Characterizing Result Errors in Internet Desktop Grids

D. Kondo[1], F. Araujo[2], P. Malecot[1], P. Domingues[2],
L. Silva[2], G. Fedak[1], F. Cappello[1]

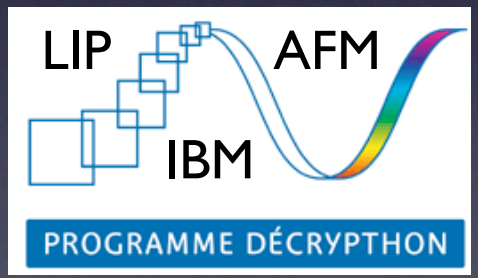
[1] INRIA, France [2] University of Coimbra, Portugal

Desktop Grids

Astronomy

Math

Biology

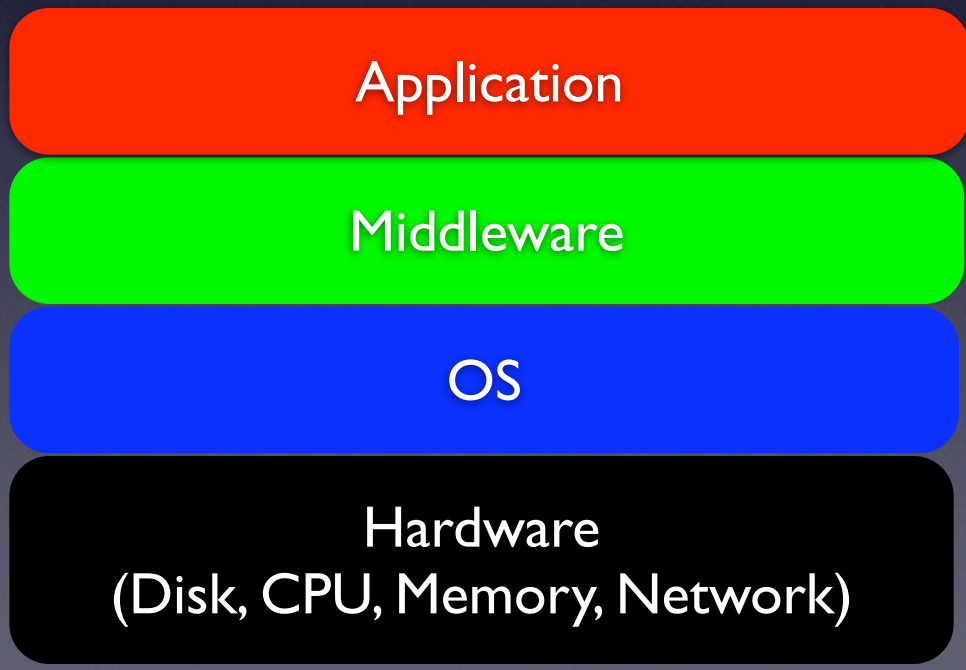


Total: ~50 applications using ~1.1 PetaFLOPS
from ~1 million active resources

Background

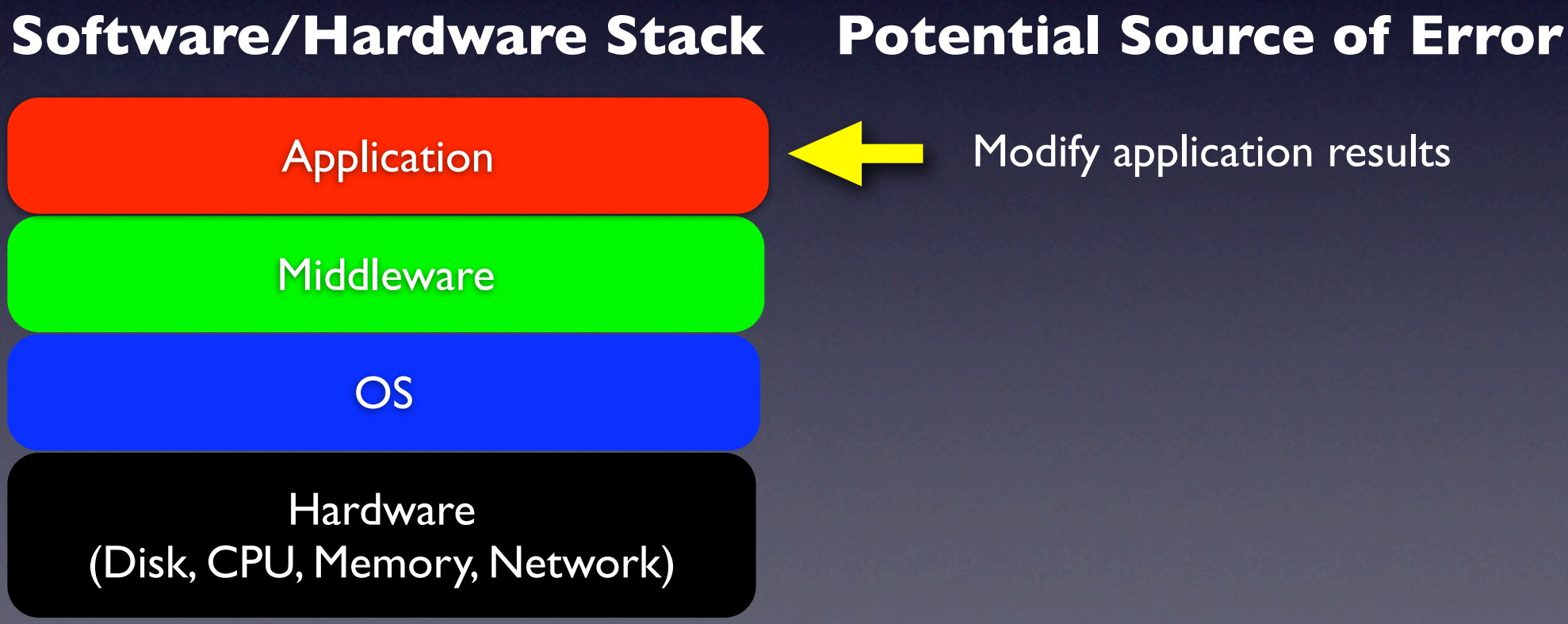
- In large-scale desktop grids involving volunteered, anonymous (and thereby potentially untrusted, insecure) resources, errors are inevitable

Software/Hardware Stack **Potential Source of Error**



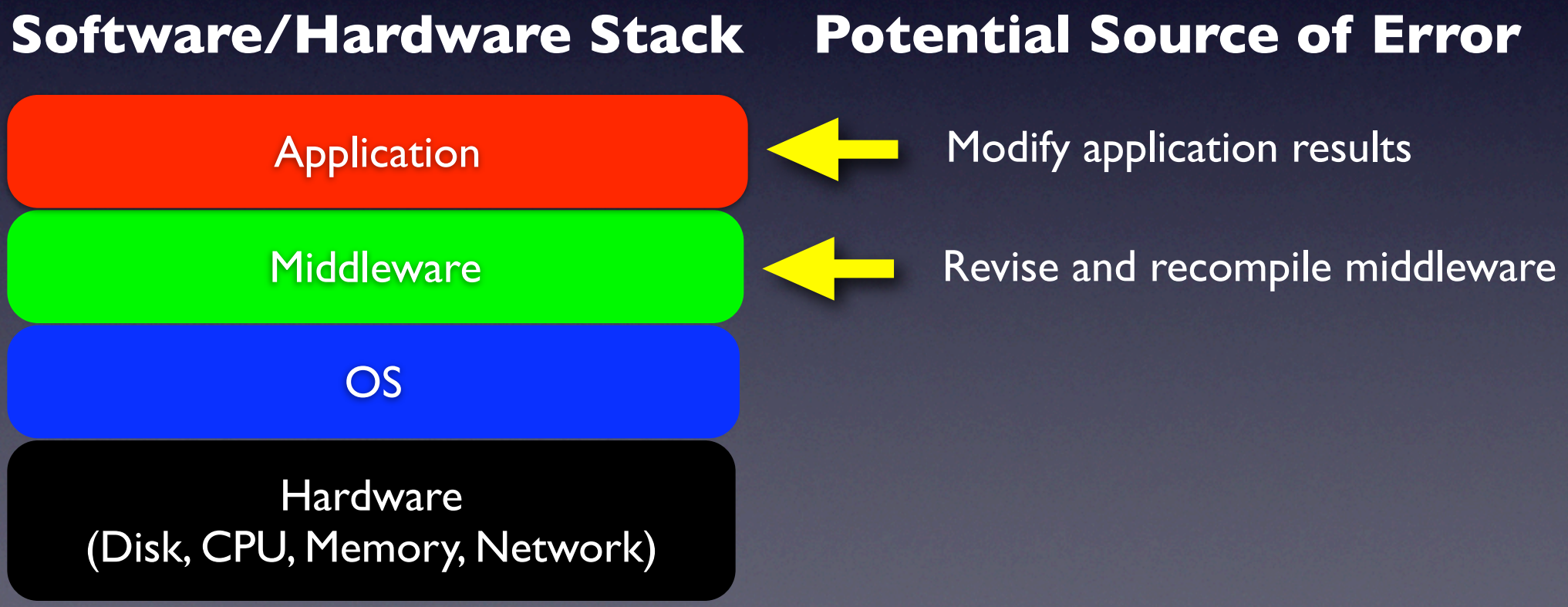
Background

- In large-scale desktop grids involving volunteered, anonymous (and thereby potentially untrusted, insecure) resources, errors are inevitable



