CCC Visioning Study:

System-Level Cross-Layer Cooperation to Achieve Predictable Systems from Unpredictable Components

www.relxlayer.org

Executive Summary

- Continued scaling → unpredictable components
- Traditional solutions are too expensive
  - **Key problem:** spend energy when designs energy limited
- Cannot solve at any single level
- Opportunities are cross layer, exploiting information
  - Memory systems offer inspiration
  - Logic: bigger challenge, but big payoff
- **Our goal:** community consensus on
  - Potential vision
  - What can be done
  - Research vector: make-a difference research questions and solution capabilities that help realize the vision
  - Articulate for funders, congress, lay public

Outline

- Goal of CCC visioning exercise
- Current, broad study vision
- Examples
- More focused vision
  - Energy margins → information margins
- March meeting context
- Plan for reaching the goal of visioning exercise (Carter)
  - this meeting, October meeting

CCC Visioning Exercise

- Opportunity for an (emerging) community to develop and articulate its vision
  - Consensus and leadership
    - What should be done?
    - What are the key challenges to be going after now?
    - What are the big target problems to solve?
  - Articulate capabilities, investment opportunities
    - Here is what we think can be done now
    - Here is the impact this can have
    - Here is a community that could make progress on this

CCC Visioning Exercise

- Make the case
  - Scientifically
  - Practical impact
  - Motivate lay public
- How would you get this?
  - What should program(s) look like?
  - How recognize work advance vision?

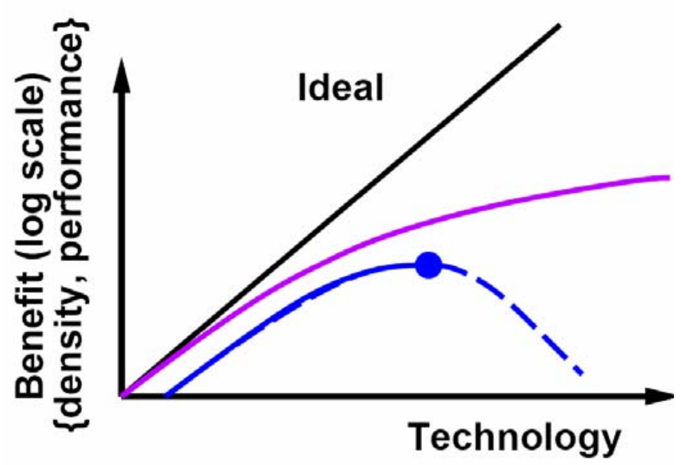
Our Goal

- Build consensus on the vision
  - What should be done? (priority, leadership)
  - What can be done? (capabilities)
  - How should it be done? (program org.)
- Communicate vision
  - Funding agencies, congress, lay public
- To enable:
  - Programs that harness our expertise to make the world better
    - Safely offer greater capabilities for limited dollar and energy budgets

Current, Broad Study Vision

What trying to do?

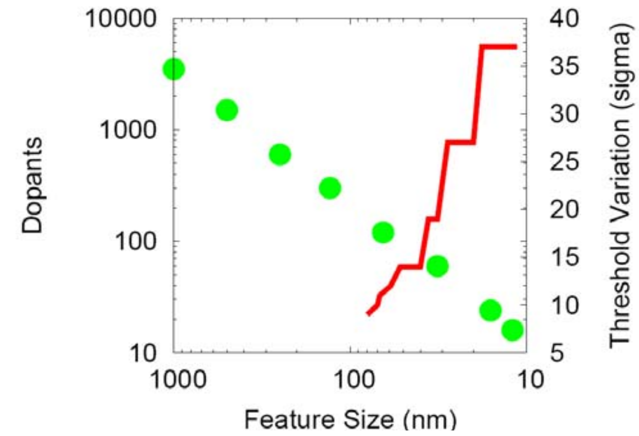
- Allow continued scaling benefits
  - Reduce energy/operation
  - Reduce $$/gate
  - Increase ops/time with limited power-density budget
- While maintaining or improving safety
- Navigate inflection points in
  - Energy and reliability



How is it done today?

- Demand reliable, consistent **device** operation
- Margin for worst-case device effect
  - Of billions, over multi-year lifetime
- Discard components when devices fail
- System-level redundancy
- Niches where above “not good enough” are small
  - Spend considerable $$, energy for reliability
    - *E.g.* Brute-force replication

Trends?



