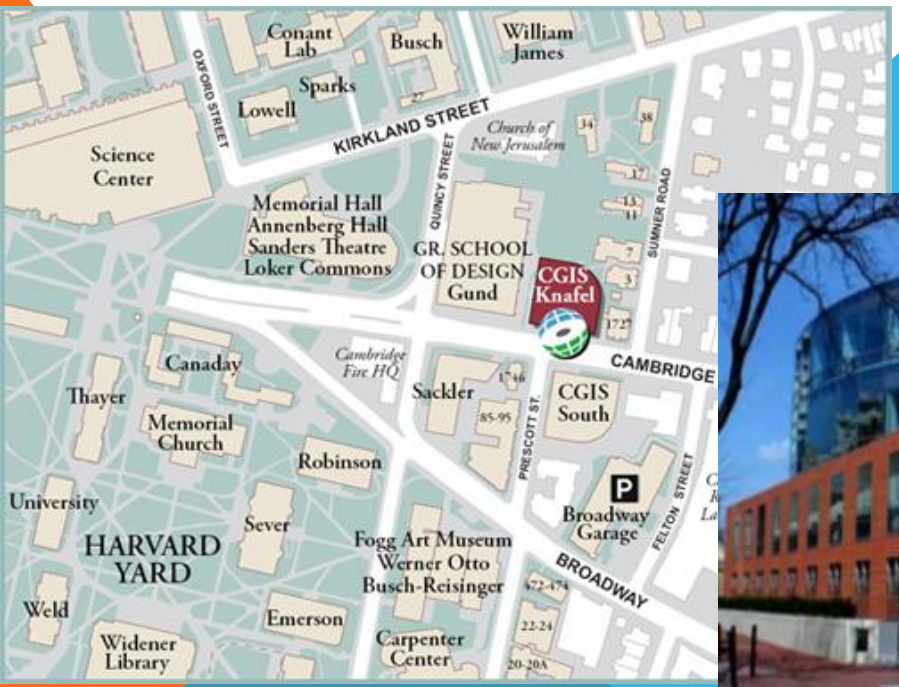
Utilizing GPU for CyberGIS

TODD MOSTAK & LEX BERMAN

CENTER FOR GEOGRAPHIC ANALYSIS
HARVARD UNIVERSITY



Temporal Gazetteers http://fas.harvard.edu/~chgis/gazetteer

Conferences and Workshops

Special Track on Historical Gazetteers: Temporal Elements
Symposium on Space-Time Integration in Geography and GIScience

2011 **AAG Seattle** **Presentations** Session Program

Working Digitally with Historical Maps
Special sessions hosted at the *New York Public Library*

2012 **AAG New York** **Presentations** Session Program

Temporal Gazetteer Web Service

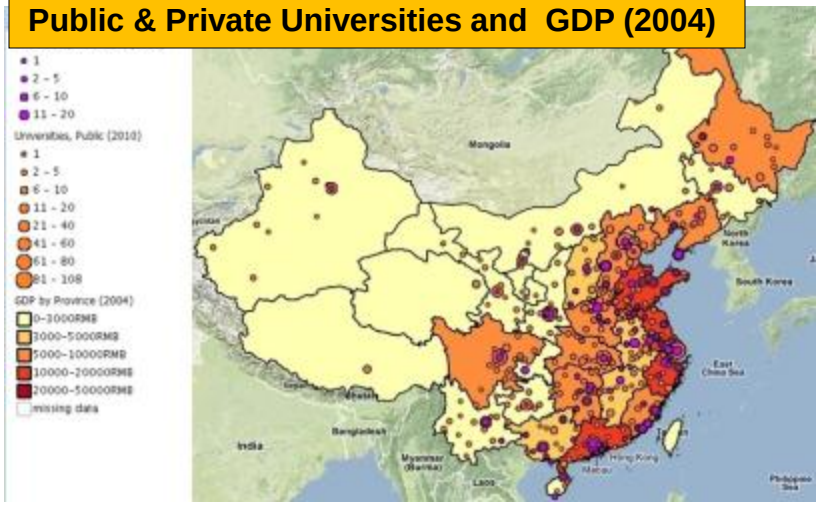
Reference Bibliography

Selected References

date	*authors*
2011-11	**Southall, Mostern, Berman**
2011-03-10	**Keith Murray**
2011-02-24	**Barker, Isaksen**
2011-12-03	**Krzysztof Janowicz**
2010-06-24	**Aeur, Fees, Zipf**
2010-06-24	**Popescu, Grefenstette**
2010-06-03	**Sean Gillies**
2009-11-06	**Kessler, Janowicz, Bishr**
2009-10-29	**Auer, Lehmann, Hellmann**
2009-09-07	**INSPIRE Working Group**
2009-03-01	**Manguinhas, Martins, Borbinha, Siabato**
2008-04-04	**Aucott**, et all
2008	**Goodchild & Hill**
2006-12-09	**Janee**
2006-12-09	**NGA Workshop**
2004-02-26	**Hill, Janee**