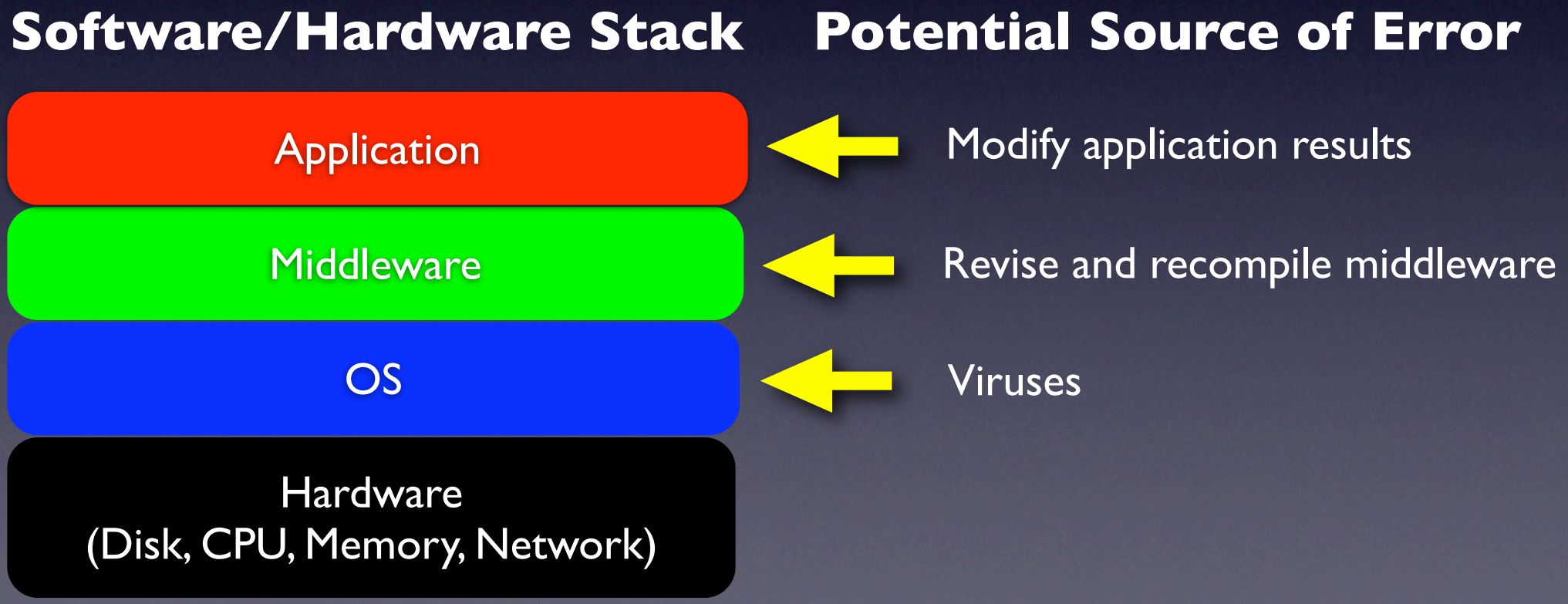
Background

- In large-scale desktop grids involving volunteered, anonymous (and thereby potentially untrusted, insecure) resources, errors are inevitable



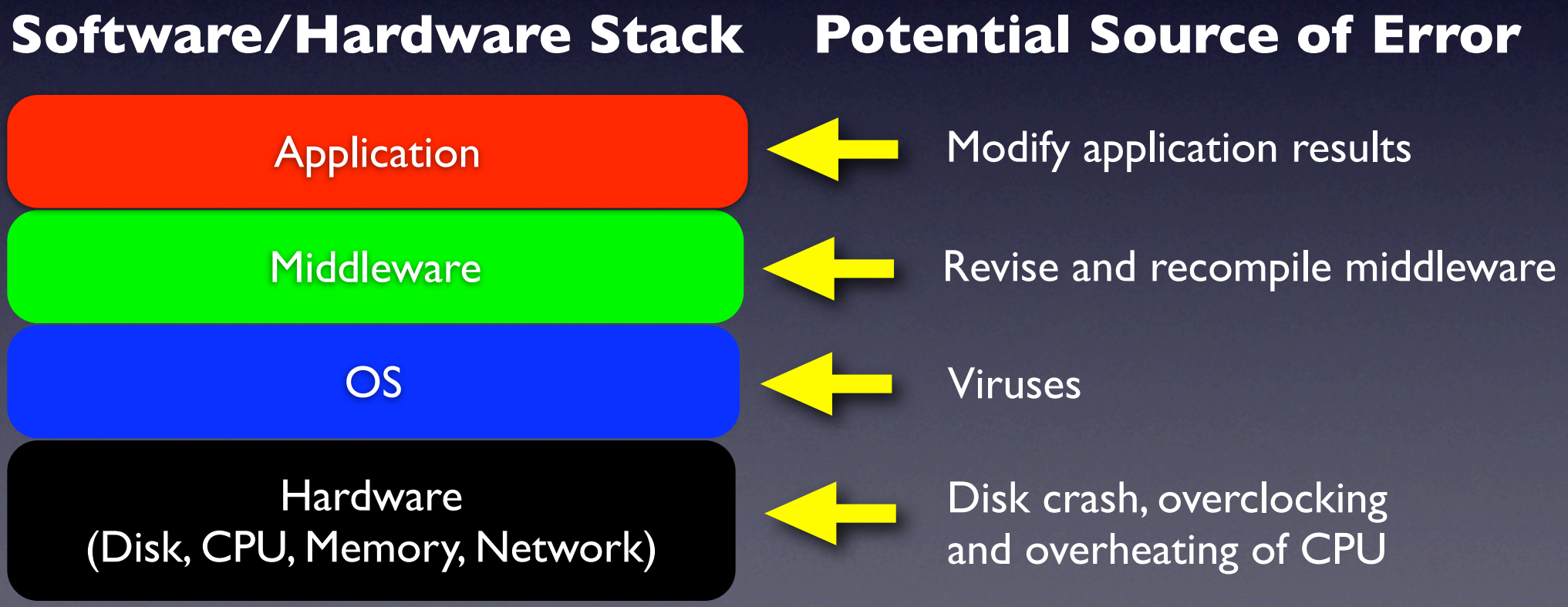
Background

- In large-scale desktop grids involving volunteered, anonymous (and thereby potentially untrusted, insecure) resources, errors are inevitable



Background

- In large-scale desktop grids involving volunteered, anonymous (and thereby potentially untrusted, insecure) resources, errors are inevitable



Motivation

- Number of application-level mechanisms for tolerating errors exist [Sarmenta, Lo]
- Effectiveness of mechanisms depend on when errors in real systems
- Yet, characterization of errors is poorly understood

Goal

- Characterize error rates in a real system
  - Frequency
  - Stationarity
  - Correlation
- Evaluate error tolerance mechanisms in light of this characterization

Outline

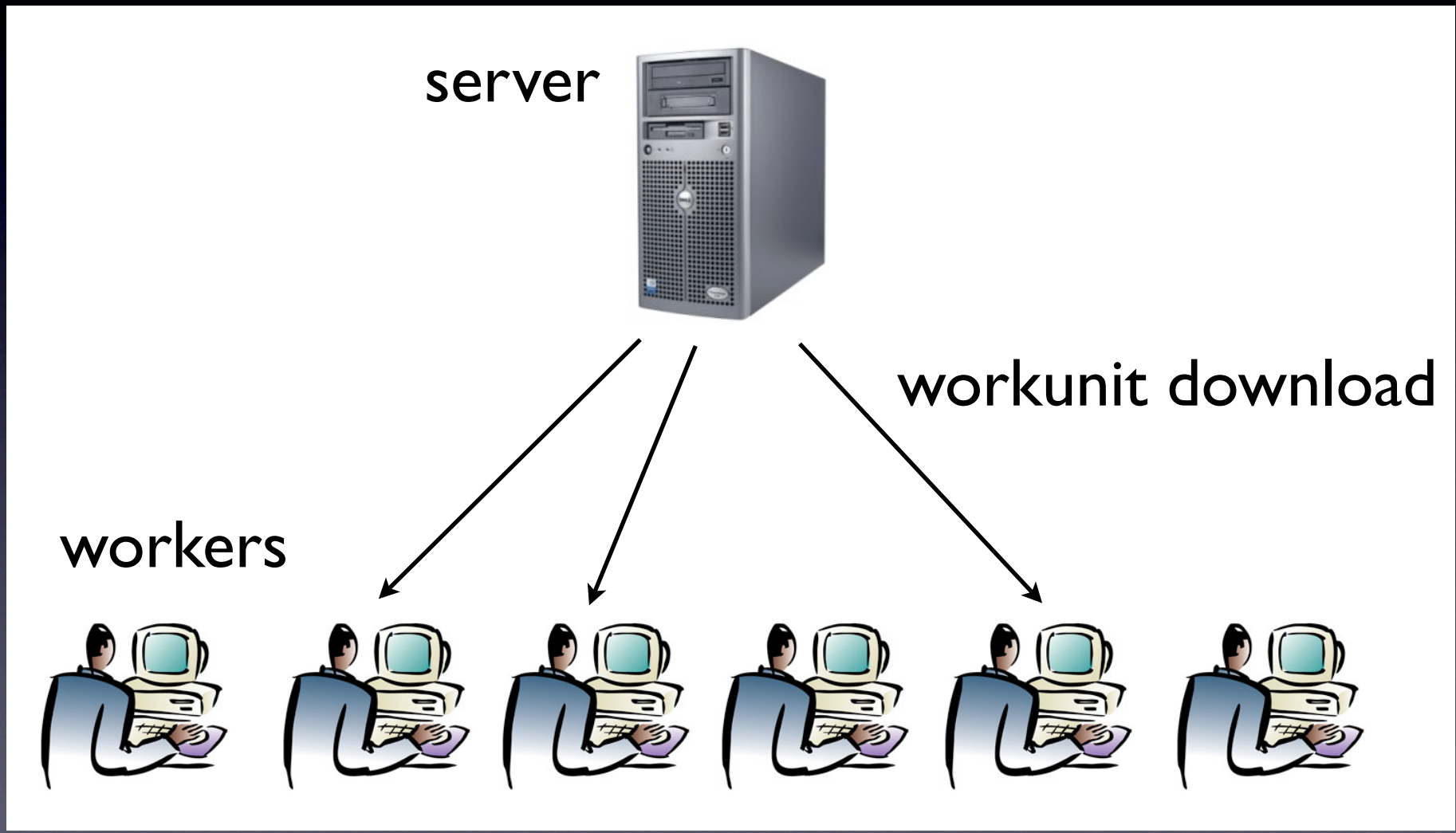
- Background Terminology
- Related Work
- Method
- Error Characterization
- Summary and Future Work

