教育部獎助國立清華大學選送學生赴國外頂尖大學（機構）研修學生

成果報告書

Massive Program Parallelism on Parallel Processors

獲獎生姓名	Lai Wan-Shu
出國地點	Illinois, USA
研修大學	University of Illinois at Urbana-Champaign
研修系所	Department of Electrical and Computer Engineering
出國期間	September, 2007 – February, 2008

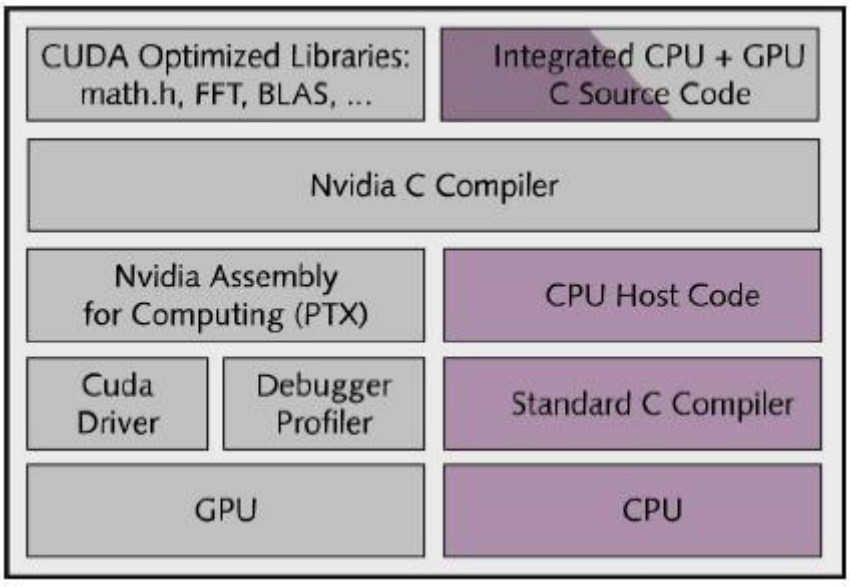
Abstract

Parallel Processing With CUDA

在多核心架構中撰寫有效率的平行程式，至今都還是學界與業界最大的軟體癥結，各家所推出琳瑯滿目的各種解法各有其優缺點：IBM's Cell BE, Intel's and AMD's x86…，具有彈性者卻需要開發者付出多的心力來控制繁瑣的細節，而較制式者雖然簡化程式開發程序與周期，卻又缺少了跨平台的優勢，也因此在程式平行化的爭奪攻防戰中，百家爭鳴。

NVIDIA 所提出的 General Purpose Graphic Processor Unit 概念，衍生出了 NVIDIA CUDA™，此為架構在 GPGPU 架構之上的 Parallel Programming Model，為 GPU 運算帶來強化功能。隨著 NVIDIA GPU 運算解決方案的發佈，同時提供 CUDA ™ 版本的 C 語言編譯器和 SDK，讓開發業者可使用繪圖處理器(GPU)開發運算應用程式。結合 GPU 運算技術和 CUDA 軟體開發環境，為現今對運算能力要求極高的資料密集型應用程式提供具彈性的大型平行運算平台。

下圖為 NVIDIA's CUDA platform for parallel processing on GPU 的架構是意圖，此 programming model 的優勢為只須撰寫與 CPU 相同執行的 C 原始碼，透過不同平台的 C compiler，便可編譯出 for CPU or GPU 的 machine code。CUDA 所提供的 FFT 和 BLAS 等標準數據函式庫可簡化程式開發，and hardware-abstraction mechanism 也隱藏了大部分底層硬體繁雜的細節，更可 simplified high-performance parallel software 的開發週期。



NVIDIA CUDA™ 可讓程式設計師快速設計解決複雜運算難題應用軟體之 C 語言編程環境，可完全發揮 GPU 的多核心平行運算處理效能。軟體設計師只需運用各種的 CUDA 軟體工具來加速他們的應用程式：從影音編碼乃至石油與天然氣探勘、產品設計、醫療成像，以及科技研究，即可發揮超級運算的威力。

