimal performance.

Acknowledgements

This work was funded by EPSRC Doctoral Training Grant EP/K503174/1 and MaidSafe.net.

References

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '01. New York, NY, USA: ACM, 2001, pp. 161–172. [Online]. Available: http://doi.acm.org/10.1145/383059.383072

[2] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 1, pp. 17–32, Feb 2003.

[3] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001*. Springer, 2001, pp. 329–350.

[4] B. Y. Zhao, J. Kubiatowicz, A. D. Joseph *et al.*, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," 2001.

[5] L. Harn and H.-Y. Lin, "Key management for decentralized computer network services," *Communications, IEEE Transactions on*, vol. 41, no. 12, pp. 1777–1779, Dec 1993.

[6] G. Kreitz and F. Niemela, "Spotify–large scale, low latency, p2p music-on-demand streaming," in *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*. IEEE, 2010, pp. 1–10.

[7] R. Dillet. (2014, April) Spotify removes peer-to-peer technology from its desktop client. Techcrunch. Retrieved 5th March 2015. [Online]. Available: http://techcrunch.com/2014/04/17/spotify-removes-peer-to-peer-technology-from-its-desktop-client/

[8] G. Paul and J. Irvine, "5G-enabled decentralised services," in *81st Vehicular Technology Conference (VTC2015 Spring)*. IEEE, May 2015.

[9] Maidsafe.net. (2014, November) Maidsafe - distributed platform overview. [Online]. Available: http://maidsafe.net/overview

[10] Storj - the future of cloud storage. Storj.io. Retrieved 16 February 2015. [Online]. Available: http://storj.io/

[11] A. Lareida, T. Bocek, S. Golaszewski, C. Luthold, and M. Weber, "Box2box - a p2p-based file-sharing and synchronization application," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, Sept 2013, pp. 1–2.

[12] J. O'Toole. (2014, February) Mobile apps overtake PC internet usage in U.S. CNN. [Online]. Available: http://money.cnn.com/2014/02/28/technology/mobile/mobile-apps-internet/index.html

[13] (2014, September) The state of broadband 2014: Broadband for all. The Broadband Commission. [Online]. Available: http://www.broadbandcommission.org/Documents/reports/bb-annualreport2014.pdf

[14] H. Pucha, S. M. Das, and Y. C. Hu, "How to implement dhts in mobile ad hoc networks," in *Proc. of the 10th ACM International Conference on Mobile Computing and Network (MobiCom 2004)*, 2004.

[15] I. Kelenyi and B. Forstner, "Distributed hash table on mobile phones," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. IEEE, 2008, pp. 1226–1227.

[16] G. Paul and J. Irvine, "Security of the MaidSafe Network," Wireless World Research Forum. WWRF32, May 2014. [Online]. Available: http://strathprints.strath.ac.uk/48569/