Module 1 Part 1:

Introduction to How to Write Fast Code?

Carnegie Mellon University
18-645

Jike Chong
Ian Lane

How to Write Fast Code?

Fast Platforms

- **Multicore platforms**
- **Manycore platforms**
- **Cloud platforms**

+

Good Techniques

- **Data structures**
- **Algorithms**
- **Software Architecture**

- **Course Goals**
  - To write fast code for your research/application, you should:
    1. **F**eel comfortable hacking up a solution
    2. **L**everage existing software building blocks
    3. **I**ndicate which platform is the best one to use
    4. **R**eason about why a piece of existing code is slow
    5. **T**ake care of potential performance bottlenecks

Instructors

- **Jike Chong**
  - Email: jike.chong@sv.cmu.edu
  - SkypeID: chong.18645
  - Office Hours: Tuesdays (6:00 – 8:00pm CA) and by appointment

- **Ian Lane**
  - Email: lane@cs.cmu.edu
  - SkypeID: lane.18645
  - Office Hours: Tuesdays and Thursday (08:00 - 11:00am CA)

Instructor: Jike Chong

- 10+ years working on ***Fast Platforms***, 6 years researching ***Good Techniques***
- **Adjunct Professor**, ECE, Carnegie Mellon Silicon Valley
  - Directs the CUDA Teaching Center and the CUDA Research Center at CMUSV
- **Head of Data Sciences**, SimplyHired.com
  - Using machine-learning algorithms in data analytics
- Prior work:
  - Three first-authored US Patents, one pending
  - Intel Research Labs, Sun Microsystems, Xilinx Research Labs, Electric Cloud, etc
  - Ph.D. from UC Berkeley, M.S. and B.S. for Carnegie Mellon University

Instructor: Ian Lane

- **Assistant Research Professor** (CMU-SV, LTI, ECE)
  - 10 years working on ***Speech Recognition, NLP, MT*** and ***Machine Learning***
  - **Co-Directs the CUDA Teaching and Research Center at CMU-SV**
  - ***Heavy user of distributed computing (Maui, Condor, SLURM, HADOOP):***
    - Large data, computationally expensive algorithms and limited computational resources
    - State-of-the-art Automatic Speech Recognition systems contain many components and can require over 40,000 CPU-hours to train (4.5 years!)
    - For effective throughput must also consider Network, Disk and Memory Access

High-throughput Computing:

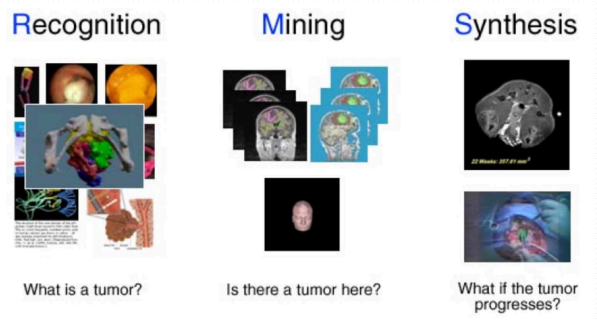
Create new technologies → Deliver practical systems

Outline

- Why Fast Code Now?
- What does 100x speedup mean?
- Course structure and organization

Why Fast Code Now?

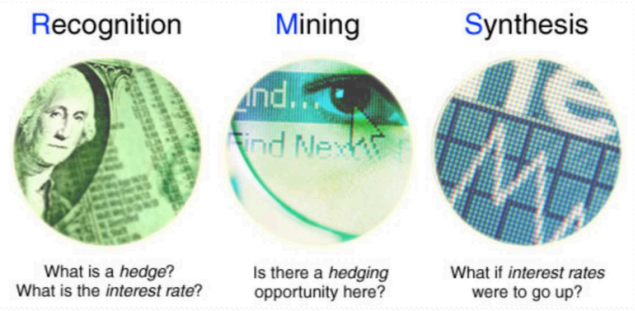
Need is driven by the applications…
…NOT by the availability of the platforms.