Study Purpose

The success of parallel application development for multi-core and many-core processors will hinge upon the existence of strong automatic parallelized compilers, as well as effective manual parallel programming tools. Both types of tools require deep analysis techniques that discover coarse-grain parallelism in large applications written in standard programming languages - a task many in industry consider impossible. Both will also require techniques for automating the tedious performance tuning process needed for parallel applications to achieve scalability and performance goals. And The GPU is specialized for compute-intensive, massively data parallel computation. Based on above reasons, this project is to parallel LINPACT benchmark on CUDA platform. Furthermore, analyze the pros and cons of CUDA.

Study Method and Procedure

I. Description

LINPACK is a software library for performing numerical linear algebra on digital computers and it was intended for use on supercomputers in the 1970s and early 1980s. LINPACK makes use of the BLAS (Basic Linear Algebra Subprograms) libraries for performing basic vector and matrix operations.

The LINPACK Benchmarks are a measure of a system's floating point computing power. They measure how fast a computer solves a dense N by N system of linear equations Ax=b, which is a common task in engineering. The solution is obtained by Gaussian elimination with partial pivoting, with 2/3·N3 + 2·N2 floating point operations. The result is reported in millions of floating point operations per second (MFLOP/s, sometimes simply called FLOPS).

How does top500 uses LINPACK BENCHMARK to measure performance?

There are three benchmarks in LINPACK BENCHMARK:

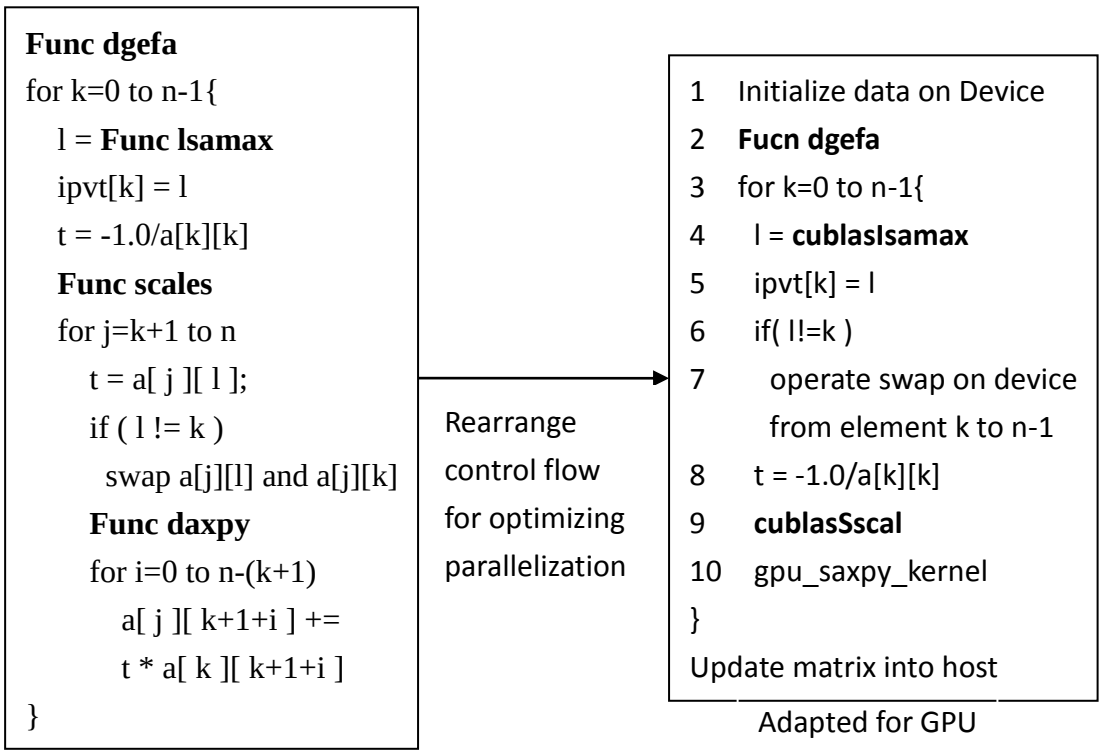
- First is LINPACK Fortran n=100 benchmark that is for a matrix of order 100 using in Fortran. The ground rules for running this benchmark are that you can make no changes to the Fortran code, not even to the comments. Only compiler optimization can be used to enhance performance. This Linpack benchmark measures the performance of two routines DGEFA O(n3) and DGESL O(n2)from the Linpack collection of software.
- Second is LINPACK n=1000 benchmark is for matrix of size 1000. The ground rules for running this benchmark are a bit more relaxed in that you can specify any linear equation solve you wish, implemented in any language. A

