The *Phoenix* Recovery System: Rebuilding from the ashes of an Internet catastrophe

Flavio Junqueira, Ranjita Bhagwan, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker

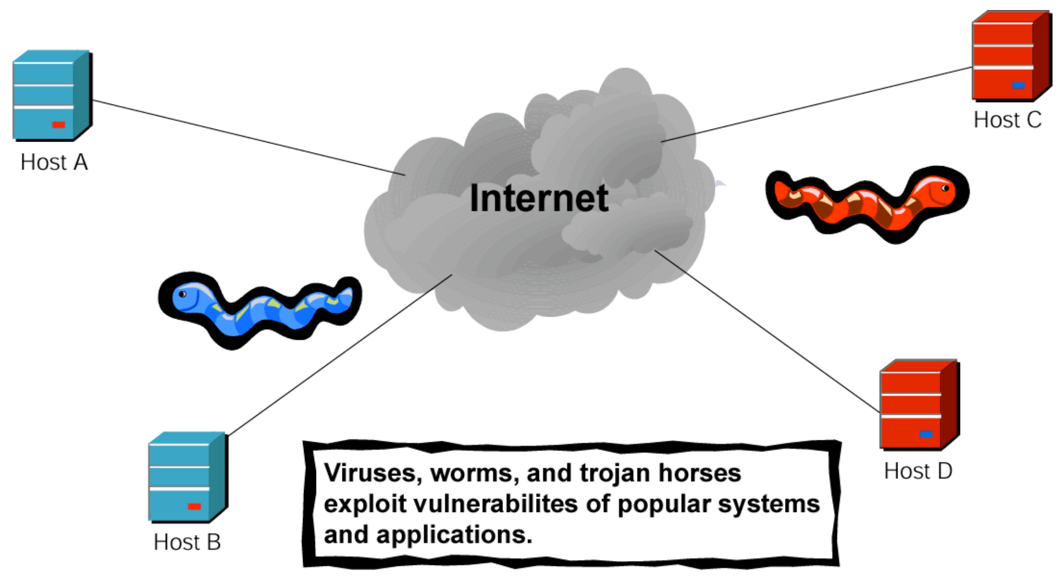
University of California, San Diego

Hot Topics in Operating Systems - HotOS'03

Motivation

- Operating systems and applications have vulnerabilities
- A large number of hosts may share the same vulnerability

- Some major outbreaks
  - Code Red: over 360,000 hosts
  - Sapphire: over 75,000 hosts



Viruses, worms, and trojan horses exploit vulnerabilites of popular systems and applications.

It is a matter of time until a major incident corrupting data on a large number of hosts happens

Our goal: build a system resilient to major Internet incidents

Introduction

- Possible approaches
  - Contain Internet pathogens: very challenging [Moore03]
  - Recover from catastrophes: replicate data
- Typical replication strategy
  - Assume independent host failures
  - Compute a threshold t on the number of failures
  - Replicate to this degree
- Shared vulnerabilities $\longrightarrow$ Dependent host failures
- Independent host failures is not a suitable assumption
- Threshold t on the number of host failures
  - From previous events, t can be large
  - Code Red worm infected over 360,000 hosts

What is a good replication strategy?

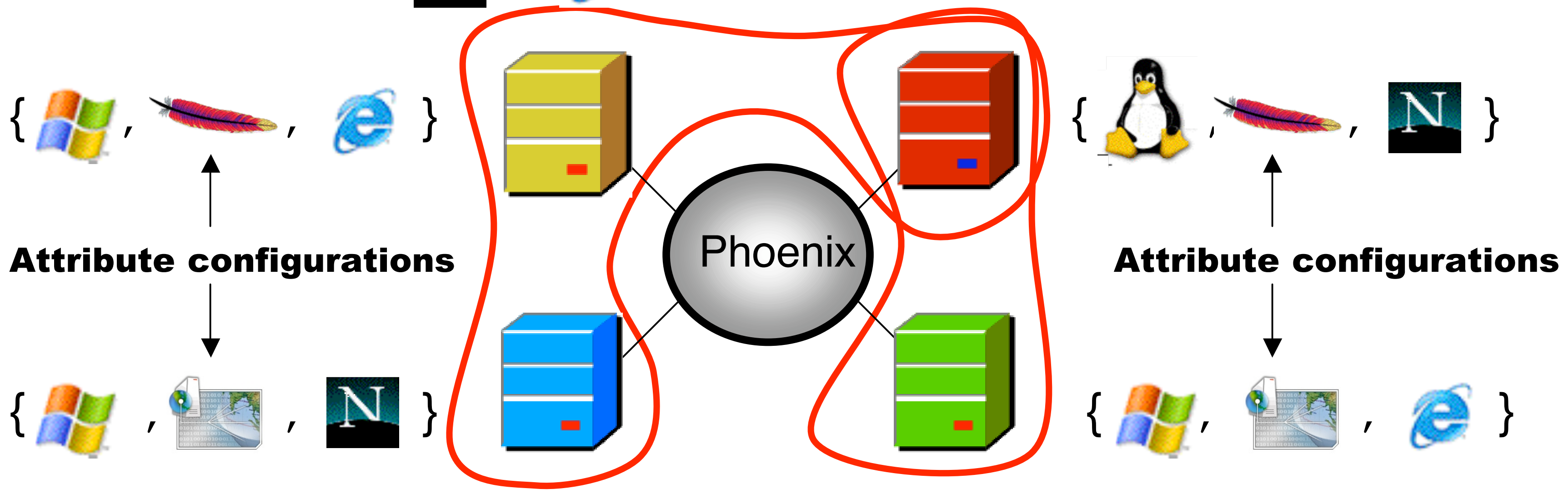
- Desirable properties
  - Enable recovery of data after an Internet catastrophe
  - Small replica sets
- Informed strategy for replica placement
  - Sets of hosts that fail independently
  - Hosts executing different sets of software systems

