國立高雄科技大學(第一校區)

電子工程系

專題報告書

私有以太坊區塊鏈實作以及其硬體加速方法調查

專題生：

劉益彤 YI-TUNG LIU

指導教授：陳朝烈博士 Dr.Chao-Lieh Chen

中華民國 一一一 年 五 月

摘　　要

本專題主要在私有以太坊區塊鏈(Ethereum)的實作與硬體裝置進行加速運算，以太坊是一個去中心化、開源且支援智慧合約(Smart Contract)的公共區塊鏈平台，將資料上傳區塊鏈有著極高的安全性，還有著可以透過去中心應用程式(Dapp)擴充區塊鏈的功能，因此十分適合組織或大型企業使用。

由於區塊鏈有著非常繁瑣的運算機制，透過OpenCL(**Open** **C**omputing **L**anguage)讓CPU和硬體加速裝置互相協同工作，達到加速整個區塊鏈發展的目標。

目　　錄

摘要......................................................i

目錄.....................................................ii

圖目錄..................................................iii

第一章 緒論...............................................1

1.1 研究動機與目的....................................2

1.2 專題相關工具......................................3

第二章 研究架構與方法.....................................4

2.1系統架構........................................5

2.2智慧合約架構..................................7

2.3硬體加速方法調查...........................8

第三章 成果展示............................................9

3.1區塊鏈節點間同步............................10

3.2 去中心化應用程式..................................11

3.3硬體加速運算之差異..........................12

第四章 未來展望.........................................13

第五章 參考文獻..........................................15

圖目錄

圖一 SOC-7666...................................2

圖二 adventure-ARK-3530.............................................2

圖三 樹梅派..............................................2

圖四 電腦主機...............................................3

圖五IPCam........................................3

圖六 繼電器.............................................3

圖七 階層圖.............................................4

圖八 節點架構圖...........................................4

圖九 專題網路架構...........................................6

圖十 請求與資料傳遞路徑....................................6

圖十一 直播功能運作流程圖.....................................7

圖十二 控制流程............................................................7

圖十三 樣本與辨識模型傳遞路徑........................................8

圖十四 樣本與辨識模型傳遞路徑............................................8

圖十五 監控介面展示......................................10

圖十六 NG回報........................................10

圖十七 及時辨識........................................11

圖十八 ROI框選與樣本產生..........................................11

圖十九 請求樣本..........................................12

圖二十 接收到的樣本 .....................................12

圖二十一 訓練畫面........................................12

圖二十二 科技部110年徵案內容.....................14

第一章　　緒論

1.1研究動機與目的

以太坊目前是規模第二大的公共區塊鏈平台，其功能建立在資料分散於每個人手中，資料難以被竄改，再加上雜湊函式的特性，幾乎無法被逆向推導回來的特性。私有區塊鏈特別適合企業使用，將企業裡的裝置都連接上區塊鏈網路，就能避免有資料被竄改而不自知，又或者是透過智能合約建立自己的虛擬代幣，讓用戶可以透過該代幣購買企業的各種服務。