requirement is that your method must compute a solution and the solution must return a result to the prescribed accuracy.

- Finally is LINPACK's "Highly Parallel Computing" benchmark.

II. Implementation

Use the CUBLAS library on top of the CUDA, through this way to execute data on GPU instead of moving data to calculating on CPU. This really can eliminate time moving data between CPU and GPU.



Pseudo code for factoring algorithm that implementing in DGEFA

- Line 4 and 9

For the ISAMAX and SCALING operation, use the CUBLAS library on top of the CUDA to execute data on GPU instead of moving data to calculating on CPU. Through the way really eliminates time updating data between host and device and ultimately get a great performance improvement.

- Memory Coalescing

Because the hardware designed feature, there is a big program which is related with memory bandwidth when the processing unit wants to get data stored in global memory. The local memory on chip doesn't big enough to store all processing data,

therefore the accessing bottleneck is un-avoided. Coalescing on devices will cause that global memory access by all threads of a half-warp is coalesced into one or two memory transactions. By this way, the wasting time for transactions will decrease and performance will increase quickly.

- Kernel function of GPU

Memory bandwidth is always the limitation of performance, so this is an important issue how accessing memory to get maximum bandwidth. The alignment used to read several words from global memory in single one load instruction in order to use bandwidth efficiently.

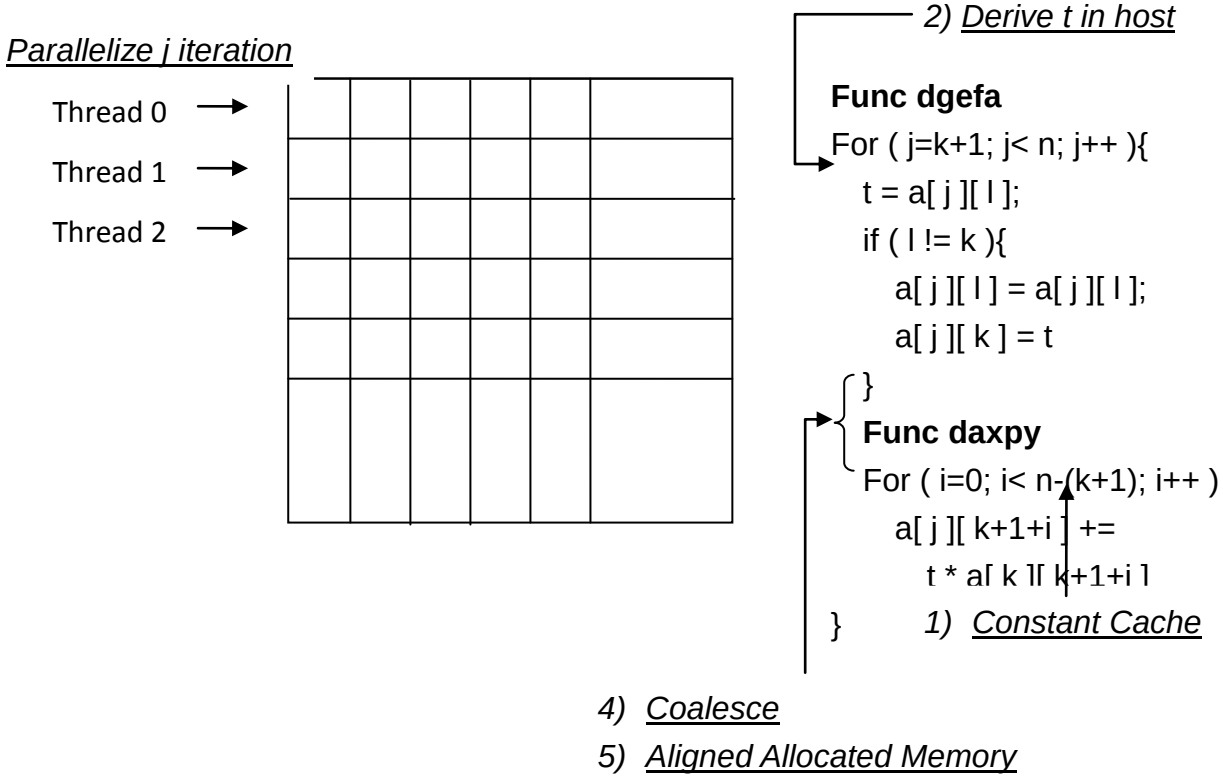
There are two different methods to archive alignment requirement:

1) Expand memory dimension to number that is nearest of power of 2 and use API "cudaMemcpy" to copy valid data row by row.

2) Use API "cudaMallocPitch" provided by CUDA v1.0 to do memory coalescing and corresponding with API "cudaMemcpy2D" to move data between host and device at once.

Method A

Parallelize j loop in **dgefa** by assigning each row to one thread on the GPU.



How to Improve Performance?

1) Constant Cache

Load a[k][k+1] vector in **daxpy** into constant memory to enhance performance because all threads read same address at same time therefore load as fast as reading register.

2) Derive t in host

Derive t for j iteration in host saved as array and perform swap when necessary.

3) Unroll loop in **daxpy**

4) Coalesce

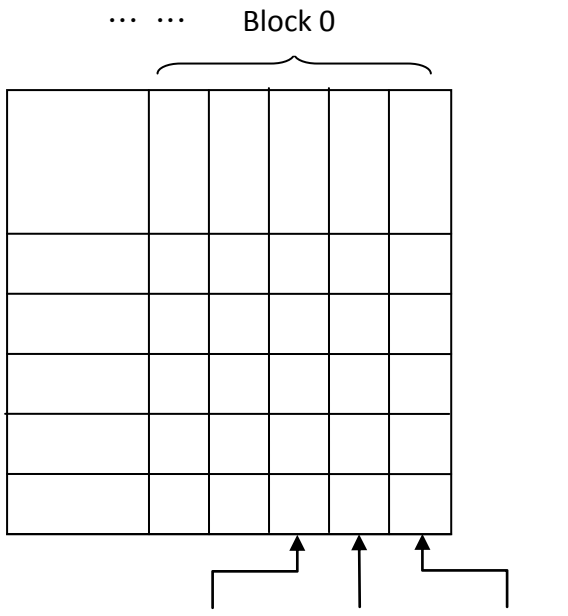
Split matrix up into block and coalescing to load data from global memory to operate.

5) Alignment of allocated memory

Allocate share memory to store result temporarily.

Method B

Also parallelize iteration in **dgefa** and difference with Method A is uses column as index to execute in Thread.



Parallelization

How To Improve Performance

1) Coalescing and Alignment

Because the start index of loop in **daxpy** varies, it can't just assign thread from start index to do coalescing. Therefore the implementation can be changed into reverse original sequence to achieve alignment requirement.

