Overview

- Project Overview
- Similar Work
- My Project
  - Challenges
  - Choices
- Target Experiments
  - Expected Results
- Project Successes
- Further Research
The Project

- Retrieve 3 million images from Wikipedia
- Store them in a directory
- Gather data about the images
  - Wikipedia stats
  - ImageMagick
- Store that data into a MySQL database
- Resize the images
- Pack them into a frame to make a beautiful collage
My Project

- Retrieve 3 million images from Wikipedia
- Store them in a directory
- Gather data about the images in parallel
  - Wikipedia stats
  - ImageMagick
- Store that data into a MySQL database in parallel
- Resize the images
- Pack them into a frame to make a beautiful collage
Why it’s important

- This step produces the information that we’ll need to pack the images in an artistic way
- Identifying data about 3 million images takes a lot of time
  - Parallelization will significantly reduce that time
- We can easily keep track of and add more information about the images in the data table
Similar Work

- Holloway, Bozicevic, and Börner
- Visualization of English Wikipedia statistics, semantic structure, and article age
Similar Work

- Herr et al.
- A visualization of the structure and dynamics of the English Wikipedia
Similar Work

- Herr, Holloway, Börner (2007)

http://gigapan.com/gigapans/4304
Similar Work (at Smith)

- Elizabeth Do (2011)
- An image collage of the 80 most frequent words in Wikipedia
Overview

- Project Overview
- Similar Work
- My Project
  - Challenges
  - Choices
- Target Experiments
  - Expected Results
- Project Successes
- Further Research
What I am doing

- Multi-threaded Java program
  - Master-worker paradigm

- Master
  - Walks enwiki directory on Hadoop0 to get images
  - Sends images in blocks to workers

- Worker
  - Process the images with ImageMagick
  - Insert image data into a MySQL table on Hadoop0
What I am doing

- Lots of Java!
  - Multi-threading in Java
  - Directory walking in Java
  - Running ImageMagick commands with Java
  - MySQL in Java (JDBC)
    - Sending queries
    - Data insertion
Challenges

- Coding challenges
- Design challenges
- Other challenges
Coding Challenges

- Installing the JDBC Driver
- Learning how to send MySQL queries in Java
- Learning how to use Java Processes to run ImageMagick commands
- Implementing a double-buffer
  - Workers telling the master that they’re ready for more images
- Optimizing parallel code
Design Challenges

- Correctly implementing the Master/Worker paradigm
  - What is the best approach to quickly sending a block of images to the (correct) worker (like MPI)?
  - Two programs? One program with several classes?
  - Round-robin? Only send images to idle workers? Double-buffering?
Other Challenges

- Tuning parameters
  - Will discuss more when I discuss target experiments

- Testing
  - Analyzing and understanding the data
  - Making sure my results are accurate and make sense

- Writing the research article
Choices

- MySQL
- Java
- Using a double-buffer
- Production on Hadoop
Overview

- Project Overview
- Similar Work
- My Project
  - Challenges
  - Choices
- Target Experiments
  - Expected Results
- Project Successes
- Further Research
Target Experiments

- What is...
  - the optimal number of threads?
  - the optimal image block size per thread?
  - the optimal number of rows to insert per MySQL query?
Expected Results

- The optimal number of threads will roughly equal the number of cores
- Optimal block size will be in the range of hundreds, not thousands
  - Conjecture based on Homework 5 results
- Optimal number of rows to insert... ???
  - The maximum packet size is 1GB
  - Might be based on the speed of the connection
Project Successes

- Master successfully walks directory
- Workers successfully access MySQL database
- Currently working on:
  - Developing the double buffer
  - Using Java Processes to run ImageMagick commands
Future Research

- Using other tools and resources to mine more data from the images (i.e. dominant color, tags, captions, groups of images from similar pages, etc.)
- Run program on (and optimize parameters for) AWS
- Develop the next stage of MySQL interaction: resizing the images based on the image data in my table, and store those images in another table
Thanks for a great semester, 352!