Background Terminology

server

workers

Background Terminology

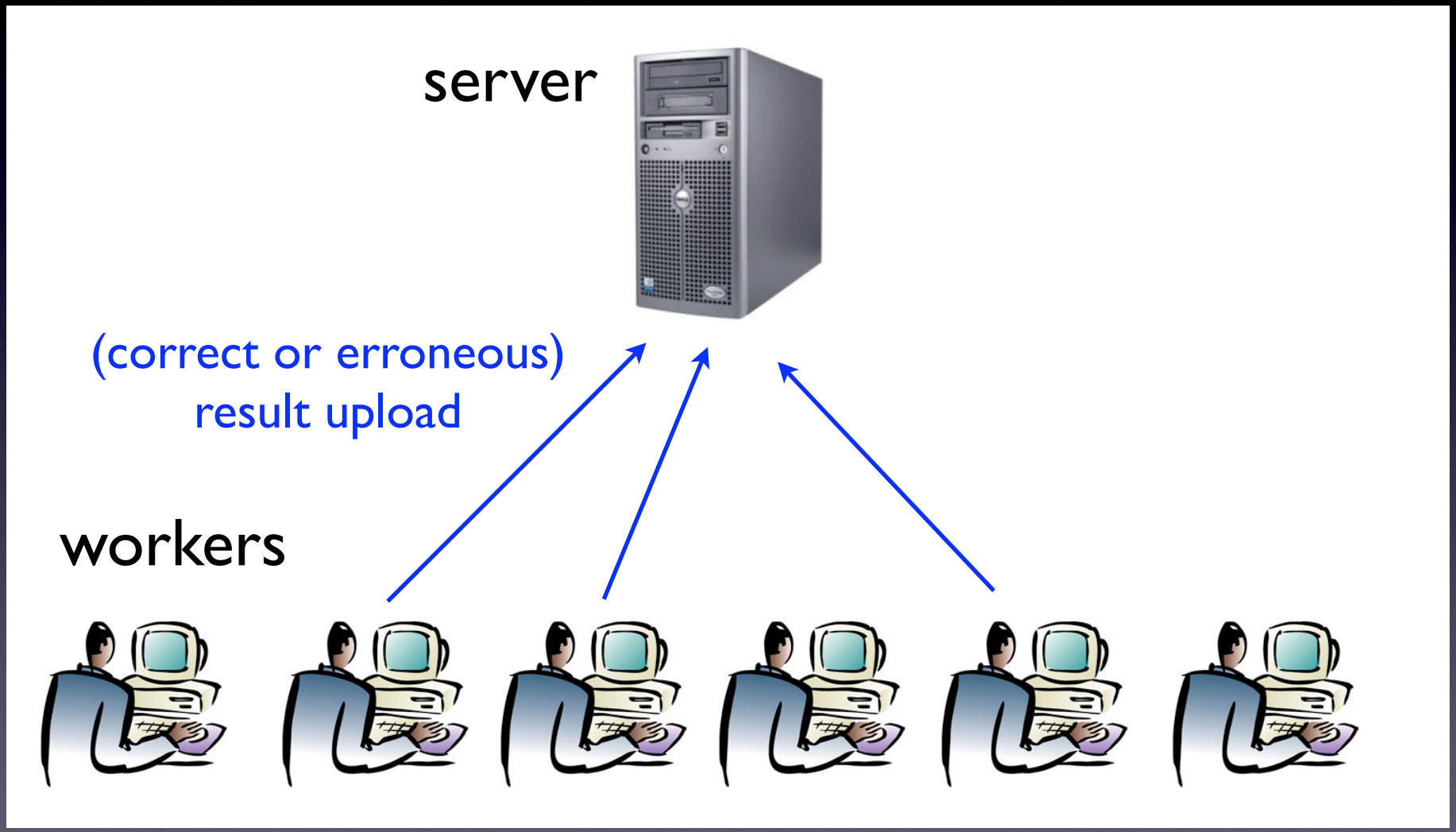


Background Terminology

server

workers

Background Terminology



Related Work

- Error Tolerance Mechanisms [Sarmenta01, Zhao01, Taufer05]
  - Majority voting
  - Spot-checking with blacklisting
  - Credibility-based methods

Majority Voting [Sarmenta01]

- Send *2m-1* instances of the same workunit to multiple workers, and the compare the results
  - Majority vote is complete after receiving *m* identical results

Majority Voting [Sarmenta01]

ε	Fraction of results that will be erroneous
φ	Probability that a worker (from the set of erroneous and nonerroneous hosts) returns an erroneous result
m	Number of identical results before a vote is considered to be complete

$$\varepsilon_{majv}(\varphi, m) = \sum_{j=m}^{2m-1} \left(\begin{array}{c} 2m-1 \\ j \end{array} \right) \varphi^j (1-\varphi)^{2m-1-j}$$

Issues

- Model assumes error rates are not correlated among hosts
- If error rate is high (>1%), much redundancy required to achieve low error bounds

Spot-Checking [Sarmenta01]

- Distribute workunit with known correct result randomly to workers
- Compare workers' result to known correct result
- If there is a difference, blacklist that worker

Spot-Checking [Sarmenta01]

ε	Fraction of results that will be erroneous
q	Frequency of spot-checking
n	Number of workunits to be computed by each worker
f	Fraction of hosts that commit at least one error
s	Error rate per erroneous host

$$\varepsilon_{scbl}(q, n, f, s) = \frac{sf(1-qs)^n}{(1-f) + f(1-qs)^n}$$

Issues

- Assumes blacklisting is efficient and effective
- Assumes consistency of error rates over time
- If error rates are low, then the number (*n*) of workunits to be computed per worker must very high

Credibility-Based System [Sarmenta01]

- Define credibility of an entity as the conditional probability of its correctness given its history of past (spot-)checks
- Workers build (or lose) credibility as they pass or fail (spot-)checks
- Compute credibility of result based on worker credibility
- Issue: assumes the error rate per host is consistent over time

