

# 625 Final Project

- Team based
  - Semester-long (due at final)
  - Two presentations & writeups
    - See <http://cse.unl.edu/~goddard/WritingResources/Templates/Generic-Technical-Paper-Skeleton.html> for outline
    - Peer evaluation after each.
  - Project choice due ~10/12
- Topics
    - See [http://www.cs.berkeley.edu/~dumel/cs267\\_Spr09/posters.html](http://www.cs.berkeley.edu/~dumel/cs267_Spr09/posters.html) for samples
  - Real projects in following slides
  - Something of interest to you
    - Parallelize routines in Android
    - Scientific computing
    - Gaming engine improvement

# Stock trading/position analysis

- PI: Holden Mai, Advise Technology
- Area: Parallelization
- CUDA: No
- Languages/Env: C# preferred
- Apply filtering rules to stock holding data
- Potentially 5M records/day, looking back over last 5 years (TB-class dataset sizes)

# Improving NetPIPE CUDA support

- Prof: Dr. Dave Turner, CS
- Area: performance measurement/tuning
- CUDA: Yes
- Make a CUDA module that measures the communication rate from CPU->GPU, GPU->CPU, memcpy within a GPU, and from one GPU to another.
- There are also various memory banks within a GPU that can be explored as well as different memory access patterns.
- Could also compare CUDA to OpenACC.

# Improving Experimental Data I/O & Processing

- Prof: Dr. Mary Cain, Psych
- Area: parallelization/workflow
- CUDA: Maybe
- Language/Env: R/Matlab?  
Python?
- Take annotated video data & speed up validation & do statistical analysis from rats doing mazes & such.

# Satellite Image Analysis

- Prof: Dr. Shawn Hutchinson
- Area: parallelization/workflow
- CUDA: Yes
- Language/Env: R
- Speed up time-series analysis of satellite images.
- Images are ~16M pixels, 4 images per year, analyzed for last 15 years
- Current 10-way parallelization has been running since May.

# Kepler Constant Calculation

- Prof: Narayan Khadka, Physics
- Area: parallelization/workflow
- CUDA: Maybe
- Language/Env: Python
- Speed up calculation of the Kepler Constant

# NOAA remote data processing

- Prof: Vahid Rahmani, BAE
- Area: parallelization/workflow
- CUDA: Maybe
- Language/Env: open
- Speed up workflow downloading and processing some observed and remote sending precipitation data from the National Oceanic and Atmospheric Administration agency

# A COMPUTATIONAL APPROACH TO THE ENUMERATION OF $G$ -PARKING FUNCTIONS

- Prof: Tyler Aden, Math
- Area: parallelization
- CUDA: Maybe
- Language/Env: Python
- Speed up an algorithm to efficiently list all maximal  $G$ -parking functions.
- Applied spanning-tree/graph algorithms



# Calculating Particle Decay

- Prof: Graham Wilson, Physics
- Area: parallelization
- CUDA: Maybe
- Language/Env: Convert Matlab to something faster – C?  
Python?
- Speed up an algorithm to calculate particle decay.
- 50K Monte Carlo simulations

# Parallelize ExaML with snakemake

- Prof: Brad Olson, Bio
- Area: parallelization
- CUDA: Maybe
- Language/Env: Snakemake scripting language
- Parallelize ExaML with snakemake to perform high throughput phylogeny on a folder full of sequence alignments utilizing checkpointing when a killable job it killed.
- “This one is probably easy...”