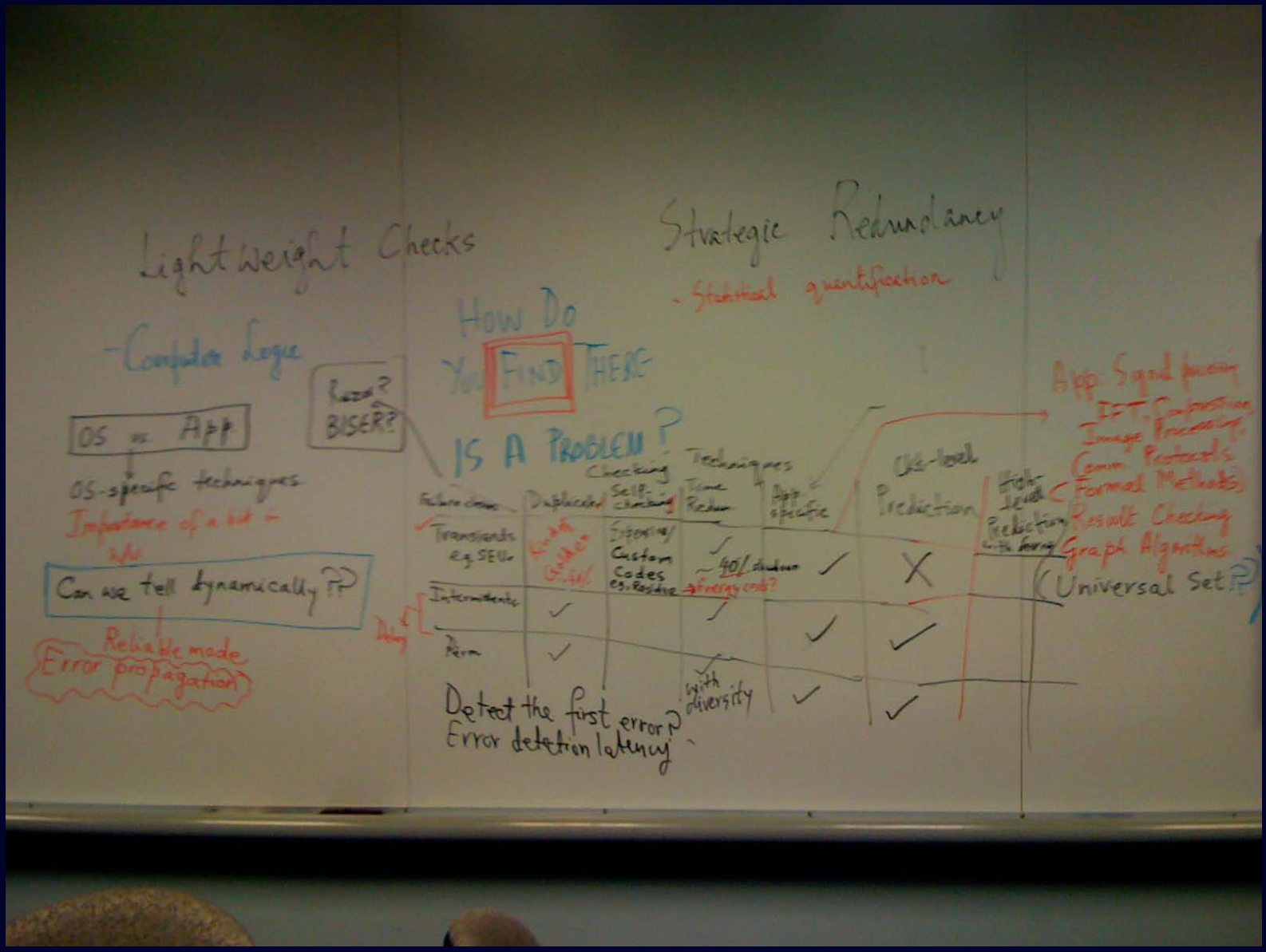
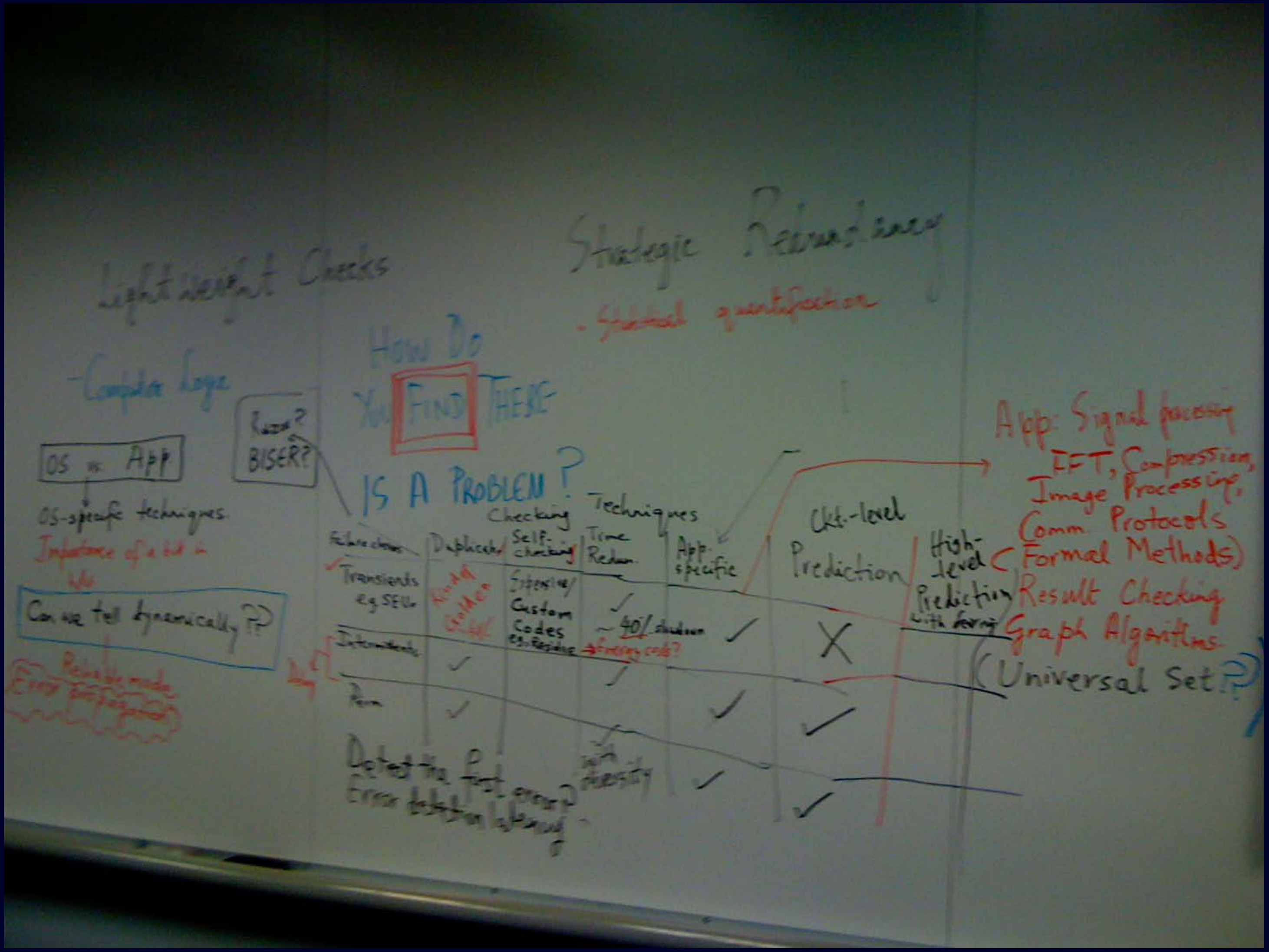
Strategic: Big Points