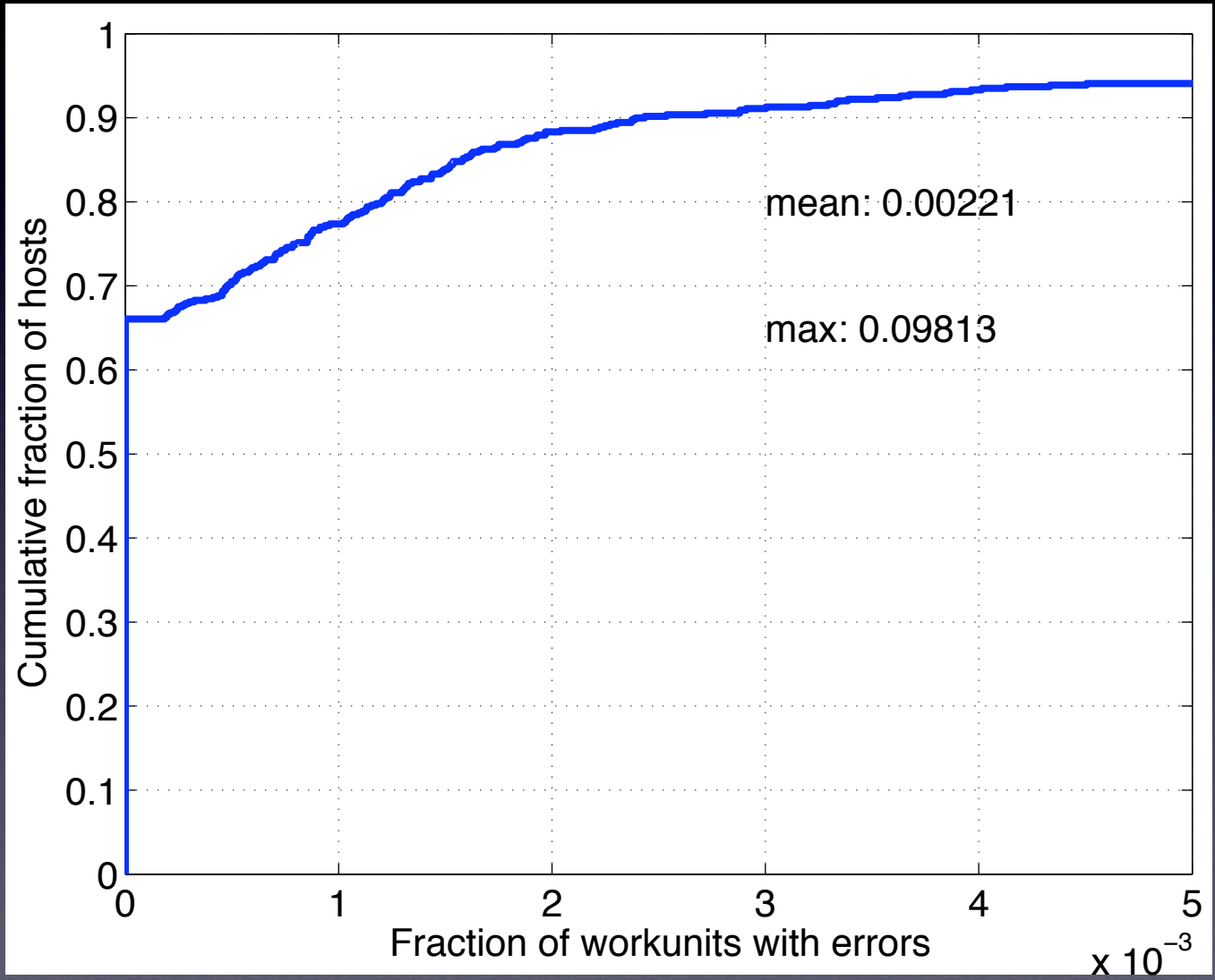
Methodology

- XtremLab: BOINC-based project for characterizing Internet desktop grids
- Application continuously computes floating-point and integer operations
- Validator conducts syntactical and semantic checks of results
- Gathered data from about 600 hosts between April - July, 2006

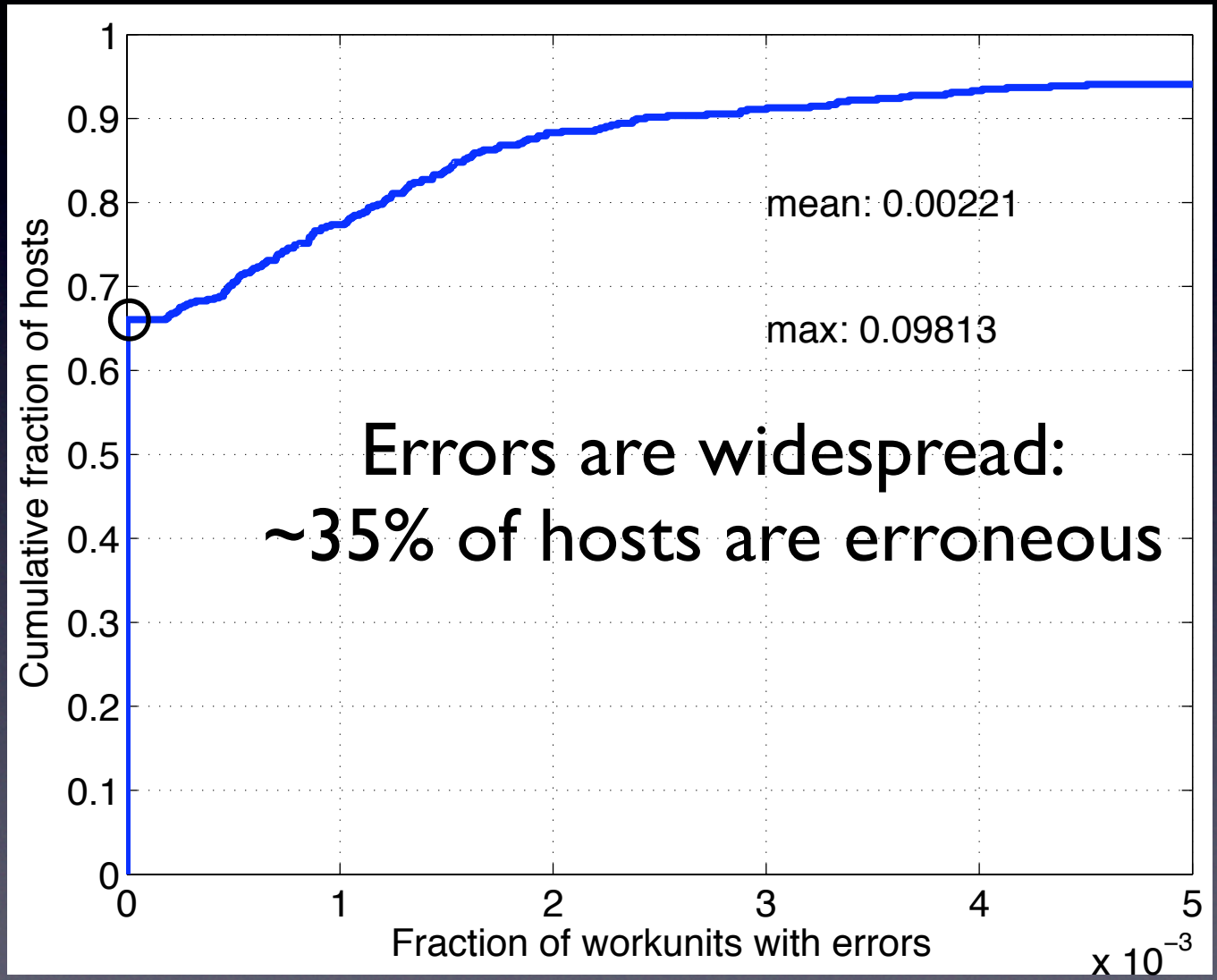
Observations and Assumptions

- Most errors manifest themselves as scrambled or truncated output
  - Likely due to I/O errors
- Detected errors would have caused a result error in a real application
  - E.g. I/O error corresponds to a corrupt write of checkpoint file

Error Rates in Entire Platform



Error Rates in Entire Platform



Implications

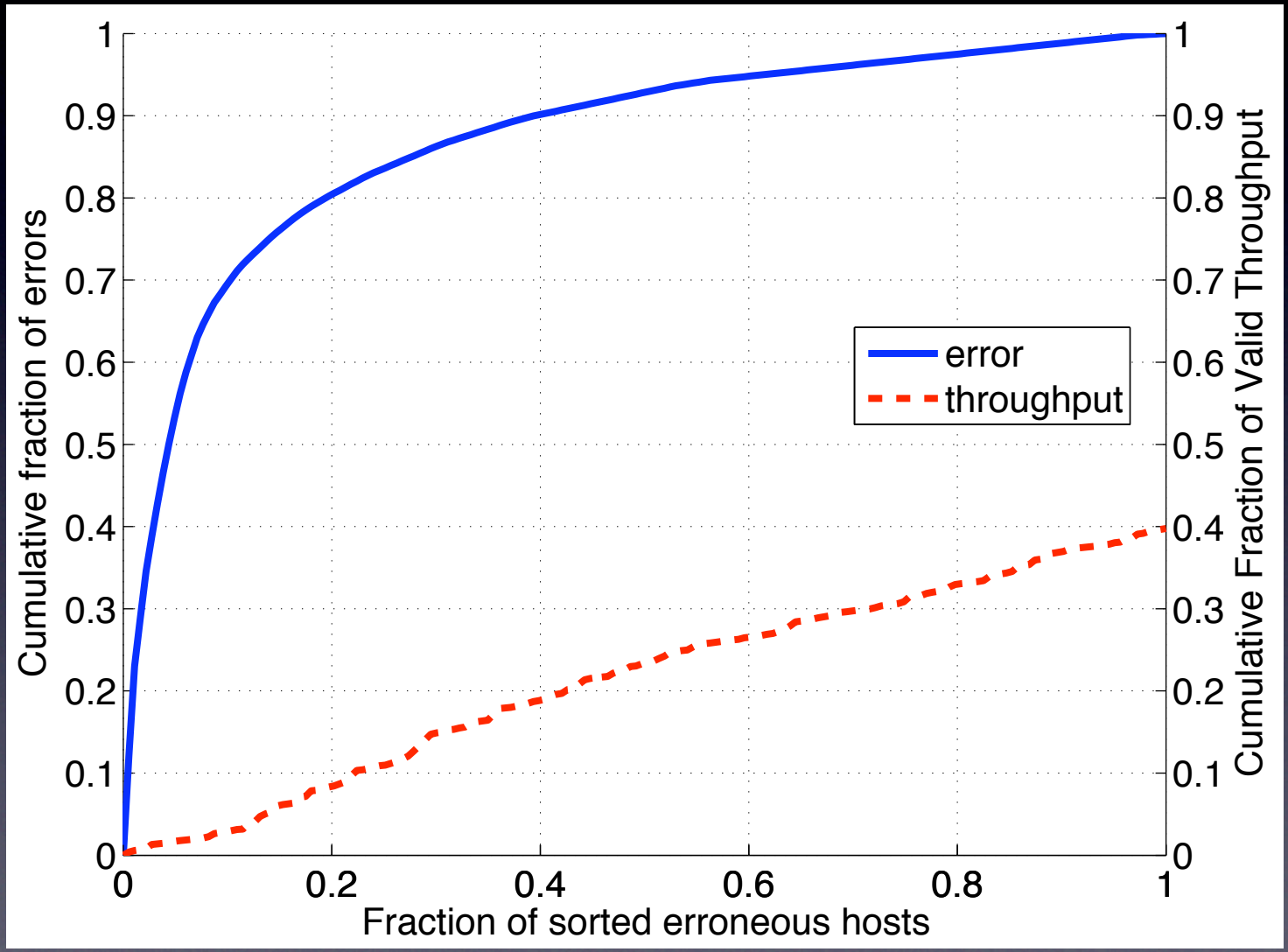
- Working example
  - 10 batches, 100 workunits each

$$\epsilon_{overall} \leq 0.01$$
$$\text{need } \epsilon_{result} \leq 1 \times 10^{-5}$$