- Power-density limited components
- Fewer dopants, atoms → increasing Variation
  - Must margin over wider range of devices
- Increasing Transistors / chip
  - More things to go wrong, sample extreme devices
- Decreasing critical charge
  - Increasing upset rates
- Decreasing opportunities for burnin
- Increasing wear-out effects
- Computations increasingly deployed into critical infrastructure and life-critical roles

What can we accomplish?

- Build reliable systems from unreliable components
  - Efficiently compensate for noisy devices through cooperation of higher levels of system stack
- To quantify: how much more efficiently (less energy, less $$) can we make it?

What's new?

Ubiquitously/pervasively exploit:

1. Design prepared for repair
2. Cooperative filtering of errors at multiple levels
3. Cross-layer codesign --- Multi-level tradeoffs
   – (generalization of hardware/software)
4. Strategic redundancy
5. Differential reliability
6. Scalable and adaptive solutions

Hints of these abound, but as point solutions
rather than systematic approach.

Why do this?

- Allow scaling to continue without sacrificing safety
  - Continued reduction in energy/op
  - Continued reduction in $$/op
  - Maintain or extend component lifetimes
  - How much further?
- Allow construction of larger, dependable systems
- Make infrastructural technology worthy of the trust we place in it

Critical Questions

1. How do we organize, manage, and analyze layering for cooperative fault mitigation?
2. How do we best accommodate repair?
3. What is the right level of filtering at each level of the hierarchy?
4. Can we establish a useful theory and collection of design patterns for lightweight checking?
5. What would a theory and framework for expressing and reasoning about differential reliability look like?
6. Can a scalable theory and architectures that will allow adaptation to various upset rates and system reliability targets be developed?

Metrics, Goals, Measure and Manage Progress

- Looking for from this workshop!
  - Challenge problems
    - Formulate goals
    - Measure success, progress toward addressing

Examples

Memory Systems

- Deal with defective fabrication
  - row/column sparing
- Accommodate transient upsets
  - Error-Correcting Codes
  - Scrubbing

Memory: Cross-Layer Optimization

- Device
  - Hardening
- Circuit
  - Differential reliability
    - Replaceable core cells vs. unrepairable periphery
    - Upsettable core cells vs. ECC control logic
- Architecture
  - ECC Protection (e.g. SECDEC 12.5% overhead)
- OS
  - Periodic scrubbing
  - Map out bad blocks
- More expensive if tried to solve at **any** single level

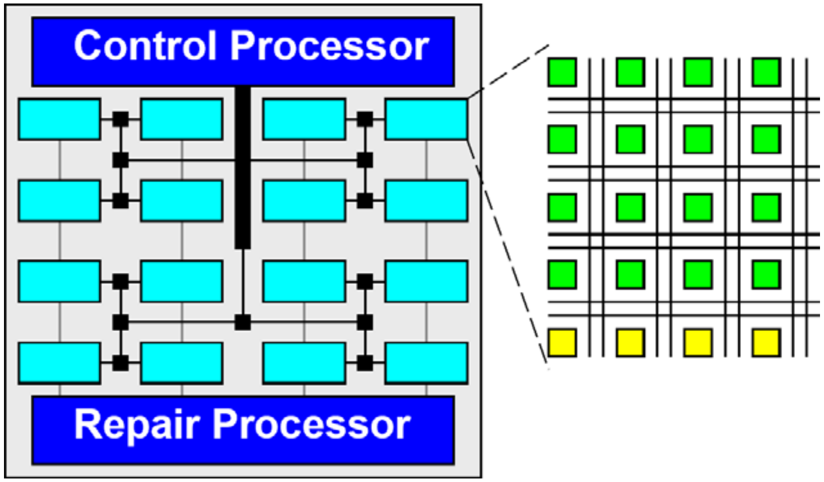
Multi-level Vision

- A traditional, ECC-protected memory
  - provides the **reliability** of large feature sizes
  - with the **density** of small memory cells
- Multi-level computational designs
  - provide the **reliability** of large-feature and large-energy devices
  - with the **density** and **energy** consumption of small-feature, low-energy devices

Computational Application

1. Prepared for repair
   - Regular, fine-grained architectures: e.g. FPGA
   - Computational model to abstract defect details
2. Errors filtered at multiple levels
   - Circuit and architecture invariants
   - Application and OS self checks
3. Cross-layer codesign --- Multi-level tradeoffs
   - Right level of filtering, handling at each level
4. Strategic redundancy
   - Invariants, end-to-end consistency, application-specific
   - Assist circuit/arch. by passing down information
5. Differential reliability
   - Management and repair circuitry built from coarser feature logic
6. Scalable/adaptive solutions
   - Tune to upset rate, criticality of computation
   - Tune level of redundancy in-system, throughout lifetime