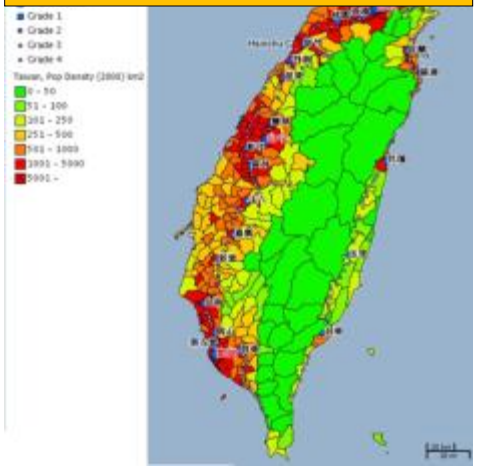
WorldMap Example http://worldmap.harvard.edu/china

Public & Private Universities and GDP (2004)



High Speed Rail (2010) & Historic Routes

Taiwan Rail and Pop (2000)



Urban Data, Beijing (1916)

GEOPS: GEOSPATIAL OPERATIONS SYSTEM

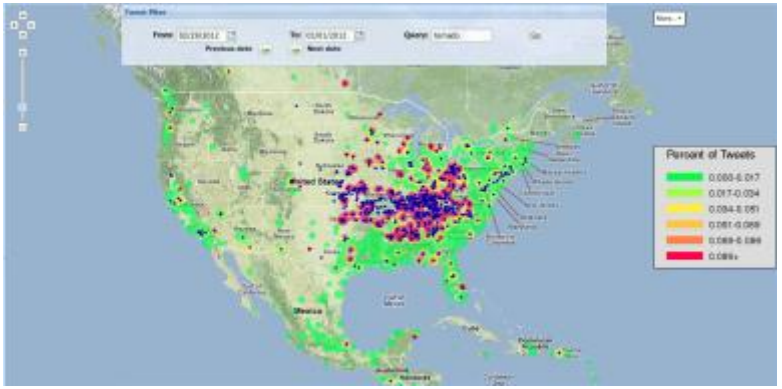
GPU ACCELERATION OF GEOSPATIAL QUERIES

Todd Mostak

WHY THE GPU?

- Inadequacy of conventional desktop GIS software for handling big datasets
- Today's GPUs possess massively parallel processing capability
  - Between 500 to 3000 cores for high-end GPUS
  - High memory bandwith: 200-400 GB/s
    - Versus 20-40 GB/s for a high-end CPU
  - Data is easily rendered graphically if it is already on the GPU
- Challenges
  - Difficult to keep all cores occupied
    - Particularly under divergent execution paths
  - Certain problems are difficult to parallelize
  - Speed between GPU and main memory is limited by PCI-E speeds
    - 4.0 GigaTransfers / sec for PCI-E 2.0
    - 8.0 GigaTransfers / sec for PCI-E 3.0

Screenshot of GEOPS + Worldmap: Relative intensity of "flu" on Twitter from January 26, 2012 to March 25, 2012



Relative intensity of "tornado" on Twitter (with point overlay) from Febuary 29, 2012 to March 1, 2012

FEATURES OF GEOPS

- **Full column-store in-memory persisted database written in C++ / CUDA with Postgres and "Grim Tweeper" connectors**
  - Data can reside on GPU or be streamed asynchronously from CPU memory
- **Text stored and searched as product of prime numbers**
  - Compresses text while allowing for fast lookups without the overhead of constructing traditional inverted indices
- **Tests for spatial intersection are performed by rasterizing vector layers (such as map polygons)**
  - Each "pixel" in the raster stores the id of the associated polygon
  - To find the bounding polygon for a feature, GEOPS converts its map coordinates to raster coordinates and retrieves the pixel (id) value at that position
- **On the fly generation of pointmaps, heatmaps and choropleths**
  - Amazon GPU instance with 2 GPUS can render Gaussian kernel heatmap of 120 million tweets in $1/20^{th}$ second
- **Scales linearly with additional GPUs**
- **Full stand-alone WMS-T webserver that also emits json to construct graphs of any attribute over time**

WORDS AS PRIMES

- Text is stored on the GPU as the product of a series of primes – one per each word
- The most frequently appearing words in a corpus are assigned the lowest primes
- To test if a word, or a series of words, are contained in a text field, we can simply check if the prime associated with the word divides into the field's prime product evenly
  - In practice, a bitshift trick is used to avoid a relatively costly division operation
- Also compresses text
  - ~ 5X for Twitter data
- Since no indexes are needed for text lookups, rows of data can be streamed onto the GPU and are available for immediate visualization and processing

Word	Freq. Rank	Prime
I	1	3
I'm	2	5
at	3	7
the	4	11
a	5	13
to	6	17
you	7	19
my	8	23
and	9	29
me	10	31

Text: To you, I'm me.

17 X 19 X 5 X 31 =50,065

Is "at" (7) in the text?