Dubey, "Recognition, mining and synthesis moves computers to the era of tera", Technology@Intel Magazine, February 2005

RMS Appears Everywhere

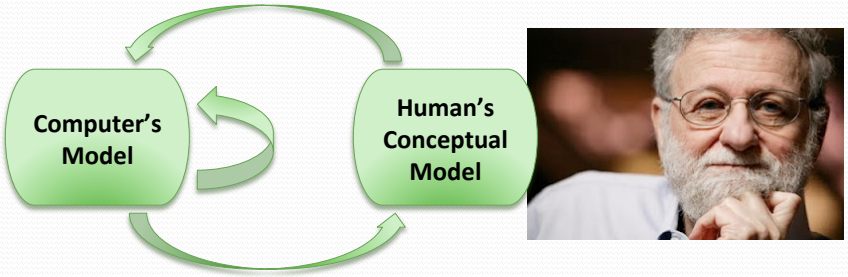
- **Recognition**: Interpretations of the world (with models)
- **Mining**: Understanding of the world (discover hidden patterns with model)
- **Synthesis**: Anticipate outcomes in the world (using the models to predict)



Chen et al, "Convergence of Recognition, Mining, and Synthesis Workloads and Its Implications", Proceedings of the IEEE, Vol. 96, No. 5, pp. 790-807, May 2008

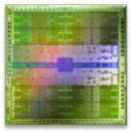
Norman's Gulf

- Donald Arthur Norman: a scholar in cognitive science, design and usability
  - Author of the book *The Design of Everyday Things*
- More computation is being dedicated to bridging Norman's Gulf
  - Making computers easier to use
- What are some examples you see?



D. A. Norman, The Design of Everyday Things. New York: Currency-Doubleday, 1989.

What is being valued today?



Hardware ⇨ **Commoditized**

```
int main(void) {
    mykernel<<<1,1>>>();
    printf("Hello World!\n");
    return 0;
}
```

Software ⇨ **Open Sourced**

Data ⇨ **King!**

Outline

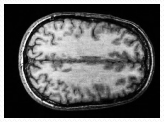
- Why Fast Code Now?
- What does 100x speedup mean?
- Course structure and organization

What does 100x Speedup Mean?

12 hours:	10x speedup →	1.2 hours
	100x speedup →	**7.2 minutes**
	1000x speedup →	43.2 seconds

- Game changing technology advances
  - Overnight jobs becomes interactive


Speech Analytics


Medical Imaging

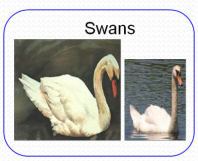
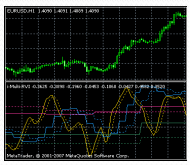
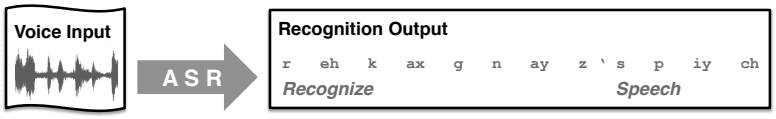
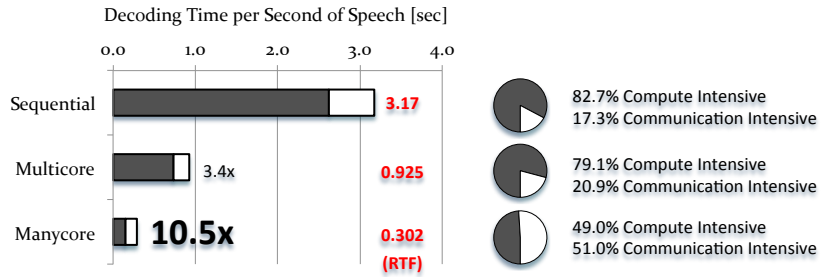


Image Recognition


Computational Finance

Automatic Speech Recognition



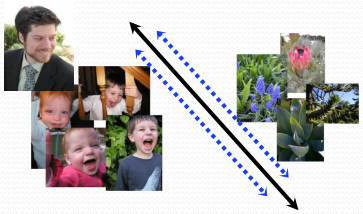


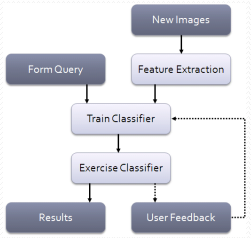
Kisun You, Jike Chong, *et al*, "Parallel Scalability in Speech Recognition: Inference engine in large vocabulary continuous speech recognition", IEEE Signal Processing Magazine, vol. 26, no. 6, pp. 124-135, November 2009.

