學號：946350
清華大學　黃世傑
論文：OverCite: A Distributed, Cooperative CiteSeer
Conference: NSDI 06, IPTPS 05 (A previous design work)
Authors:
Jeremy Stribling, Jinyang Li, NYU and MIT CSAIL, via UC Berkeley
Isaac G. Councill, Pennsylvania State University (PSU)
M. Frans Kaashoek, Robert Morris, MIT Computer Science and AI Laboratory

(a) Problem statement
這篇論文是一篇偏向實作的論文。這篇論文在 05 年有出現在 IPTPS (International workshop on Peer-to-Peer Systems)，論文的標題與在 NSDI 06 的不太一樣 (OverCite: A Cooperative Digital Research Library)，在 IPTPS 主要是描述它的設計部份，而在 NSDI 06 除了重新把問題描述一次之外，也加上了他們實作的部份，現在他們的實作品已經可以在網路上使用(http://overcite.org)。

目前在 CS 領域的人尋找論文，有幾個管道，比如像是 Google Scholar，CiteSeer，或是透過 ACM Portal。這篇論文針對 CiteSeer 提出了幾個重要的數據，表示 CiteSeer 的流量相當的大(Table 1)，每日的平均流量高達 34GB，且 CiteSeer 提供了 cache data，所佔用的空間達到 803GB，而目前管理 CiteSeer 的賓州大學沒有很好的解決方法，雖然很多人願意免費提供硬體支援，但無奈 CiteSeer 的設計是 centralized 的設計，即使很多人願意提供資源，但卻難以整合進目前運行的 CiteSeer 中。

Number of papers	674,720
New documents per week	1000
HTML pages visited	113,000
Total document storage	803 GB
Avg. document size (all formats)	735 KB
Total meta-data storage	45 GB
Total inverted index size	22 GB
Hits per day	800,000
Searches per day	250,000
Total traffic per day	34 GB
Document traffic per day	21 GB
Avg. number of active conns	68
Avg. load per CPU	66%

Table 1: Statistics of the PSU CiteSeer deployment.

目前 CiteSeer 已經很容易出現 System Too Busy 的訊息，所以如果想讓 CiteSeer 繼續運行下去，找出一個解決法是必要的，此篇 Paper 就是一個實作品。目標是讓 Load-Balance，且未來加入新的硬體資源也需較爲容易(Scalable)。

(b) Solutions and results briefing
此篇論文認爲，使用 centralized 的方式就是問題的主因。所以他們提出了使用 decentralized 的設計，利用了近幾年來在 structured P2P 已經實際在使用的技術，

稱爲 Distributed Hash-Table (DHT)。DHT 的主要概念爲，每個人都使用同樣的一個 Hash Function(目前常用的是 SHA-1)，將自己的一些 attribute 做 Hash 後即可得一 unique ID，比如每個人的 IP 都會不同，就將自己的 IP 給 Hash 得 unique ID，在這裡不去考慮 collision 的情況。每個 Node 都得到 UID 後，再將自己手上有的文件也做 Hash，有可能將整份文件做 Hash，也有可能只 Hash 檔名，看應用而不同。如此一來網路上的電腦與文件都會有一個 UID，而文件就由和自己的 UID 最相近的 Node 來管理（存放）。