III. Performance Comparison of Rolled Single Precision LINPACK

Parallelization by Row

N=1024(matrix order)

	Performance	Execution Time of degfa	Execution Time of kernel
Base Case	772911 Kflops	906.263296 ms	348.797626 ms
Constant Memory	815032 Kflops	821.953726 ms	260.840708 ms
	p.s. Moving a[k][] to constant memory costs 1.285961 ms		
Derive t for j iterat	767950 Kflops	913.513822 ms	348.663349 ms
	p.s. Forming array and swapping spends 7.870732 ms.		

Parallelization by Column

	N=1024	N=2048	N=4096	N=8192
Padding with 2n	0.13 ms	0.26 ms	0.59 ms	1.38 ms
	5,294,105	21,052,334	73,846,120	240,940,528
cudaMallocPitch	0.11 ms	0.20 ms	0.46 ms	1.05 ms
	6,622,509	27,255,974	93,254,016	309,014,848

Parallelization by Element

	N=1024	N=2048	N=4096	N=8192
Padding with 2n	0.13 ms	0.27 ms	0.60 ms	1.41 ms
	5,389,492	20,747,686	72,721,576	237,568,128
cudaMallocPitch	0.10 ms	0.20 ms	0.47 ms	1.07 ms
	6,850,004	26,697,658	91,540,952	302,387,936

後記

鑒於學校廣布出國獎助學金之德，因此才有機運藉由實驗室交流名義得到這次的出國一探的難得機會。在七月底托福成績正式公布後，才算申請程序完全通過，正式開始收拾行囊；雖然需處理的行政手續和雜務真的是不勝枚舉，不過總算是順利成行，在 9/1 搭乘離開台灣的班機 :)

長達十二小時平穩的國際旅程後，轉搭國內線航班，飛機從 Chicago 飛到 UIUC 所在地 Champaign 只需短短的半小時，在飛機內從高處往下面看，發現腳底的景觀沒有起伏的山和蔚藍的海，取而代之的是廣闊無邊際的土地，不禁也讓人覺得心胸為之開闊。剛帶著雀躍的心情抵達時，剛離開飛機座艙不免就感受到耀眼陽光的熱力四射，乾燥的空氣也顯得格外的清新。

接下來的半年在這陌生的環境的經歷，可堪稱為記憶中最精采且畢生難忘的。

機場通往學校路途中的道路兩側，有許多長得比人高的玉米田，很適合拿來做成迷宮和捉迷藏，但是田中的玉米都採收完畢；在這兒的馬路旁幾乎看不到行人，路上也沒有洶湧的車潮，給人的感受跟台灣很不同，不會擁擠且感覺生活步調十分愜意舒適。通往學校中心的主要道路上，路途兩旁都是廣大翠綠的草原，雖然到處都是滿佈綠意，但車子開的馬路旁可以遮蔭的大樹倒是很少；這裡真不愧是典型的美國小鄉村，就如同電影拍攝的場景一般，每棟小房子都獨樹一格，各有特色、爭奇鬥艷；讓從小在鋼筋水泥建築大樓中生長的我感到十分的驚艷，景色環境氣氛十分的舒適平靜。

棲身六個月的公寓，坐落於學校有些距離的東南方，所以離開公寓前往校內的路上，會發現建築物漸漸增多且每棟建築物都各有特色。University of Illinois at Urbana Champaign 因為校區內參雜許多非學校的建築物，學術建築與商店散布在 Urbana and Champaign this twin city，因此 UIUC 才獲得『大學城』名號。學校裡的每一棟建築物也都也很有特色和壯觀，校園四周都像公園一樣綠意盎然，寒冷的空氣和灑落的溫暖陽光，讓人走在路上就覺得心情十分的雀躍；在這邊過馬路不必再提心吊膽，不僅公車司機不會橫衝直撞駕駛，和汽車駕駛都會有禮貌都會讓行人優先通行，都是讓人對這邊很驚訝的地方。

