**A) Cover page** (Half a page to one page). Includes the following:

**a) Project** Topic Team Member **Information** (Names and Emails) (1 point)

**b) Problem Description**, challenges especially in terms of software and algorithms development, motivation, any novelty (in terms of algorithms, math, etc.) compared to any prior work (2 points) Yani

**c) Project Timeline** (what you by the proposal deadline, by phase I, by phase II, and what you expect to do by the Phase III deadline. Add any information the course staff and supervisors should be aware of (any change of plan since proposal, phase I and phase II deadlines and the reasons (2 points). Yani

------------------------------------------------------------------------------------------------------------------------

B) **Report** (4 to 6 pages): Includes the following:

**a)** Section 1: abstract your report paper/report ( between 100 and 300 words). (5 points)

**b)** Section 2: Introduction and background i.e., prior work summary, the key novelties of your work) (half a page to 1.5 pages. (15 points) Ryan

Yani add Geo aware CDN

**c)** Section 3: Your Framework:

Formal (mathematical) problem description, e.g., for optimization, a formal goal and the set of constraints. Yani

Challenges of the problems.

Presentation and analysis of your solutions, i.e., models, algorithms, mathematical models and solutions.

Analysis of time and memory complexities. For technology dependent parameters, such as power consumption try your best to model them in high level in order to write the optimization goal formally (mathematically).

\*\*\*Write the corresponding pseudo-codes, flow charts, etc. (25 points)

**d) Implementation and Experimental Results** (50 points) including the following:

d1) Description of the experimental setup (packages you used, languages, and tools you wrote, designs you tested, etc.) (5 points)

d2) software implementation (35 points)

For your software implementation: explain the structure, languages,

Please also look the following additional code requirements for phase III: Your code has to have a C++ part. That part is yours to choose, but by default, you may implement a bottleneck in C++ so it would be faster.

The **C++ part** should including the following: a) composition [will discuss], b) inheritance, c) virtual, [dynamic binding] and also static binding, d) pure virtual (abstract class), e) constructors/destructors, and initialization, f) inline functions, g) an array of function pointers [ of at least size 3, meaning 3 function pointers], h) shallow copy, i) deep copy, j) multi-D arrays, k) at least 4 different STLs (e.g., any 4 of vector, deque, forward\_list, map, set, etc.), l) C++ reference. m) directives, including conditional directives to avoid multiple inclusion, n) debug/test mode (using a lot of printing for debugging, but using conditional directives, o) friend.

Your code has to have a Python part.

The **Python** part should include the verification code that tests your code, e.g., for corner and some random cases. Ideally your Python will restart your whole tool, gathers the results, and displays them in a file, using graphical means. it could also help with user interface (e.g., gui). Feel free to use any packages.

Both C++ and Python should have well-drafted comments.

The rest of your tool, could use any packages, tools, languages, etc.

d3) experimental results (10 points). The results should be in terms of tables, waveforms, diagrams, etc. Comparison with baseline approaches would be highly recommended and (gets extra credit) to show your work outperforms the prior work. This baseline comparison, however is NOT mandatory. The results needs to be analyzed and observations described.

**Deliverables:** Your code and your report. Make sure to add any of your github addresses for your code.