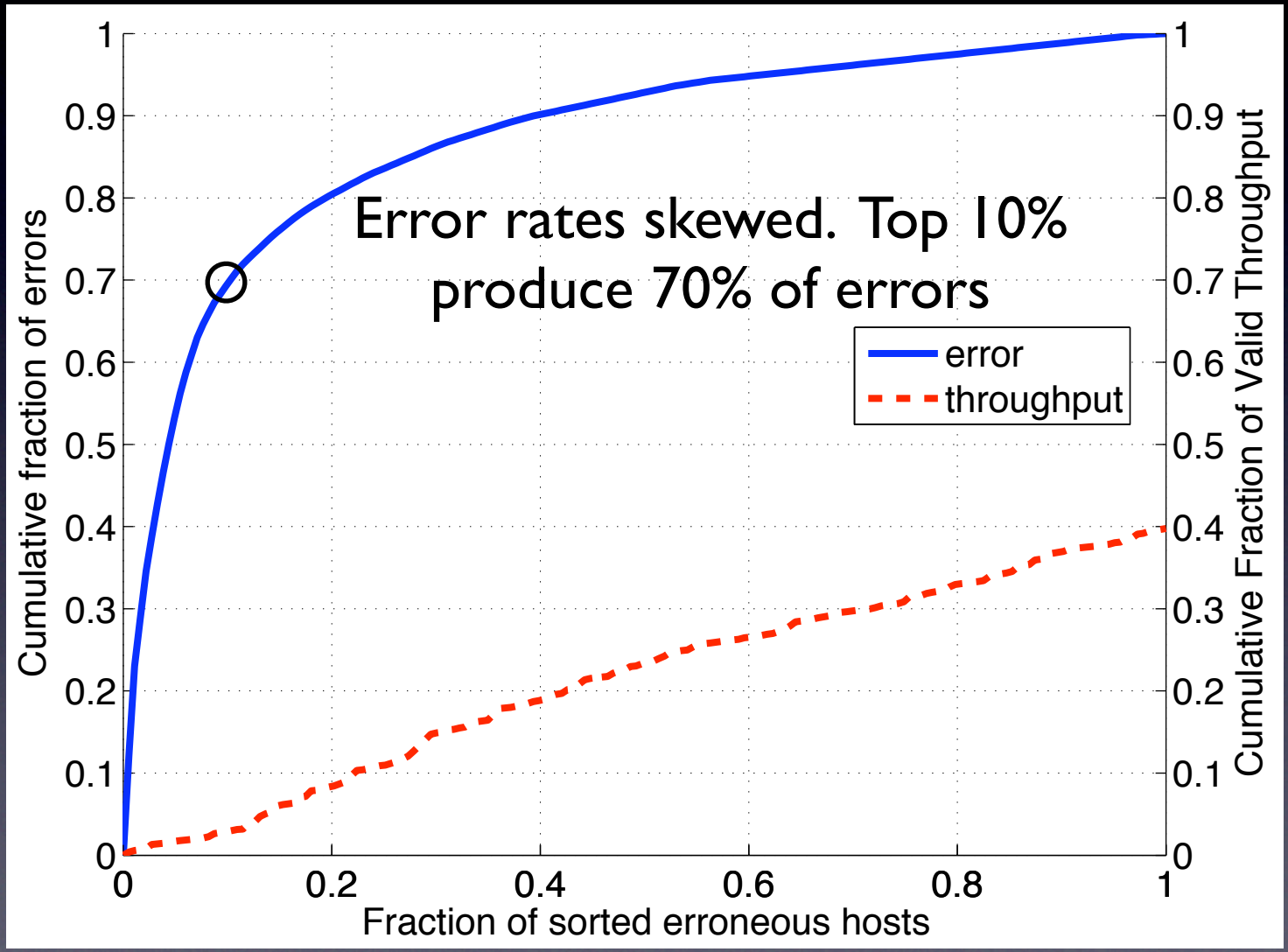
- To get $\epsilon_{result} \leq 1 \times 10^{-5}$
  - Majority vote: need majority vote (*m*) of 2
  - Spot-checking: number of workunits (n) per worker > 5300*
- Blacklisting all erroneous hosts is most likely not efficient

* q=0.10, f=0.35, s=0.003

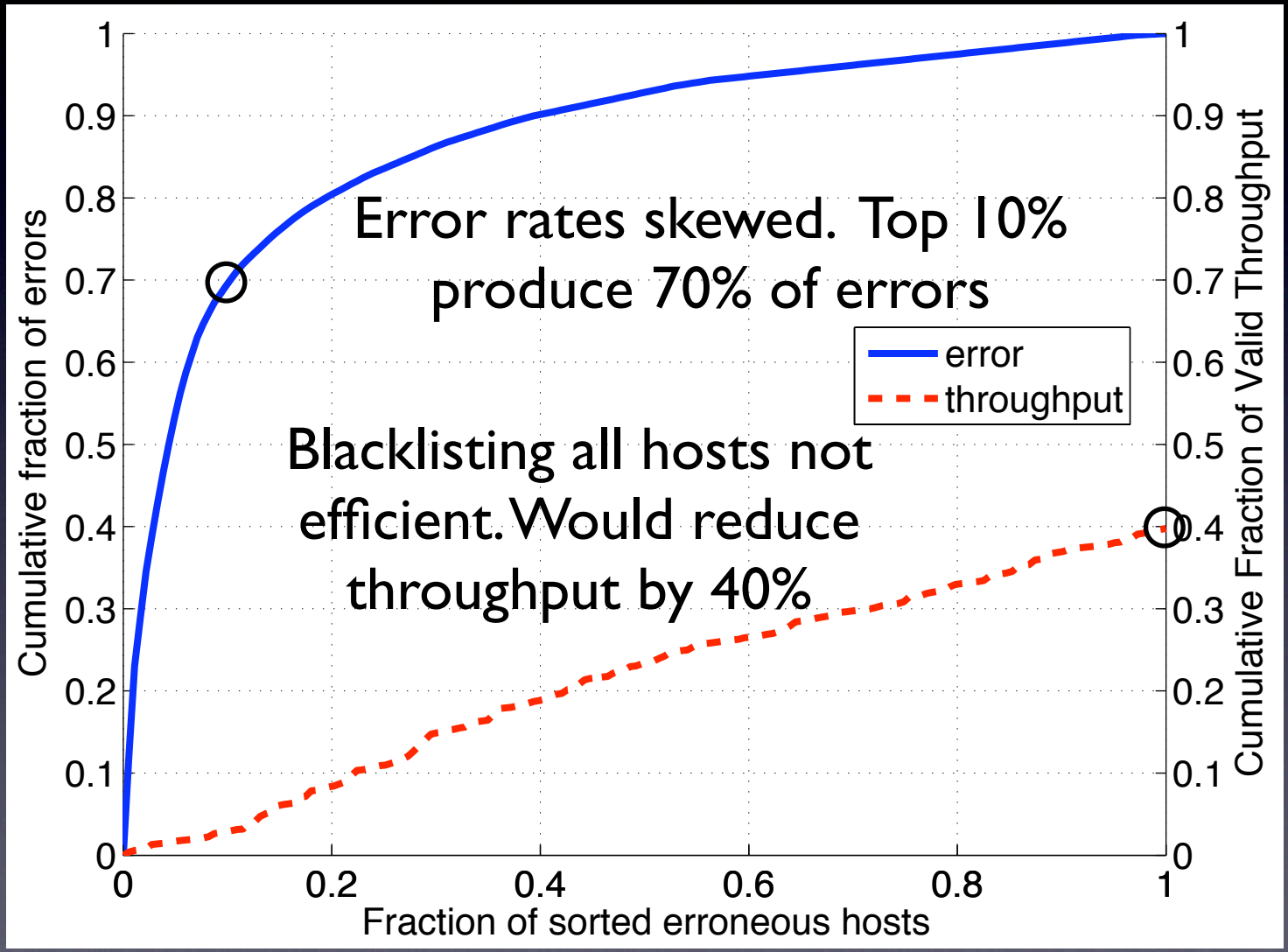
Cumulative Error Rates and Effect on Throughput