這裡的學術研究環境瀰漫著另一股認真、嚴肅地研究風氣，美國就是個從小培養小孩子獨立與自主性的文化，也因此選擇繼續深造或者就業，都是基於自我的意願，在這不乏有著從職場退下後仍然繼續學習者；普遍的狀況皆為大學畢業後，先至業界工作一段時間，當確認真的有需要繼續在學業上增進時時，才又重返學界；因此跟台灣大部分的學生只是為了大環境的壓力而升學、為了文憑而念書的心態，在根本上有些大相逕庭。

美國各個學校的學制都不太相同，而 UIUC 就是採取三學期制的學校，每學期大概短短的三個月，所以不巧的是我到達的時間剛好是橫跨兩個不完整的學期，

因此無法完整的參與某堂特定課程；橫跨 departments 豐富的課程可供選擇，可以看出他們對課程的重視程度，授課不是每位教授研究之餘的副業，而是大力推廣宣廣研究成果與招收實驗室新成員的大好時機，每堂課的教授都竭盡心力來準備授課內容與提高學生對此門課的參與程度；每位不同作風的教授傳授的不同的課程也有十分多元化的呈現：同儕之間的討論互動，小組合作的成果發表，或者是導讀論文的課堂討論，教授們在上課時都十分尊重不同的聲音，修課同學回應給教授認真的教學態度是加倍的認真，感覺教授跟學生之間不再有一道高牆與隔閡，取而代之的是真的亦師亦友的學習環境。

由胡老師帶領的 IMPACT LAB，當下的研究主力都投資於與 NVIDIA 的 CUDA GPGPU Project；剛踏入實驗室時，還對稀少的成員感到訝異，因為在台灣粗糙地來說，教授的名氣與研究生的人數幾乎是呈現完全正比，但沒想到胡文美教授這揚名海內外的名號，實驗室的研究成員居然只有十位左右，真是不禁讓我咋舌。

『在對的時間作對的事情』，當一早在辦公室打開電腦，就該認真放重心與專注力投身於研究上，在太陽下山前達到進度，減少熬夜趕進度的情況發生，實驗室那股肅穆的氣氛，真是讓人不認真都難；對自己認真負責的態度，也許就是我最該向這裡每個人學習之處 :) 胡老師真是位很值得尊敬的學者，雖然已經揚名海內外，但是還是十分謙虛有禮，那種學者的風範才真的是值得讓人肅然起敬；雖然因事務繁忙並無法花太多的時間待在實驗室中，但是對每個人的關心仍毫不馬虎，帶頭的學長還因為覺得新近來學弟妹們知識不夠因而發起每星期的讀書會，好的研究員可是需要讀過上千篇論文的；讀書會這個聽起來很時髦的名詞在這可是相當的普遍；運作模式大致為每星期指定篇論文與導讀者，參與者皆得事先研讀，討論時由導讀的人先為這篇論文做簡介與總結、報告這篇論文的讀後心得與辯護其優缺點，每人由不同的角度切入同一核心，可在讀書會中聽到許多自己沒想過的觀點與辯護自己的認知，是一種另類新奇的學習方式。

交換期中，剛好適逢 2007 micro conference 在離學校不遠的 Chicago 舉辦，因此有榮幸隨著 IMPACT 的成員遠征，這也是我第一次參與國際間 conference 經驗；廣大的會場排列整齊的座椅，台上口沫橫飛的演講者，台下踴躍的提問者…從全國各地遠道而來的菁英都藉此盛會相會與互相切磋，對於反對地論點提出質疑辯論，對於援的觀點不吝於給予讚頌、稱揚，身歷其中才深刻體會了解到閉門造車非王道，研究的精義還是要推廣與能清晰地表達創意的想法與認真的成果，公開除了可廣為人之外，接納與傾聽許多不同的意見，也是有助於研究方向的修正與正向。除了上述的優點之外，在此時學界業界集和交流之處也是拓廣人脈的好時機，在吃飯或者是 coffee break 之時隨便與一位身旁的人交談，可能都會有些意想不到的驚奇收穫。