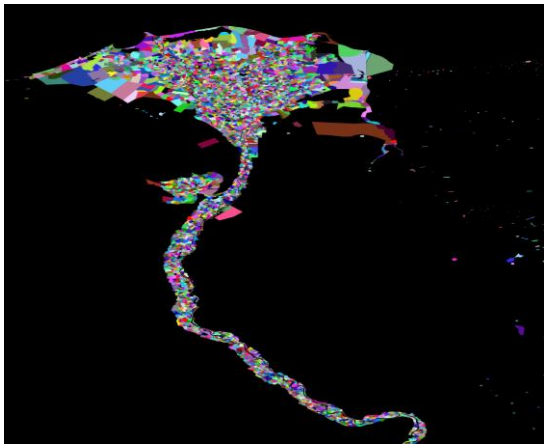
No, since 50065 % 7 ≠ 0

Is "you" (19) in the text?

Yes, since 50065 % 19 = 0

SPATIAL INTERSECTIONS

- GEOPS rasterizes polygons inputted in vector (shapefile) format and then lookups features against the resulting raster
- Much faster than conventional R-Tree techniques used by systems such as PostGIS
- If feature falls on pixel marked as border between polygons, geometrical techniques can be used to disambiguate
- Next step: use linearized quad-trees as rasters

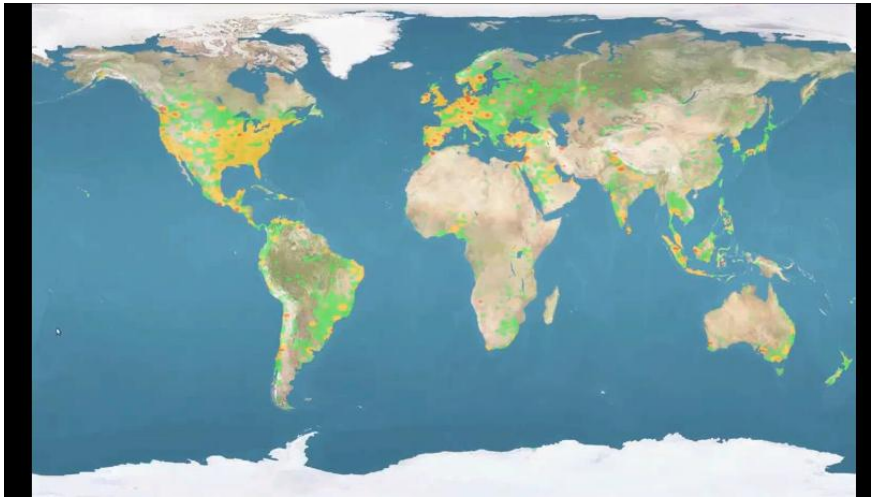


Egyptian Administrative Districts Rasterized with GEOPS

Query: SELECT country_name, count(*) FROM tweets, boundaries WHERE ST_INTERSECTS(tweets.geom, boundaries.geom) GROUP BY country_name;

	PostGIS (cached)	GEOPS-DB (CPU – 1 thread)	GEOPS–DB (CPU – 6 threads)	GEOPS-DB (GPU)
Tweet Lookups	100,000	117,851,856	117,851,856	117,851,856
Time (sec)	118.4	1.471	0.2410	0.1805 (.1722 kernel execution)
Lookups per second	844.53	80.171 million	489.011 million	652.92 million (684.39 million w/o memory transfer)
Speedup (versus PostGIS)	--	94,929.7 X	579,033.3 X	773,116.4 X (810,379.74 X)
Platform	Intel i7-3930k 3.2 Ghz (single threaded) 16 GB RAM	Intel i7-3930k 3.2 Ghz (single threaded) 16 GB RAM	Intel i7-3930k 3.2 Ghz (6 threads) 16 GB RAM	Nvidia 560 Ti 448 cores 1.25 GB RAM

RENDERING HEATMAPS



- GEOPS uses CUDA to render heatmaps of "relative intensity" of text terms and other attributes over geographic space
- "Matches" projected to one raster, all observations to another
  - If using multiple GPUS, these "point" rasters are then sent to one GPU to render final heatmap
- Amazon EC2 instance can project 1.2 billion points / sec (2 X C2050 GPUS)
- Variable-sized Gaussian filter is then applied to each raster
  - Scales in constant time for # of features, linearly by raster area and geometrically by filter size
- First raster is divided by second to get relative intensity
- One GPU can render 1920 X 1080 heatmap in 0.05 seconds (20 frames / sec)

THE GRIM TWEEPER (AKA THE POOR MAN'S FIREHOSE)

- The Grim Tweeper is a distributed cloud harvester for Twitter that allows for collection and geotagging of tweets within user-defined bounding circles
- 3 Parts:
  - Cloud Harvesters on EC2 Micro Instances ($0.02 / hr each)
  - Master node that collects tweets from harvesters and geotags them in real-time
    - Tweets are then placed in temporally partitionedPostgreSQL tables
    - A plug-in module to GEOPS allows streaming tweets onto GPU in real-time
  - Web front end allows for control of harvester instances and real-time monitoring of their output (volume and location)
- Tweets geotagged using in-memory hash table of GeoNames database and user-provided location tags (along with location of bounding circle – IP address determined)
- 15 harvesters over half of US collected 50 million tweets / day, suggesting that most of the firehose is collected (300-400 millions tweets per day for whole world)
  - Impossible with standard API access (without firehose access)