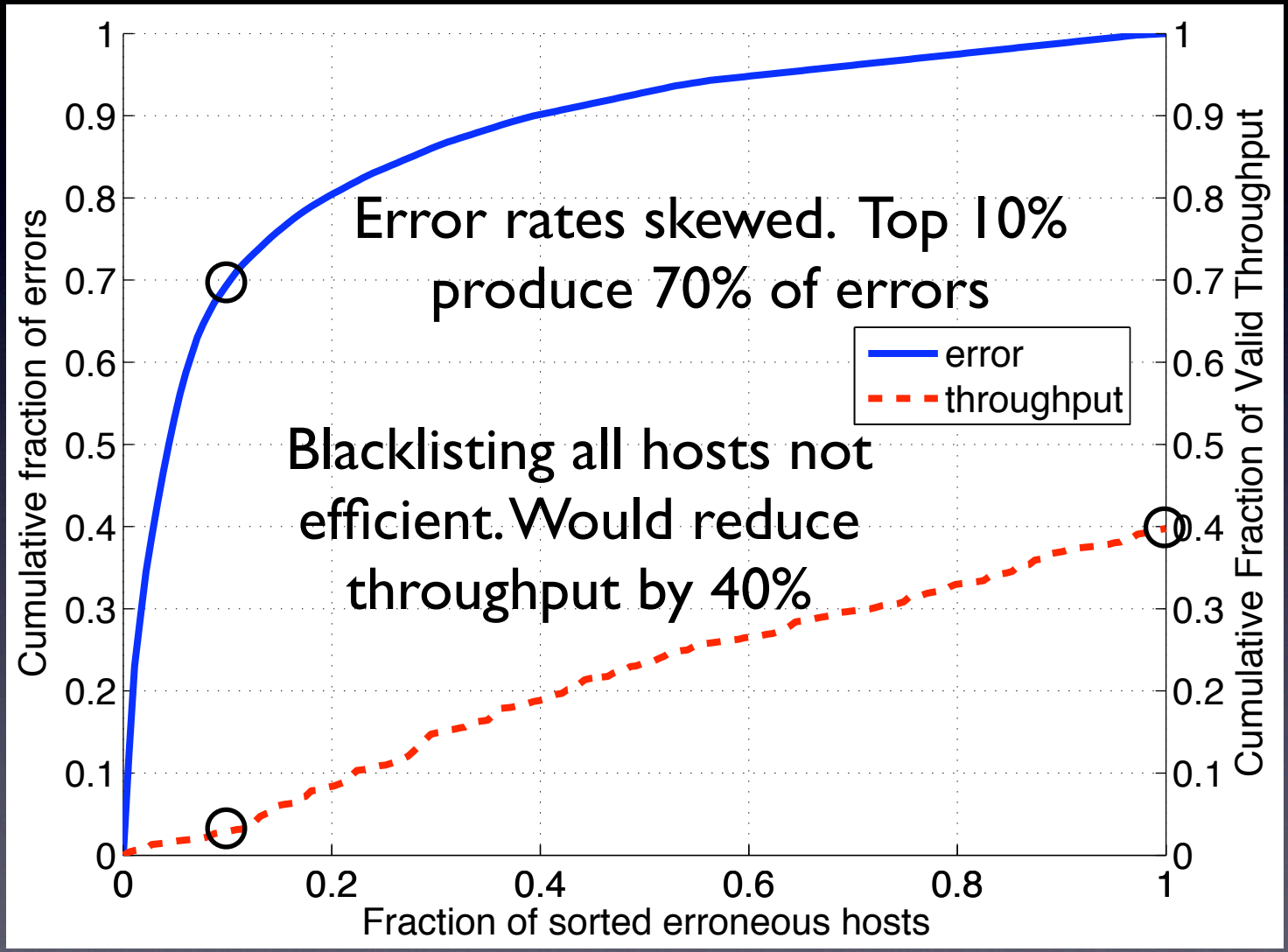
Cumulative Error Rates and Effect on Throughput



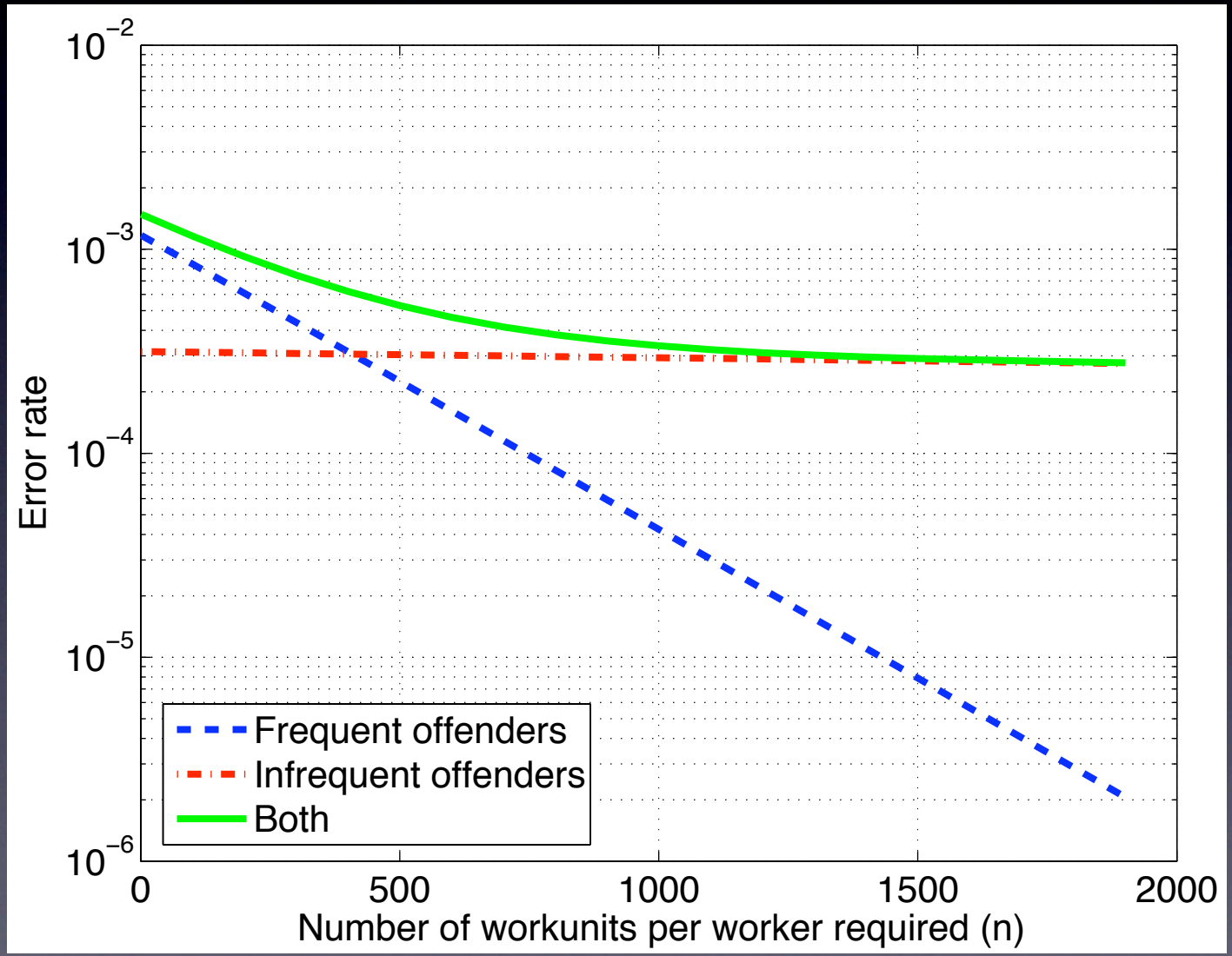
Cumulative Error Rates and Effect on Throughput



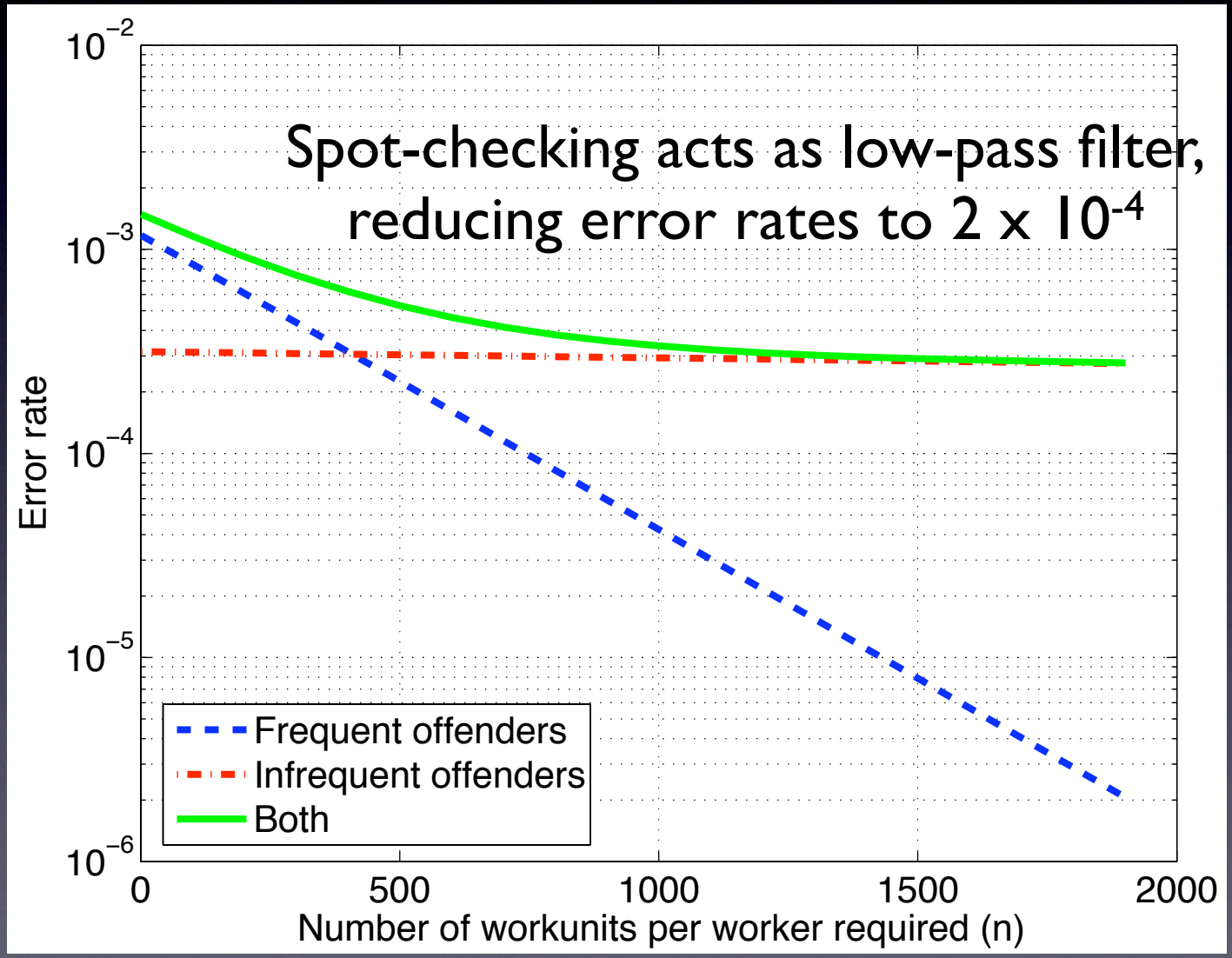
Cumulative Error Rates and Effect on Throughput