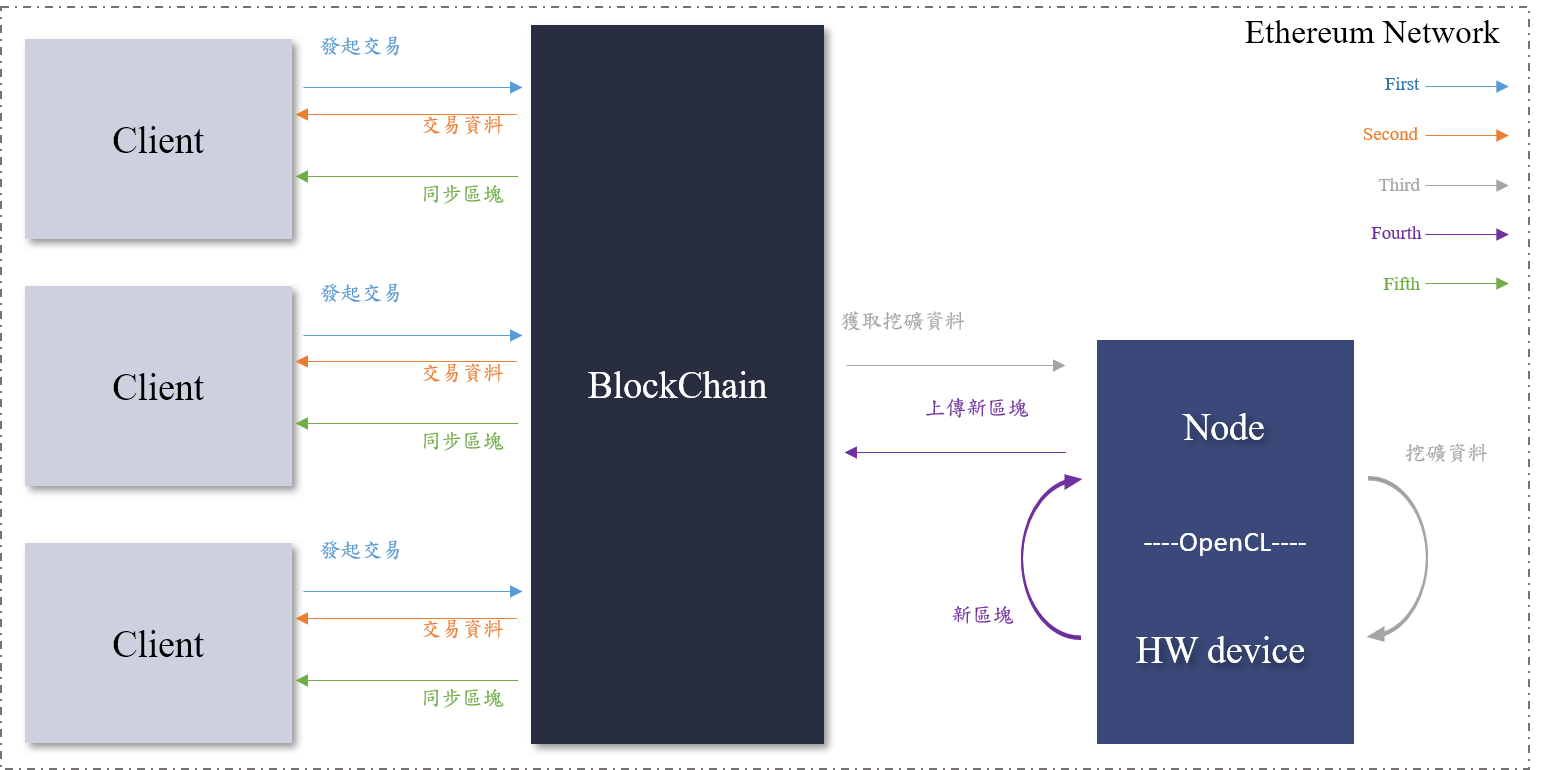
在以太坊上使用各種服務時，給予礦工手續費可以使自己的服務被優先處理，但隨著私有區塊鏈社群的成長，交易量也會越來越大，如果區塊鏈整體算力沒有隨之成長，在用戶們的競爭之下，手續費也會增多，長久以往對於整個社群不是一個好的發展，透過硬體裝置的加速，可以提升礦工的算力，把手續費平衡回來。

1.2 專題相關工具

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

第二章　　研究架構與方法

2.1 系統架構



圖二: 私有以太坊區塊鏈網路架構圖

* First : 用戶端(Client)發起交易。
* Second : 區塊鏈(BlockChain)回傳交易資料，但此時資料正在等待處理 ，查詢該交易資料後會發現，區塊鏈(BlockChain)並不知道該資料該放在哪個區塊(紅框)。