- FIND (detect) a problem? – Most important
- Low-cost application-specific checks exist
  - ABFT, signal processing, compression, matrix
  - Question: Universality ?
- How can we cut down cost? – Power is the king
  - “Importance” of bits (from application)
  - Error detection latency trade-offs
  - Utilize failure characteristics
    - Continuous checks may not be necessary

Snapshots


Lightweight Checks
Strategic Redundancy
- Statistical quantification
- Computer Logic
How Do You FIND THESE
IS A PROBLEM?
OS vs App
OS-specific techniques
Importance of a bit -
Can we tell dynamically??
Reliable mode
Error propagation
Techniques
Checking
Self-checking
Time Redun
App-specific
User-level
Prediction
High-level Prediction
Duplicated
Transients eg SEUs
Intermittents
Perm
Custom Codes
with diversity
Detect the first error?
Error detection latency
App Signal proc
IP, Compression, Image Processing, Comm Protocols
Formal Methods
Result Checking
Graph Algorithms
(Universal Set??)

Snapshots


Lightweight Checks
Strategic Redundancy
How Do You FIND THERE IS A PROBLEM?
OS vs App
BISER?
OS-specific techniques
Can we tell dynamically??
Checking Techniques
Duplicate
Self-checking
Time Redun.
App. specific
Ckt.-level Prediction
High-level Prediction
App: Signal processing
FFT, Compression, Image Processing, Comm. Protocols
(Formal Methods)
Result Checking
Graph Algorithms
(Universal Set??)
Transients e.g. SEUs
Intermittents
Perm
Custom Codes
Detect the first error?
Error detection latency