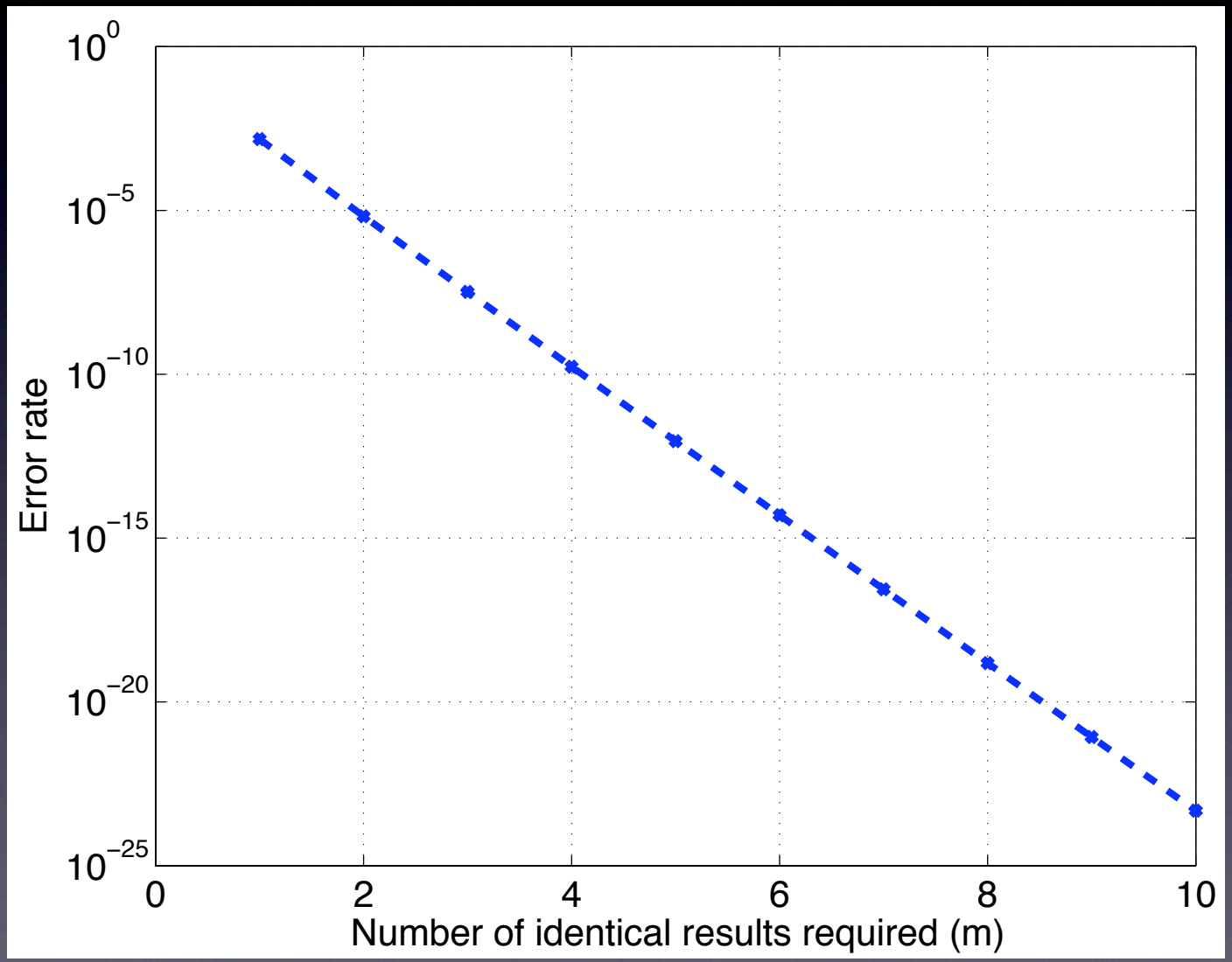
Spot-Checking with Blacklisting Revisited



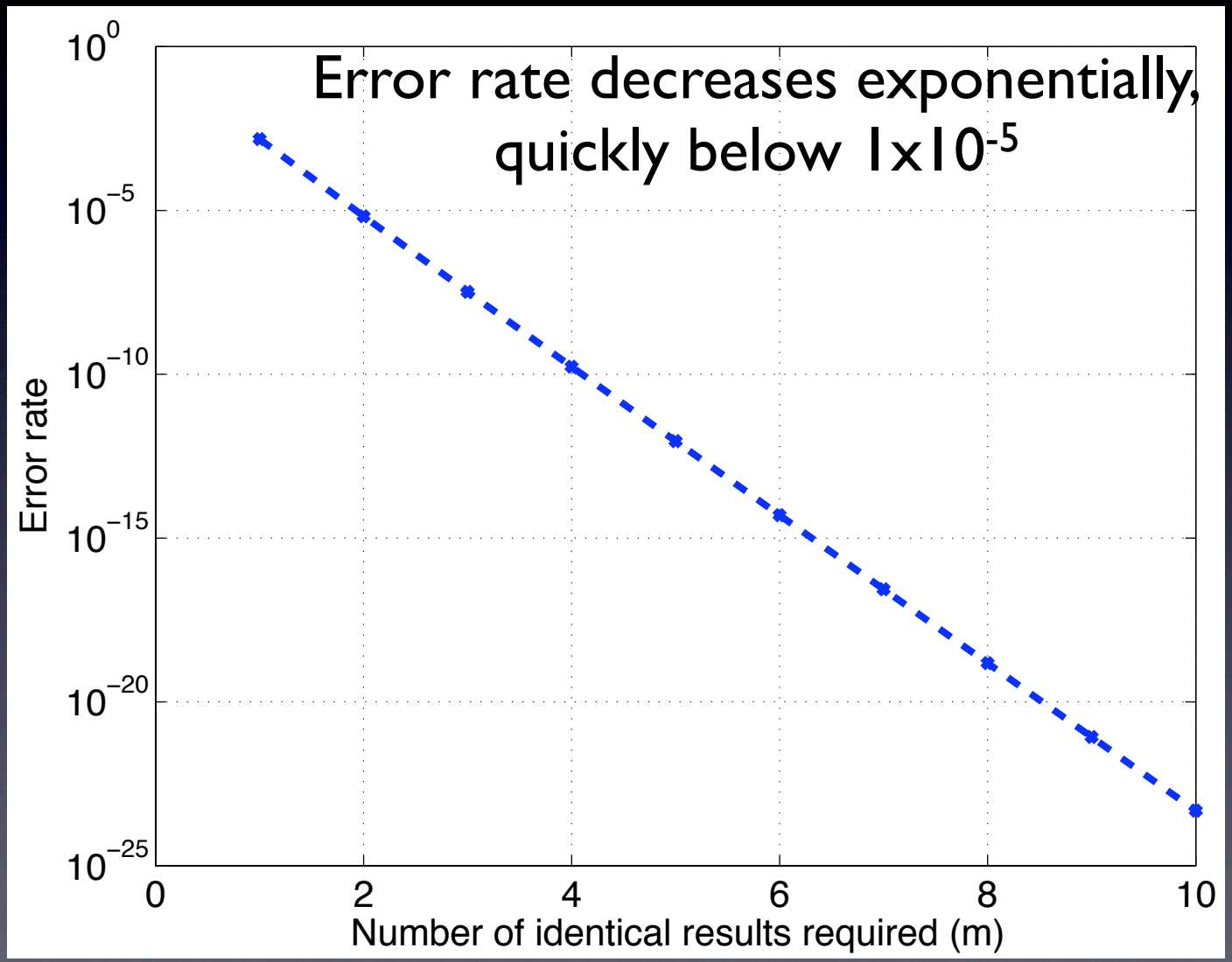
Spot-Checking with Blacklisting Revisited



Majority Voting Revisited



Majority Voting Revisited



Implications

- To get ϵ_{result} down to 2×10^{-4}
  - Spot-checking is a possibility: most benefit when *n* is [0,1000]
- To get $\epsilon_{result} \leq 2 \times 10^{-4}$
  - Use majority voting as ϵ_{result} exponentially decreases with *m*

Error Rate Stationarity

- A process is stationary if its statistical properties do not change over time
- Determine how stationary mean of host error rate (*s*) is over time
- Determine change in mean error rates over 96-hour periods for each host

Statistics for Host Error Rates over 96-hour periods

	Statistic		
Host Group	μ	σ	σ/μ
All erroneous	0.0034	0.018	3.48
Top 10% erroneous	0.0335	0.030	0.89
Bottom 90% erroneous	0.001	0.002	2.01