圖二: 處理前的交易資料

* Third : 節點(Node)透過區塊鏈(BlockChain)獲取需要進行處理的資料，再將資料傳送給旗下的硬體加速裝置(HW device)。
* Forth : 運算完成後將結果回傳給節點(Node)，節點(Node)在將結果回傳至區塊鏈(BlockChain)，這個步驟俗稱挖礦、出塊或打包。
* Fifth : 區塊鏈(BlockChain)將新區塊傳送給用戶端(Client)，用戶端(Client)同步新區塊資料，這個時候就能看到該交易資料被放在第1913個區塊裡。



圖三: 處理後的交易資料

2.2 智慧合約架構

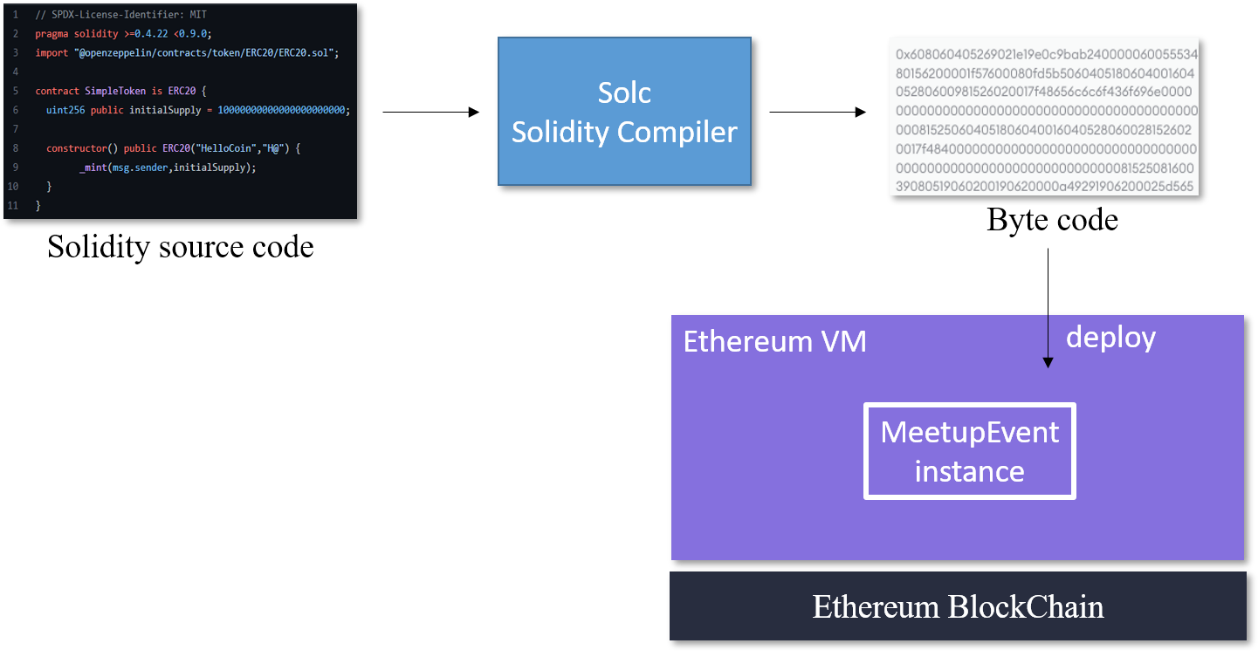


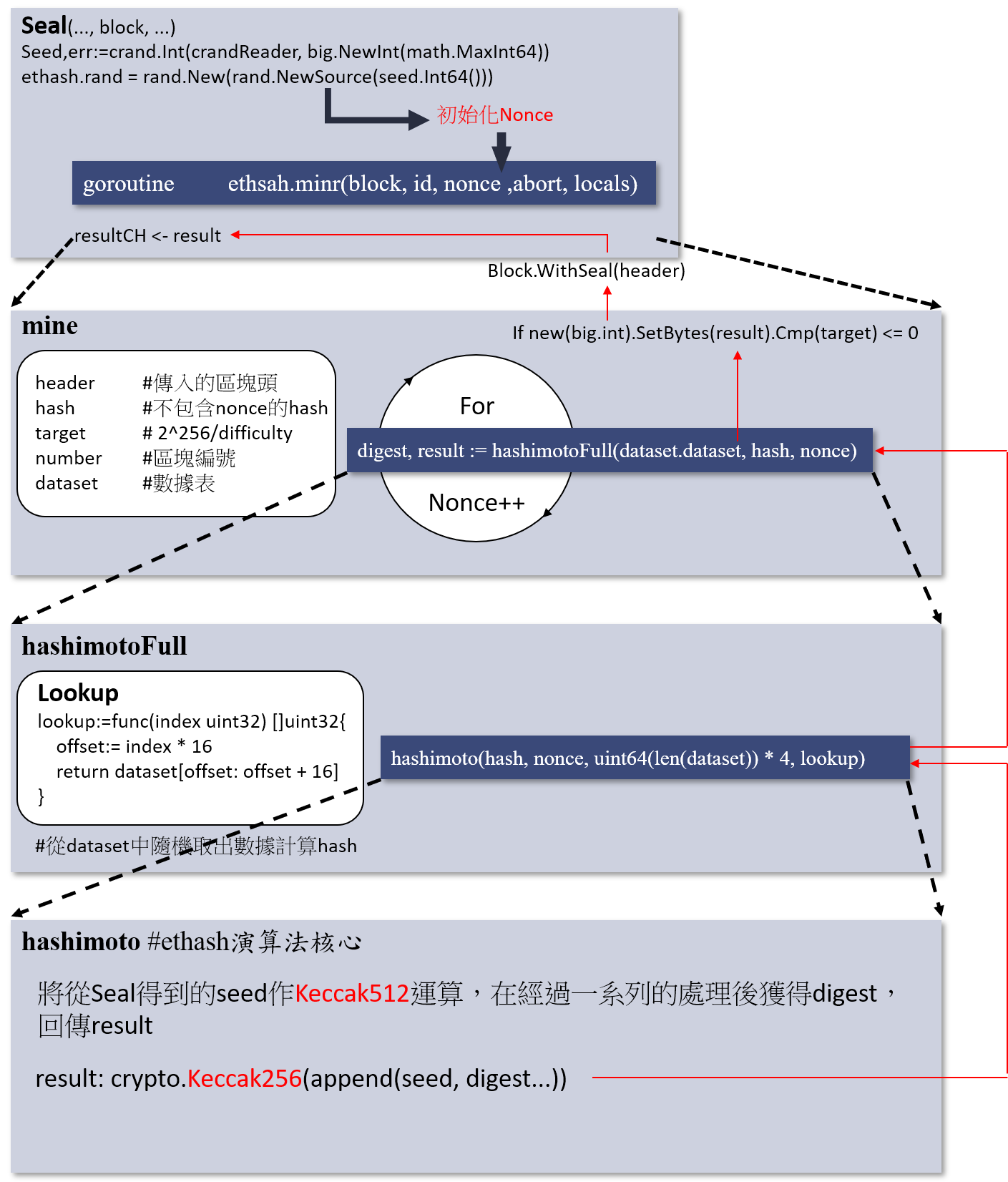
圖: 智慧合約架構圖

透過Solidity編寫ERC20智慧合約，創建名為HelloCoin的代幣，可以把這段程式碼當成Class。當我們要使用這段程式的時候，要先部署到區塊鏈上，Solidity原始程式會先經過Solidity編譯器，轉成該程式的Byte code， Solidity原始程式會被創造出一個Instance，這個Instance會被放在區塊鏈上，這個Ethereum虛擬機會運行在各個Ethereum的節點上。

以太坊部署合約的流程其實與普通交易沒有很大的差別，發起交易時把Byte code當成Input Data替換掉轉帳金額，接收者會被系統替換成區塊鏈本身，將合約部署到區塊鏈上，當合約部署成功後會得到一個合約地址，取得位址後搭配正確的interface資訊就可以執行此合約。