Fast, Robust Pediatric MRI

- **100X** faster reconstruction
- Higher-quality, faster MRI
- This image: 8 month-old patient with cancerous mass in liver
  - 256 x 84 x 154 x 8 data size
  - Serial Recon: 1 hour
  - Parallel Recon: less than 1 minute
- Fast enough for clinical use
  - Software currently deployed at Lucile Packard Children's Hospital for clinical study of the reconstruction technique

Support-Vector Machines





- Algorithmic changes and parallel implementation lead to performance speedup: Core 2 Duo versus G80

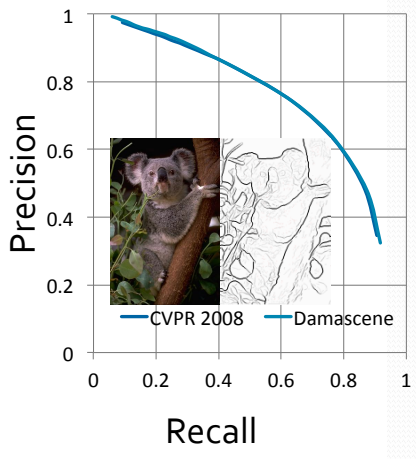
Computation	LIBSVM	Our algorithm2-core Parallel CPU	Our algorithm, 16-core Parallel GPU
SVM Training (geo-mean)	771.8 s	---	38.5 s
SVM Classification	41.42 s	4.21 s	0.38 s

100X speed-up

896 downloads since release in 10/2008

Fast support vector machine training and classification, Catanzaro, Sundaram, Keutzer, International Conference on Machine Learning 2008

Image Contours Detection

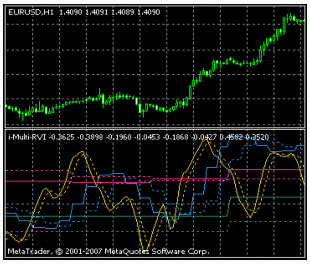


- We achieve equivalent accuracy on the Berkeley Segmentation Dataset
  - Comparing to human segmented "ground truth"
- F-measure 0.70 for both
- Human agreement = 0.79
- 3.8 minutes to 1.8 seconds: 126x speedup
- 616 downloads since release in October 2009

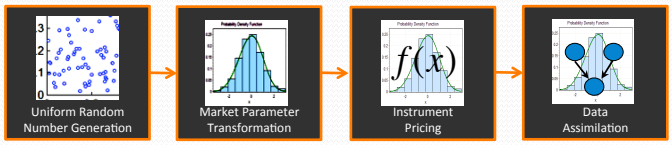
"Efficient, High-Quality Image Contour Detection" Catanzaro, Su, Sundaram, Lee, Murphy, Keutzer International Conference on Computer Vision, 2009

Computational Finance

- **Value-at-Risk Computation** with **Monte Carlo Method**
- Summarizes a portfolio's vulnerabilities to market movements
- Important to algorithmic trading, derivative usage
- Improved implementation to run **60x faster** on a parallel microprocessor



Four Steps of Monte Carlo Method in Finance



Matthew Dixon, Jike Chong, Kurt Keutzer, "Acceleration of Market Value-at-Risk Estimation", Workshop on High Performance Computing in Finance at Super Computing 2009, November 15, 2009.

NY Times TIFF to PDF

- In 2007, the New York Times decided to make all the public domain articles from 1851-1922 available free of charge
  - Needed to convert from TIFF to PDF

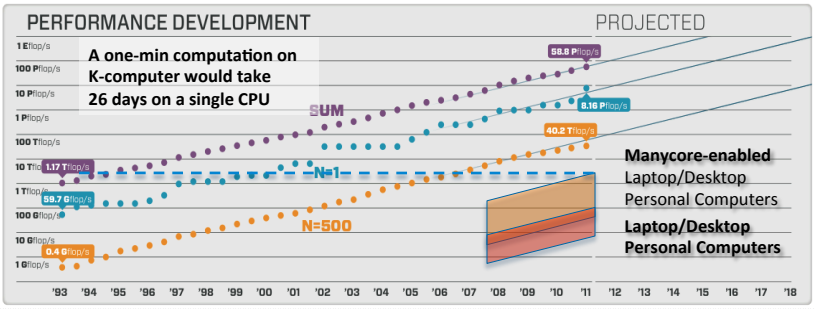
Data set:	4TB of raw image TIFF to 11 million PDF
Compute Instances:	100 Amazon EC2 instances
Time taken:	24 hours
Cost:	$240