Only about 10% of the error rates were within 25% of the mean

Implications

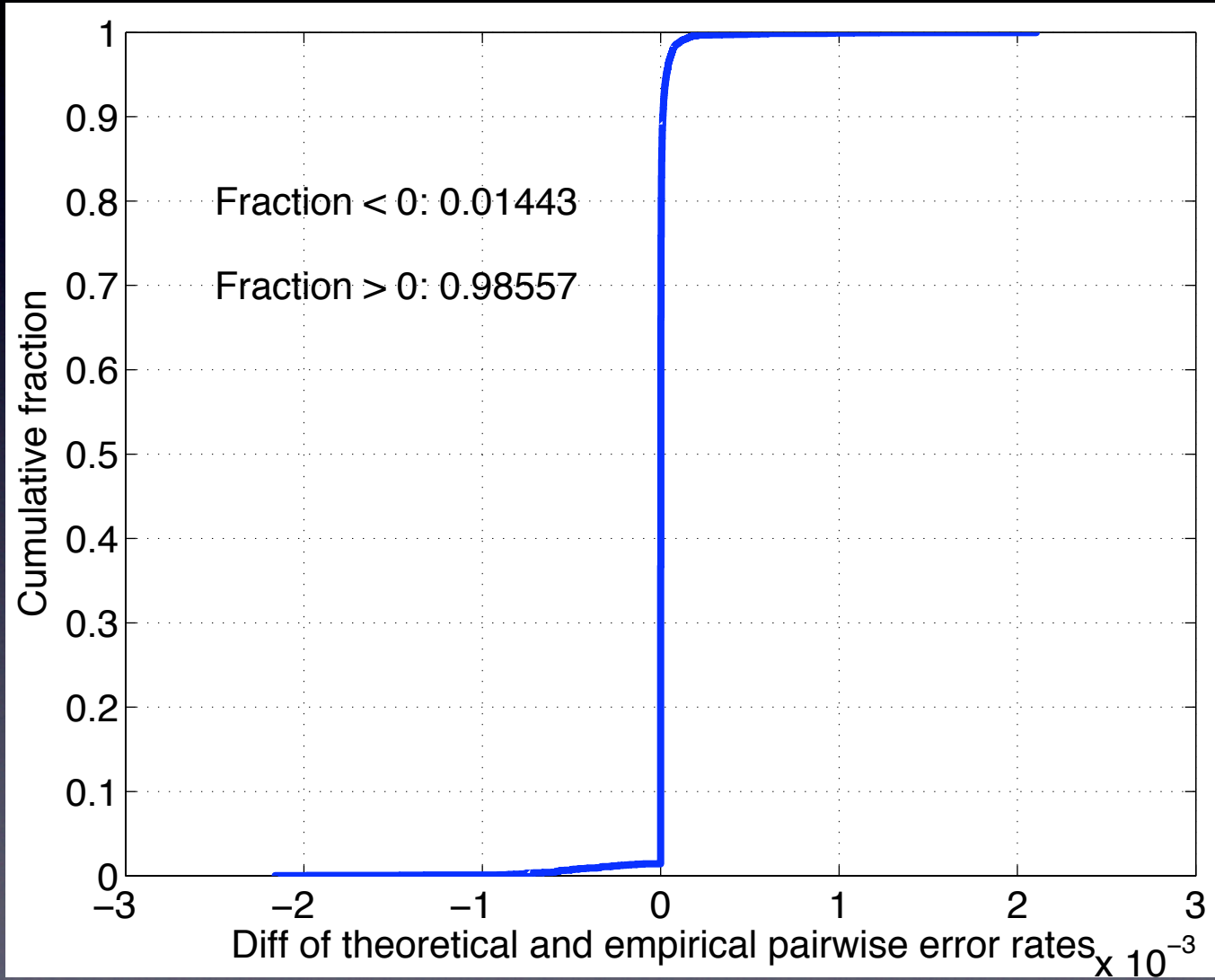
- Spot-checking and credibility-based systems may have limited effectiveness
  - Both depend on the consistency of error rates over time
    - Host with low error rate could build high credibility, and then triple its error rates

Correlation of Error Rates

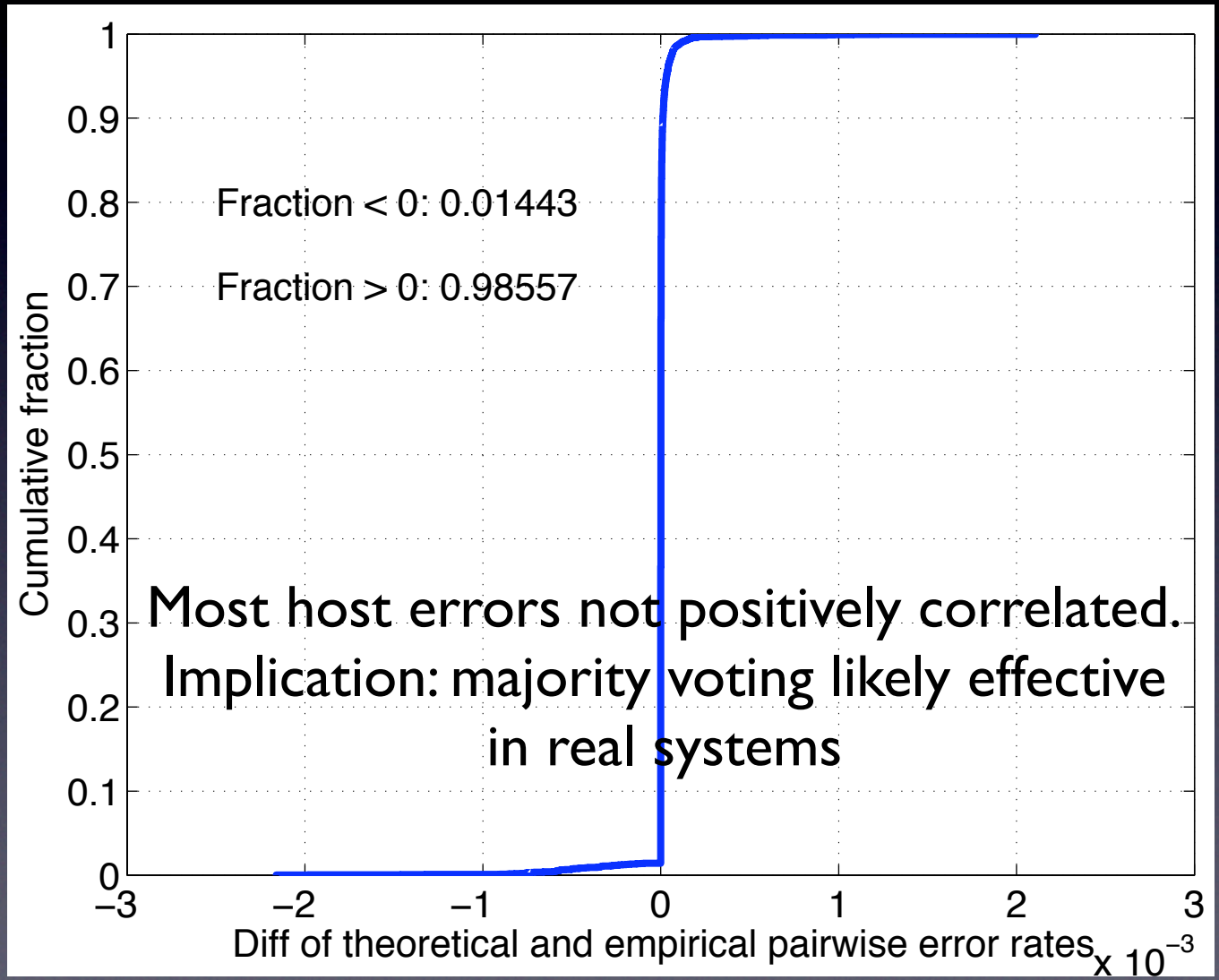
- Determine independence of error on one host with that on another
  - Independence: P(A and B) = P(A)*P(B)
- Determine empirical joint probability that any two hosts have error simultaneously } P(A and B)
- Computed "theoretical" probability of two hosts from error simultaneously } P(A)*P(B)
- If error rates are not positively correlated
  - P(A)*P(B) - P(A and B) ≥ 0

    theoretical - empirical ≥ 0

Pairwise Host Error Rates



Pairwise Host Error Rates



Summary of Characterization Results

- A significant fraction of hosts (about 35%) will commit at least a single error over time
- The mean error rate over all hosts (0.0022) is quite low
- A large fraction of errors (0.70) result from a small fraction of hosts (0.10)
- Error rates over time vary greatly (as much 3.48 times)
- Error rates between two hosts often seem uncorrelated (more than 0.986 of hosts do not have positively correlated errors)

Summary of Implications

- If one can afford redundancy or one needs an error rate to be less then 2×10^{-4}, then majority voting should be considered
- If one can afford an error rate greater then 2×10^{-4} and can make batches relatively long, spot-checking with blacklisting should be considered
- Fluctuations in error rates over time may limit the effectiveness of spot-checking and crediblility-based systems

Future Work

- Use of synthetic application
  - Important to have application regularity (I/O, computation)
  - Not that different from real desktop grid applications (cannot be obtrusive)
    - Compute-intensive, small-memory footprint, “light” periodic I/O for application-level checkpoints
- Characterize and run real desktop grid applications
  - Profile applications
  - Execute workunits representative from each profile

Thank you