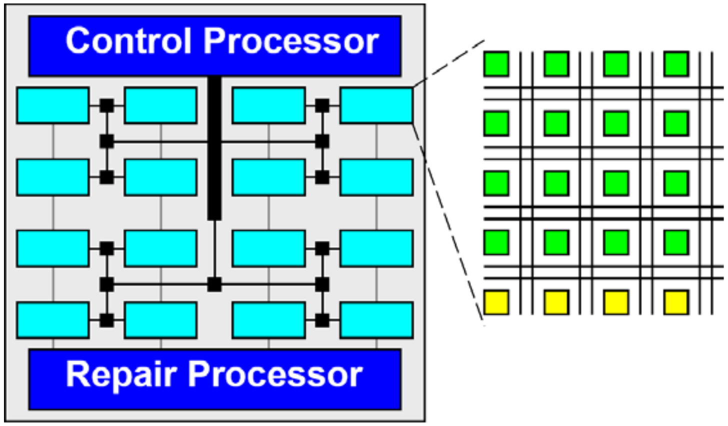
Concrete Example



- **Start with 1000 core design –** with failures every 30ms
- **Reconfigurable cores –** allow repair
- **Differential Reliability** – spend 2% area (energy) on reliable supervisor and repair processors
- **Granularity** – 200 ms intervals → >99% of cores complete interval without error
- **Application-Assisted Checking** – validation
  - Lightweight check, certificates, invariants, error magnification, safety-properties, interleaved test
- **Repair** errors during time-slice intervals

Ex: DeHon, Knight, Savage, Shrobe, Smith

Operation Example



- **Matrix Solve:** Ax=b
- **Check:** Compute residue |Ax-b|
- Transient error in arithmetic
  - May not matter → won't notice
  - Convergent algorithm → may just slow down
  - Corrupt result → detect with check, recompute
- Hard error in arithmetic
  - Re-execution → also fails check
  - Diagnose core with self-test
  - Reconfigure to repair
- High-level monitor policy effectiveness

Ex: DeHon, Knight, Savage, Shrobe, Smith

Energy → Information Margins

Vision: Energy→Information

- What doing?
  - Scaling continue: reduce energy/op
- How done today?
  - Energy margins to prevent failures
  - *E.g.* High voltages, large capacitance, replication
- Trend?
  - Increasing system size, variation, upset rates
  - Increasing energy needed to compensate variation and upsets → end of scaling benefit
- What can we accomplish?
  - Reduce energy margin to minimum [quantify]

Energy→Information Examples

Problem	Energy	Information
Vary Devices	Margin worst	Measure and avoid
Failure Edge	Margin avoid edge	Detect and recover
Lifetime Degrade	Margin worst case end-of-life	Detect and reconfigure
Vary Env.	Guess worst-case	Measure and adapt
Uncommon Events	Margin to tolerate	Cooperative avoidance

Vision: Energy→Information

- What's new?
  - Aggressively exploit information dimension
  - Replace energy margins with information margins
- Why do this?
  - Reaching inflection point where spending energy for reliability defeats scaling
- Key Questions?
  - (same as broad goal)
- Metrics?
  - System-level energy reduction permitted to achieve a level of reliability (under a specified noise level)
  - **Milestones:** ramp in reduction achieved

March Meeting Context

March Workshop

- First Meeting
- Santa Clara, CA following SELSE
- Large industry presence
- Wide ranging to make sure uncover key issues
- Organized around key questions
  – Flesh out; see what's missing

Where's the Problem?

- Work in complex, multi-dimensional space
  - Feature size, environment → noise rates
    - Noise: variation, aging, transient upset
  - Total system size, composition, application(s) → tolerable component reliability needs
  - Energy-delay-area-reliability
- Clear there are new reliability challenges
- Unclear if constituencies see primary challenge
  - In same corner of space
  - With common underlying causes/solutions
- Need to separate voices to achieve clarity

Constituency Challenges

Two goals

1. Segregation to understand forces driving most pain in each area
2. Quantify key challenge problem
   - Being clear about what advances and research
     - Address the Pain
     - Advance the Science and Engineering
   - How measure progress