The New York Times

http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/

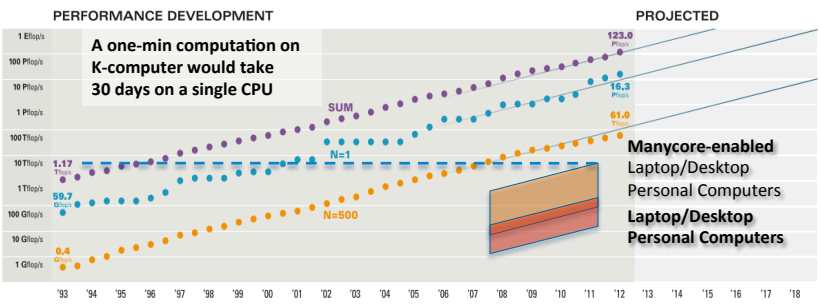
HPC Top500 Ranking

	NAME/MANUFACTURER/COMPUTER	LOCATION	COUNTRY	CORES	R_{max} Pflop/s
1	**K Computer** SPARC64 VIIIfx 2.0GHz, Tofu interconnect	RIKEN	Japan	548,352	8.16
2	**Tianhe-1A** 6-core Intel X5670 2.93 GHz + Nvidia M2050 GPU w/custom interconnect	NUDT/NSCC/Tianjin	China	186,368	2.56
3	**Jaguar** Cray XT-5 6-core AMD 2.6 GHz w/custom interconnect	DOE/SC/ORNL	USA	224,162	1.76
4	**Nebulae** Dawning TC3600 Blade Intel X5650 2.67 GHz, NVidia Tesla C2050 GPU w/ Iband	NSCS	China	120,640	1.27
5	**Tsubame 2.0** HP Proliant SL390s G7 nodes (Xeon X5670 2.93GHz) , NVIDIA Tesla M2050 GPU w/Iband	TiTech	Japan	73,278	1.19



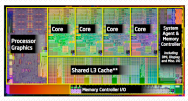
HPC Top500 Ranking

	NAME	SPECS	SITE	COUNTRY	CORES	R_{max} Pflop/s
1	**Sequoia**	IBM BlueGene/Q, Power BQC 16C 1.60 GHz, Custom interconnect	DOE / NNSA / LLNL	USA	1,572,864	16.33
2	**K computer**	Fujitsu SPARC64 VIIIfx 2.0GHz, Tofu interconnect	RIKEN AICS	Japan	705,024	10.51
3	**Mira**	IBM BlueGene/Q, Power BQC 16C 1.60 GHz, Custom interconnect	DOE / SC / ANL	USA	786,432	8.153
4	**SuperMUC**	IBM iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband QDR	Leibniz Rechenzentrum	Germany	147,456	2.897
5	**Tianhe-1A**	NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050	NUDT/NSCC/Tianjin	China	186,368	2.566

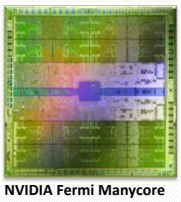


Philosophy on Platforms

Provide theoretical background and hands-on practices…
…to innovate with multicore/manycore/cloud based platforms.



Intel Sandy Bridge Multicore Processor (Core i7-2600K)



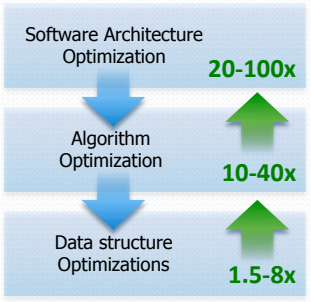
NVIDIA Fermi Manycore Processor - GTX580

Yahoo! Hadoop Cluster ~2000 nodes in one cluster

- ***Significant scientific advances*** will be empowered by multicore/manycore/cloud based platforms over the next decade
- Knowledge of these new computing capabilities will command a ***dominating advantage*** over your peers…
  …in developing innovative techniques to solve challenging problems

Philosophy on Techniques

- Efficient ***software architecture*** is the most important to designing fast code
  - Software design patterns
- Understanding the implementation platform will help reason about application performance bottlenecks
- ***Hands-on experience*** provide confidence for you to effectively use these technologies for your research and development needs



Outline

- Why Fast Code Now?
- What does 100x speedup mean?
- Course structure and organization

Secret Word to Earn You a Spot!