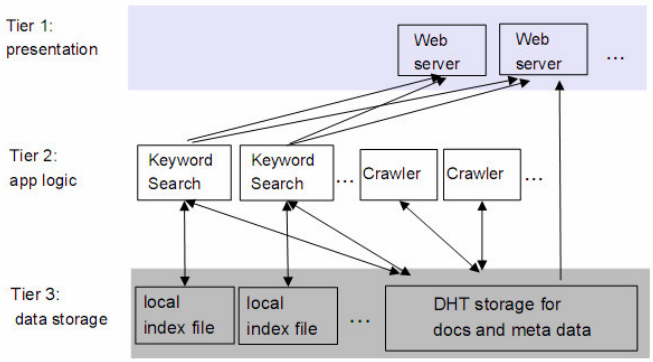
上述的是 DHT 的原理，此篇論文根據“論文”的特性先設計了一個 data structure(Table 2)，像是標題(Title)，作者(Author)，引用到的論文(Cites)・・・等，就會有對應的 Key ID，這些資料合稱爲論文的 metadata，儲存到上述的 DHT 結構中，而論文的本身也存到 DHT。簡而言之，DHT 的角色就是一個分散式儲存的架構，讓每個提供儲存資源的電腦不需要一個 centralized 的機制去做管控，而可以透過分散式 Hash Table 的特性做 Self-Organization。Self-Organization 的好處在於，未來有新的硬體支援要加入的話，新加入的硬體也一樣會得到 UID，透過 DHT 底層的溝通機制，會做 take over 的動作，若有文件的 UID 更接近此新加入的 Node，就會由此 Node 接管，這個特性就解決了上述提到 centralized 設計 scalability 不佳的問題。而 DHT 本身就是分散式儲存，在此是假設 Hash Function 的分佈是很平均的分佈，一般密碼學用到的 hash function 會有這種特性。

Name	Key	Value
Docs	DID	FID, GID, CIDs, etc.
Cites	CID	DID, GID
Groups	GID	DID + CID list
Shins	hash(shingle)	list of DIDs
Crawl		list of page URLs
URLs	hash(doc URL)	date file last fetched
Titles	hash(Ti+Au)	GID
Files	FID	Document file

Table 2: The data structures OverCite stores in the DHT.

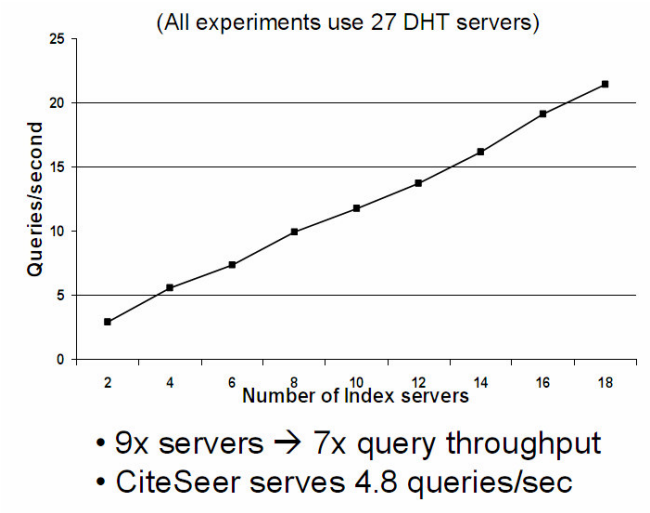
在此不詳述仔細的實作過程，因爲這些細節是根據要實作的應用而不同的，而實質上都是 DHT 的應用。

設計的架構就是一般我們所知的 Three-Tier 架構，也就是 MVC (Model-View-Controller)架構，如下圖所示。



在 Tier 1 就是 View 的部份，可以透過 Round-Robin DNS 的方式讓 client 依序連到不同的 Web Server，而使用者的輸入會傳遞到 Keyword Search Engine，Tier 2 即會做 DHT 的解

析，去 Tier 3 找到需要的資料。另外，Crawler 也存在於 Tier 2 中，會自己去網路上找論文，若找到新的論文就會解析出 metadata，並且將 metadata 與論文本身存到 DHT storage 中。



目前 OverCite 已經實際在運行中，根據分析數據發現，提供 9 倍數量的 server 即可提高 7 倍的 query/sec，效果相當的好。

(c) Comments: provide your person thoughts and comments

這篇論文裡幾乎沒有什麼理論的部份，DHT 也已經是在 P2P 系統中常用的技術，但他們找到了一個很好的應用，在此之前我從來沒想到 CiteSeer 竟然本身有那麼大的問題，一般這種 Server 給人的印象就是反正只要有錢，硬體給他花大錢買下去就行了，但這篇論文卻指出，很多人願意幫助 CiteSeer，但卻無從幫起，我覺得很有趣。另外，我覺得他們成功的將一個 centralized 的服務做成 decentralized 的，這個 scenario 很值得學習，未來若有什麼 centralized 的服務要改造成 decentralized 的時，就有一個成功的案例可以依循，我覺得這是很有用的。