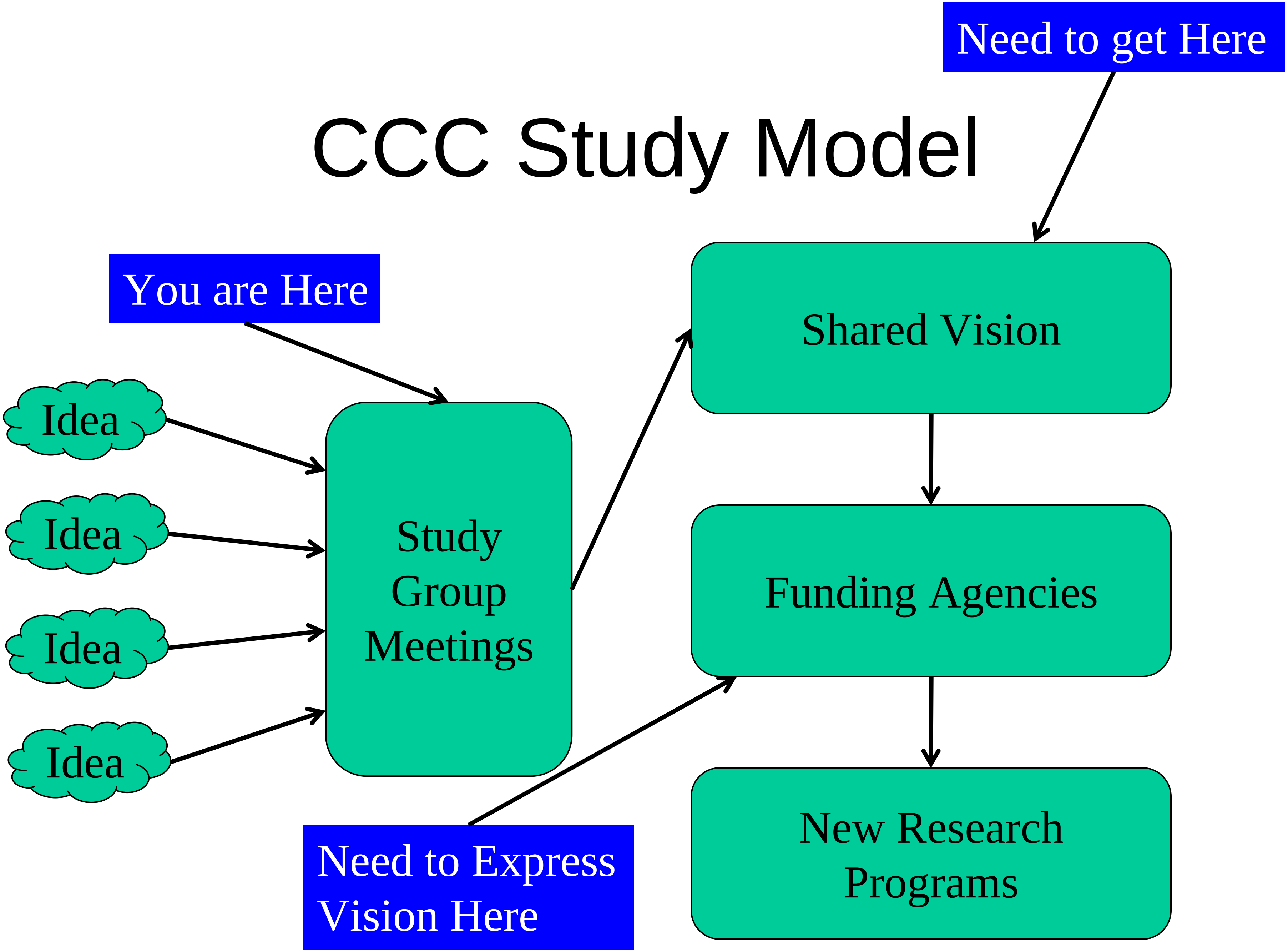
Additional Take-Away

- Key questions appropriate
- Identified need for roadmap to make challenge concrete
  - To make case
  - Give researchers grounding to understand relevant tradeoffs, calculate concrete benefits
- Caution: do no harm
  - Software complexity already leads to system failures