2.3硬體加速方法調查

圖: ethash挖礦流程圖



ethash的是根據Dagger-Hashimoto演算法更改而成，一開始會根據區塊資料生成seed，根據seed計算出一個16MB的隨機cache，由客戶端儲存，再根據cache生成一個大小為1GB的dataset，dataset的每一個數據的是藉由cache中的一小部分數據計算出來，由節點儲存，大小會隨時間線性成長，節點會將dataset中的資料隨機取出，取出的資料會用來計算hash，也就是圖中的最後一個部分 ” hashimoto ” 。

在hashimoto裡有兩個特別重要的部分，分別為Keccak512、Keccak256算法，Keccak又被稱為Sha3算法，所以Keccak512和Keccak256就等於Sha3-512、Sha3-256。

SHA-3第三代安全雜湊演算法(Secure Hash Algorithm 3)，雜湊(Hash)是將不定長度資料的輸入，轉換成固定長度的雜湊值，好的雜湊很少會出現衝突(由於雜湊演算法被計算的資料是無限的，而計算後的結果範圍有限，因此總會存在不同的資料經過計算後得到的值相同，這就是雜湊衝突)，在密碼學中，雜湊具有不可逆性。

Hash還有著微小的更動，導致出來的結果完全不同的特性，將本次專題的名稱最為輸入值，進行Keccak256運算:

私有以太坊區塊鏈實作以及其硬體加速方法調查 🡺

bc1972722bb76634a26667fc9ceba275cc210b7085816d1d56893a58ac5aa9a4

再將專題名稱後面加上123:

私有以太坊區塊鏈實作以及其硬體加速方法調查123 🡺

6facfbc84a43bdc77e05346ec7922c3ae1aaa9e942ac616dcbc968c3fc9ee9db

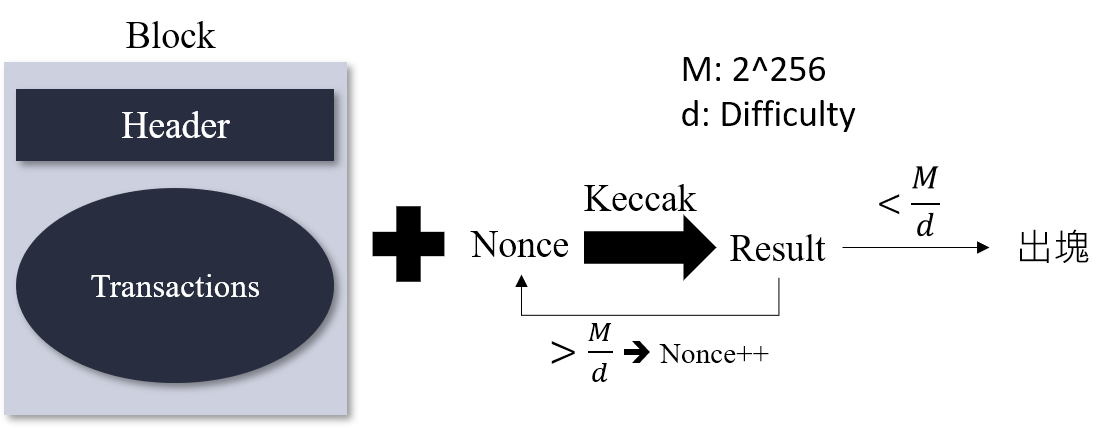


圖: 挖礦簡易流程圖

我們可以將整個區塊簡單地分成兩部分Header、Transactions，Header的資料有當前區塊編號、時間戳、被誰挖出來的、難度、區塊檔案大小、手續費和最重要的前一個區塊的Hash，Transactions是各式各樣的交易紀錄，Nonce其實只是一個配對值而已，由於Hash不可逆的關係，想要得出結果(Result)只能靠配對值進行暴力運算，Nonce每次加一都代表著Hash計算了一次，這也是區塊鏈算力的依據，每秒鐘算幾次Hash( H/s )，以圖為例，挖礦時必須將Header和Transactions加上Nonce進行Keccak運算，運算出來的值(Result)只要小於一定值( M/d )就代表找到答案了。