- Secret word: **FLIRT**
- Send us an email with:
  - Subject: "**L01: Code FLIRT – Lastname, Firstname**"
  - To: "**18645@sv.cmu.edu**"
- Please use the same name as shown on your student record
- ***Send it now!***
- We will take the time stamp of this email into account when managing the waitlist

Organization of This Course

- Bi-coastal broadcast
  - Every Tuesdays and Thursdays
  - Silicon Valley (Blg 23, Rm 118) 12:00 noon-1:20 pm PDT
  - Pittsburgh (HH1108) 3:00 pm-4:20 pm EDT
- Class website:
  - **fast645.info**
- During class time:
  - Lectures, Team presentations
  - Q&A sessions, Midterm/Final Exams
- Outside class time:
  - Three homework assignments
  - Three mini homework projects
  - One term project (focused on your own research or interests)

Grading and Expectations

- GRADING
  - 10% Homework assignments (Assignments to help with the projects)
  - 10% Midterm examination
  - 30% Homework projects (Three equally weighted mini projects)
  - 20% Term Project
  - 30% Final examination
- EXPECTATIONS
  - Attend majority of lectures
  - Hand in all assignments and mini-projects
  - Complete a term project
  - Attend the final exam

How to Earn Extra Credit!

- **Purpose**:
  - Help peers, do good work, get rewarded!
  - Reward can push up grade by ½ step: e.g. A to A+
- **Three Reward Opportunities**:
  - Use the Forum to get help and provide help from each other for setup the infrastructure for projects
    - Most useful answer to a discussion forum question gets **1** credit points
  - Mini homework projects will be tested **daily** for performance with **MOTION CHART**!
    - Top two performing team will get **2** credit points for each person for each day they are on top
  - Outstanding Final projects
    - Top 2-4 projects will be awarded **20** credit points for presenting at the HPC and GPU Supercomputing Group of Silicon Valley

Software Reuse and Plagiarism

- **Question**: Can I learn to use code from the web?
- **Answer**: Yes! But only under certain circumstances…

Plagiarism	Acceptable SW Reuse
Collusion Unacknowledged code from a third-party	File-level Reuse Third-party code factored out to a separate file
Reverse Engineering Unacknowledged re-use of some code abstraction	Acknowledgement in Documentation Third-party code clearly distinguished in documentation
Translation Unacknowledged re-use by translating functions from one language to another	Adequate Testing Third-party code must be tested to one's own requirements
Code Generation Unacknowledged assistance by using some code generators	
Reuse Without Testing Reuse third-party code without testing against one's own requirements	

Gibson, "Software Reuse and Plagiarism: A Code of Practice", In Proceedings of ITiCSE'2009. pp.55-59

Structure of Lectures

- **Module 1: Background**
  - Hardware, applications
- **Module 2: Multicore Programming**
  - Focus on application design with OpenMP
- **Module 3: Manycore Programming**
  - Focus on application design with CUDA
- **Module 4: Cluster Programming**
  - Focus on application design with Hadoop
- **Module 5: Special Topics**

In Each Lecture

- We will provide a set of questions
  - You should be able to answer them after the lecture
- Today's question:
  - What does it mean to **FLIRT** with parallel computing technologies and working with the principles and practices of Writing Fast Code?
- You will be asked to answer those questions at the end of the class

Structure of Team Presentations

- Two teams with the best performance will present their code
- Two teams with the slowest performance will have their code presented
- Multi-locational teams gets 5% extra points for project
- Best presentations will be invited to present at the HPC and GPU Supercomputing Group of Silicon Valley
  - Members representing engineers and managers at major Silicon Valley companies such as: Google, Apple, Intel, NVIDIA, AMD
  - A great opportunity to make connections for cool internships

Structure of Q&A Sessions and Exams

- In-class Q&A Sessions
- Midterm:
  - Covers background, multicore/manycore application development
- Final:
  - Covers all material in the course

Structure of Projects

- Three mini projects throughout the semester
  1. Multicore project with OpenMP
  2. Manycore project with CUDA
  3. Cloud project with Hadoop
  - Projects are done in teams of two or three students
    - Where possible try and form teams across geographic locations
  - Projects use ***Git*** code repositories
    - Project files must be checked out, optimized, and checked in
- Term project
  - An application area of your choice
  - One of the suggested projects
  - Term project introduction on

