Related projects: STINGER, GraphLab, MapD, Titan,

GraphLab, C++ distributed machine learning at scale, developed 4 years at Carnegie Mellon
NetworkX, python based,
SNAP Stanford Network Analytics Project, C++, with Python interface
Lumify -- not related. Tool for data fusion, analysis, and visualization. For example, connecting entities together, assigning properties to nodes/edges. I think this can extract some entities and relationships automatically (NLP?). Doesn't use graph algorithms. Does use graph *layout* algorithms.

Databases

- SciDB - Top-performing for numeric data
- Accumulo - Top-performing for heterogeneous data
- H-Store - OLTP, main-memory (no backing store?)
  - Anti-cacheing: all data initially resides in memory, and when memory is exhausted, the least-recently accessed records are collected and written to disk.
- Titan, Neo4J, other TinkerPop databases
- PostgresSQL
- HBase - emphasizes random access?
- Apache Phoenix - on top of HBase. Accelerated access to HBase data. Not sure of specifics.
- NewSQL - Google Spanner,Clustrix, VoltDB, MemSQL, Pivotal'sSQLFire and GemFire XD, SAP HANA,[12]FoundationDB, NuoDB,[13][14]TransLattice, ActorDB,[15]and Trafodion.[16]
- BlinkDB - approximate queries. Give a time bound or error bound and it will sample data and return it to you within the bound.
- MonettDB - column store
- H2O - dynamic selection of row vs. column store layouts based on workload history (sliding window of N past queries). Dynamically groups columns together based on what attributes are accessed together. This is for dense data-- very different from sparse situation where rows have different columns and there can be millions of columns.
- Too Many DBs!

AccumuloGraph
Apache Hama
TinkerPop3 - graph computing interface

Graphite - real time online graphing of time series data

Cascading - engine on Hadoop. Has Accumulo data source binding.

Articles

- GraphComputing article
- Solution to SuperNode? (this is the problem where one monster vertex in a graph has a bajillion edges)

OpenCL - see OpenCL file

Charting data / viz

- Dimple d3 Javascript library - put in website code, runs on the raw dataset

Biology application from Dr. Ganesan

> Hi Dylan,
> The protein scoring/gene scoring applications are fairly common in computational biology. I was referring to an application called HMMER, that uses Hidden Markov Models to score protein sequences, that uses Viterbi algorithm
>
> http://hmmer.janelia.org/
>
> Here is an article about the web server running the software that users can run jobs on.
>
> http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3125773/
>
> There are other similar software for gene similarity search that uses similar dynamic programming algorithms
>
> http://blast.st-va.ncbi.nlm.nih.gov/Blast.cgi
>
> The scoring algorithm is quite compute intensive so either the Iterator function might be a good choice to implement or the Reducer in the Map-Reduce framework. I can give a short presentation on the algorithm and the application this week. Hanyu works on accelerating the HMMER algorithm on GPUs..integrating it with a large-scale database will be an interesting complement to the current work.

MapDpaper --

- Since we can’t increase clock speed due to heat and space constraints, increase the
  - chip complexity: branch prediction, pipeline size, etc.
  - number of cores -- Yea parallel programming
- GPUs designed for graphics operations: 1M identical matrix ops/sec. Titan has 2668 cores, $1000, 500W, 5 Tflops single precision
- Limited memory on GPUs - about 6GB
- MapD is a column store DB,
- hierarchy: GPU memory < CPU memory < hard disk
- Avoid PCI transfers GPU->CPU when possible. For example, when querying for data on GPU that has not changed, take it from main memory instead.
- Targeting of where the query should go. Maybe directly to a GPU for rendering? Native support for visualizations like histograms and time series.
- Heavy Query optimization and planning for how to execute the query in a kernel. Caches execution plans so they can be reused.
- During low-activity times, runs query simulations to keep track of approx. how many results queries will generate
- SQL syntax

also MapGraph