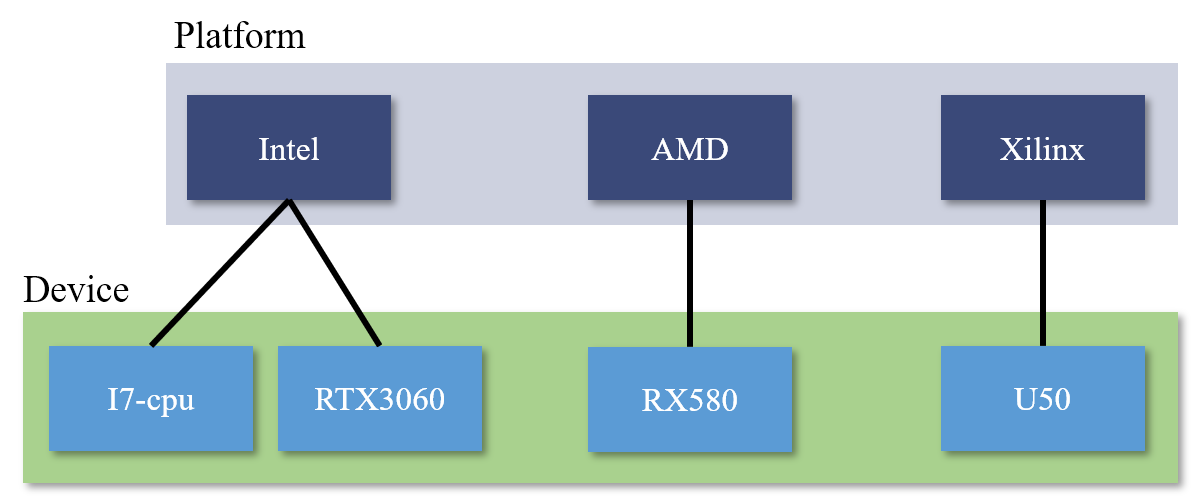


圖: OpenCL異構平台示意圖

OpenCL(**Open** **C**omputing **L**anguage)為一個異構平台，此平台可由CPU、GPU、DSP、FPGA或其他類型的處理器與硬體加速裝置所組成，在本次專題中，可以充分利用硬體加速裝置強大的並行計算能力以及與CPU的協同工作，更高效的利用硬體完成ethash演算法的核心 “ hashimoto “ 的部分，特別是Keccak512和Keccak256。

計算hash時，其中會有許多XOR的運算，而且nonce的值就代表著mine函數的迭代次數，mine會呼叫hashimotFull裡的hashimoto，換句話說nonce的值代表hashimoto裡的Keccak512和Keccak256被使用了幾次，此時硬體加速裝置特別適合用來替代CPU，我們可以使用OpenCL裡的kernel，針對不同的需求寫出不同的Kernel，使用hashimoto時讓硬體加速裝置把這個既不困難卻異常繁瑣的工作給處理掉。

區塊鏈因為區塊中有著代表前一個區塊的Hash，代表當前區塊的Hash會被放到下一個區塊的Header，區塊與區塊互相串連起來，所以就叫做區塊鏈。

區塊鏈有著很高的安全性，因為只要更動了前一個區塊裡的資料，不論是Header或Transactions，都會導致區塊的Hash改變，與下一個區塊對不上，下一個區塊對不上也會讓再下下一個區塊對不上，導致整個區塊鏈出錯，藉由這個特性，可以防止區塊鏈資料被竄改。

第三章　　成果展示

3.1區塊鏈節點間同步

3.2 去中心化應用程式

3.3硬體加速運算之差異

第四章　　未來展望與

第五章　　參考文獻