千里迢迢的美國行，也總不能就每天都待在 Champaign 小鎮中默默無聞地度過難的的長途旅行；也因此感恩假期的 St. Louis、聖誕節的紐約之行、跨年的芝加

哥之旅與回程 stop by Los Angeles，都為這趟飄洋過海的異國之旅增添了更多精采且美好的彩色回憶，其中也不乏許多刺激冒險的探險經歷，許多事件想來令人莞爾一笑。在九月到達至三月離開的這期間內，剛好趕上夏季的結束、經歷落葉繽紛的秋天與蕭瑟的冬天，這種四季分明的季節景色在這可堪稱一覽無遺。夏季時因緯度較高讓太陽在晚上快八點才下山，入夜後溫度下降十分的快速；早晨時陽光灑落在茂密的綠葉間，道路兩旁一望無際的平坦草地也是碧草如茵、生機洋溢，轉頭張望鎖入眼簾的風景，都如詩如畫般的美麗讓人心曠神怡；在某天起床時氣候突然轉冷，一天中氣溫最高時沒有超過二十度，低溫甚至只剩三度，走在路上沒有衣服遮蓋的部分都有快被凍僵的感覺;更雪上加霜的是有時候會颳風下雨，原本道路旁綠意盎然的樹，都漸漸轉成了紅色，路面上落葉紛紛讓風吹過捲起小漩渦狀，真的是超有秋天蕭條的感覺啦。周遭環境隨著季節轉換，也替每天出門增添了不少樂趣；秋意瀰漫的季節在最後一片樹葉落至地面時結束，短短地不超過三個星期，宣告了冬天的到來。在感恩節後通常都會開始下起雪來，今年也很準時感恩節當天下起了大雪來，為了過冬而光禿禿的樹搭配陰沉沉的天空讓環境感覺十分的蕭條，轟隆隆的雷聲還讓人期待可以看到落電就擊中在前面廣闊平原的景象，望著廣大的平原其中坐落著稀疏的建築物，讓人覺得心情十分的舒暢。

第一次體驗看到街上到處都是外國人、身處異地的真實感受，美國真不愧是個文化大融爐，走在街上看到的可不只美國人，在這裡外國人可堪稱一點都不稀奇，在加上在 UIUC 就讀的外籍學生比例十分的高，因此黑頭髮黃皮膚的亞洲人在這邊倒還不算少見。談到文化差異，就不得不說雖然身處於網際網路，資訊發達與訊息傳播快速的二十一世紀，但是各地的文化分歧還是造成適應不同環境的重大阻礙；美國的價值觀與含蓄的傳統中國思想落差甚巨，廣納百川的胸懷、直言不諱的態度、相處的相敬如賓和對於運動的熱愛…只能說說是各地文化的差異所造成的價值觀經年累月而成，也都是各地風俗特異和獨到之處。

在此表達衷心感激學校與教育部提供此機會讓我能親眼見證遠在另一端截然不同的世界，身體力行體驗完全不同的生活，『行萬里路，勝讀萬卷書』真的是此趟旅程的最好注解。親身的去感受體驗，才會有更為深刻的記憶與思維，藉由這趟旅程我覺得從中學習而得到最多的就是培養獨立的精神與心靈的成長，擺脫過去稚氣的思考與依賴，轉換為更為寬廣縝密的心思與寬闊的胸襟。在臨別之際胡老師跟我深談的一席話我將會收藏在心中：**只要這趟旅程能讓你有即使是現在無發察覺的些微改變，但此的人生小轉折隨著時間的流逝，也會有放大作用，深刻影響你往後的人生，這也是教育的意義，希望妳能越來越好**。真的有無法言喻的感謝，能在我人生的經歷中有著這我以前完全沒有想過的際遇，也深深感到極大的幸運;希望以後能有更多人透過這個獎學金來對平凡無奇的人生有另一個重大的衝擊與改變 :)