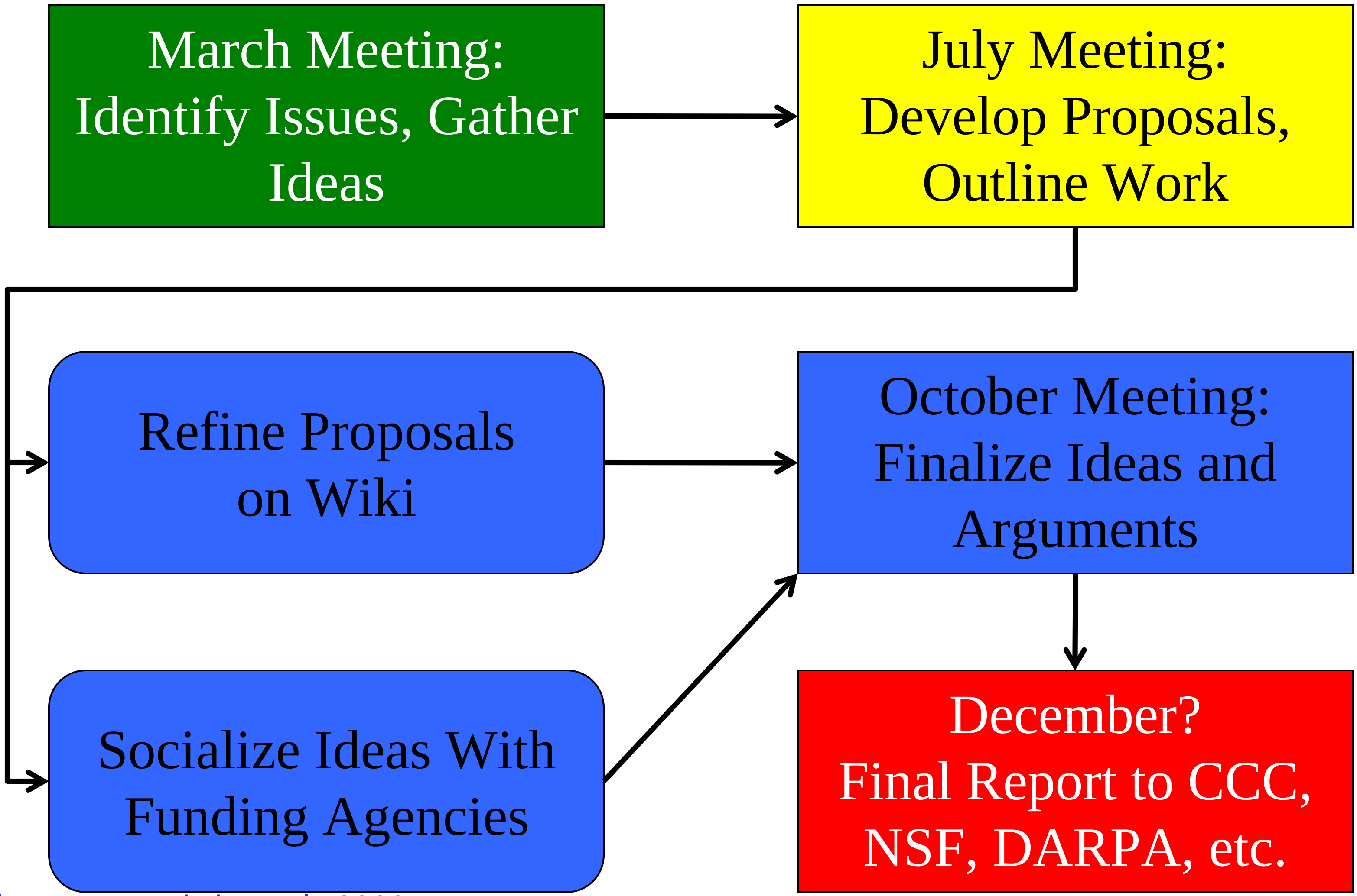
Visioning Plan

Nicholas P. Carter

CCC Study Model



CCC Study Flowchart



October Meeting

- IBM Austin Conference Center
- Thursday/Friday's in October
  - ~~1st, 2nd~~
  - 8th, 9th
  - 15th, 16th
  - ~~22nd, 23rd~~
  - 29th, 30th
- NSF PM availability may trump

What are our Deliverables?

- Within next month or so: outline of vision, plan, conclusions that organizers can use as input to discussions with funding agencies
- Before October meeting: fleshed-out vision, examples, justifications, roadmap
- By end-of-year: Final report making the case for research and funding

Desired Meeting Outputs

- From each group:
  - Outline of what you will contribute
  - Plan for what will get done between meetings
  - Volunteers to make it happen

- Think of this meeting as a kick-off
  - Real work will happen via the Wiki/email

Area Groups

- Need vision
  - Key problems
  - Ways you can use/collaborate with other areas
    - How can we make this one research area, not three glued together?
- Need examples
  - Strawman solutions
  - Costs, risks

Metrics Team

- How will we measure success?
- Can we use those metrics to explain where the problems are today?
- How do we compare ideas without hard numbers about current reliability?

Roadmap Team

- How is the problem going to change over time?
- How can we get reliability-related factors into organizations like the ITRS?
- Are there industry-safe ways to express these ideas?

Conclusion

- March meeting was about generating ideas
- This meeting needs to be about refining, focusing, selecting

- What have I missed?