Structure of Projects

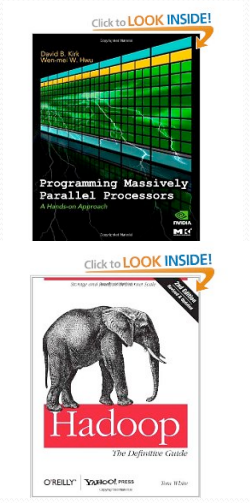
- Programming assignments starting with fully working kernels
- Accelerate them with techniques discussed in class
- Automated scripts will pick up each teams project everyday
  - cron job starts at: 8pm PDT, 11pm EDT
  - Top three teams each day will accumulate bonus points
    - Can be used to push up a grade-level in borderline cases
- Write-up and presentations during class

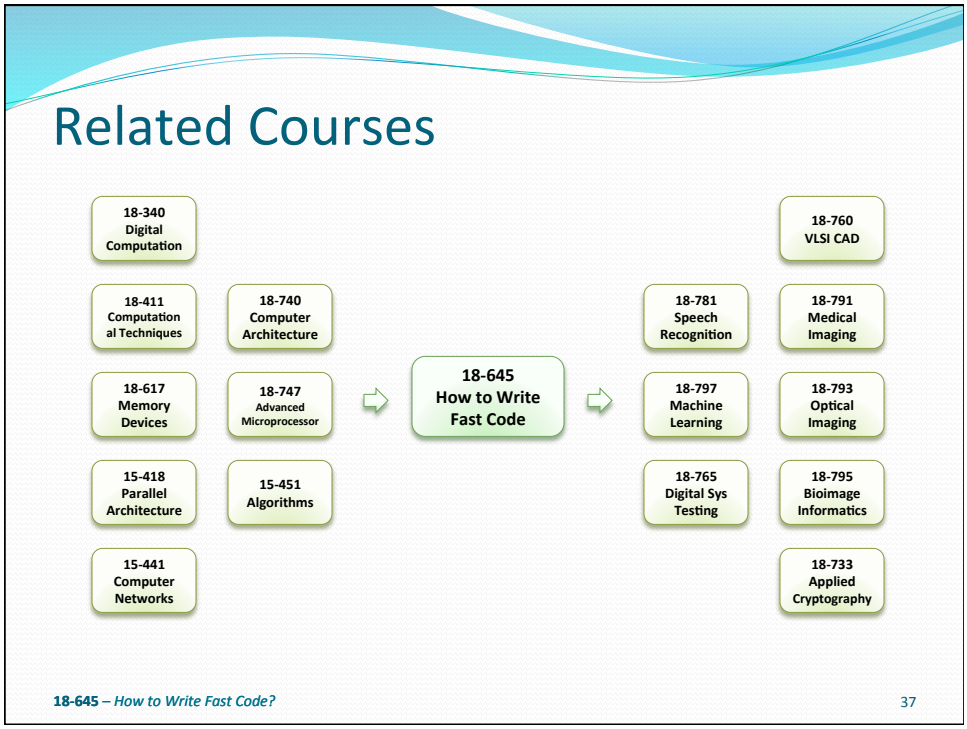
Structure of Homework

- Purpose:
  - Internalize concepts discussed in class
  - Opportunity to setting up the project environment
- Two week's time to complete the assignment
- Discussion among peers is allowed and encouraged
- However, tasks assigned and write-ups must be completed individually

Recommended Books

- Required:
  - Kirk, Hwu, Programming Massively Parallel Processors, Second Edition: A Hands-on Approach, Morgan Kaufmann, 2012
- Optional:
  - White, Hadoop: The Definitive Guide, 3rd Edition, Yahoo Press, 2012




Related Courses
18-340 Digital Computation
18-411 Computational Techniques
18-617 Memory Devices
15-418 Parallel Architecture
15-441 Computer Networks
18-740 Computer Architecture
18-747 Advanced Microprocessor
15-451 Algorithms
18-645 How to Write Fast Code
18-781 Speech Recognition
18-797 Machine Learning
18-765 Digital Sys Testing
18-760 VLSI CAD
18-791 Medical Imaging
18-793 Optical Imaging
18-795 Bioimage Informatics
18-733 Applied Cryptography
18-645 – How to Write Fast Code?
37