Our replication strategy

- Classes of software systems: *attributes*
  - E.g. Operating system
- Potentially vulnerable software systems: *attribute values*
  - E.g. Linux, Windows
- Replicate data on a set of hosts that have different values for each attribute: *cores*
- Tolerating the failure of k values
  - No permutation of k attribute values covers all the hosts in a core
  - Current assumption: k=1
    - At least two distinct values per attribute in a core
- Definitions
  - Attribute configuration: attribute values of a host
  - Diversity: distribution of attribute configurations

An example

- Attributes
  - Operating system:{ , }
  - Web server:{ , }
  - Web browser:{ , }
- Cores
  - Red and Green (orthogonal core)
  - Red, Yellow, and Blue



In this work…

- Feasibility of this approach
  - What is the impact of diversity on storage overhead and load?
- Simulations
  - Levels of diversity
  - Attribute sets
- Reminder
  - Storage overhead: size of the replica set (core)
  - Storage load: given a host h, number of cores h participates

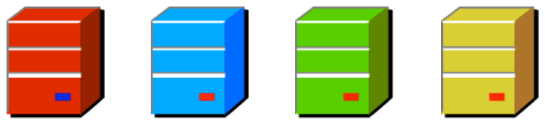
System model

- A set H of hosts
- A set A of attributes
- Every attribute has the same cardinality y
- A mapping M from hosts to attribute configurations
- Diversity
  - Determined by M
  - Often skewed in practice (93% Windows) [OneStat]

- Modeling diversity
  - Single parameter $f \in [0.5,1)$
  - A share f of the hosts has a share $(1-f)$ of the attribute configurations

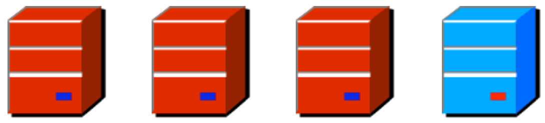
Attribute configurations: 🟨 🟥 🟦 🟩

Example 1:



$f = 0.5$

Example 2:



$f = 0.75$

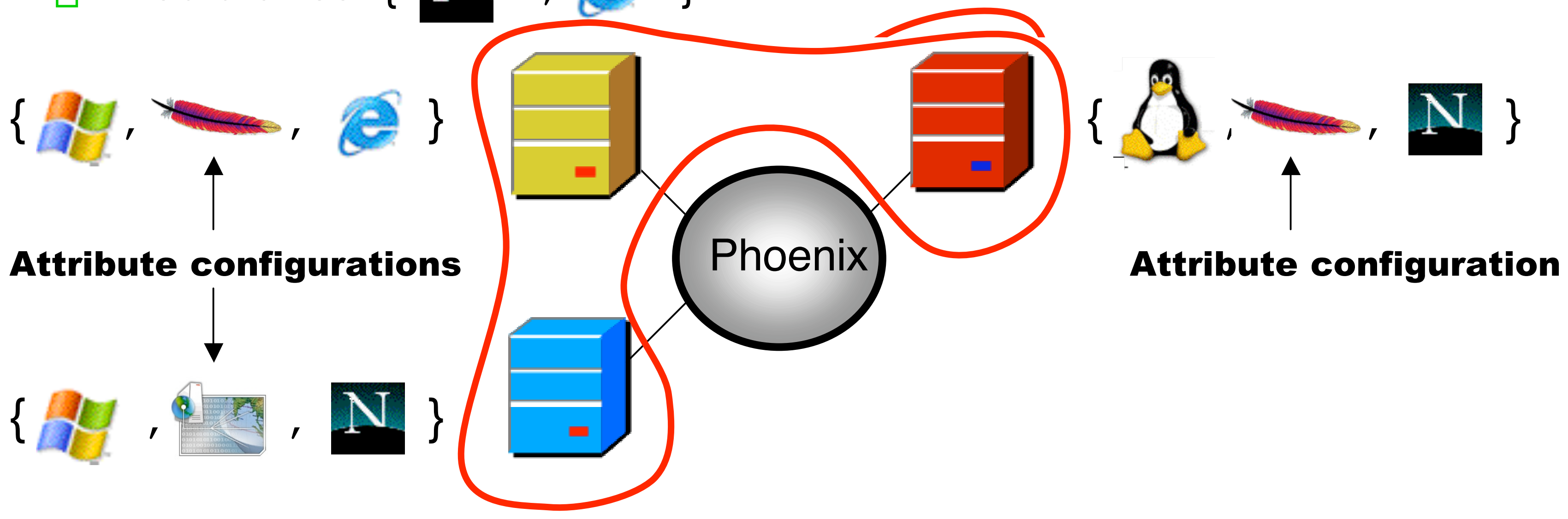
Choosing a core

- Decision problem is NP-Complete (Set cover)
- Finding a core for host h_i

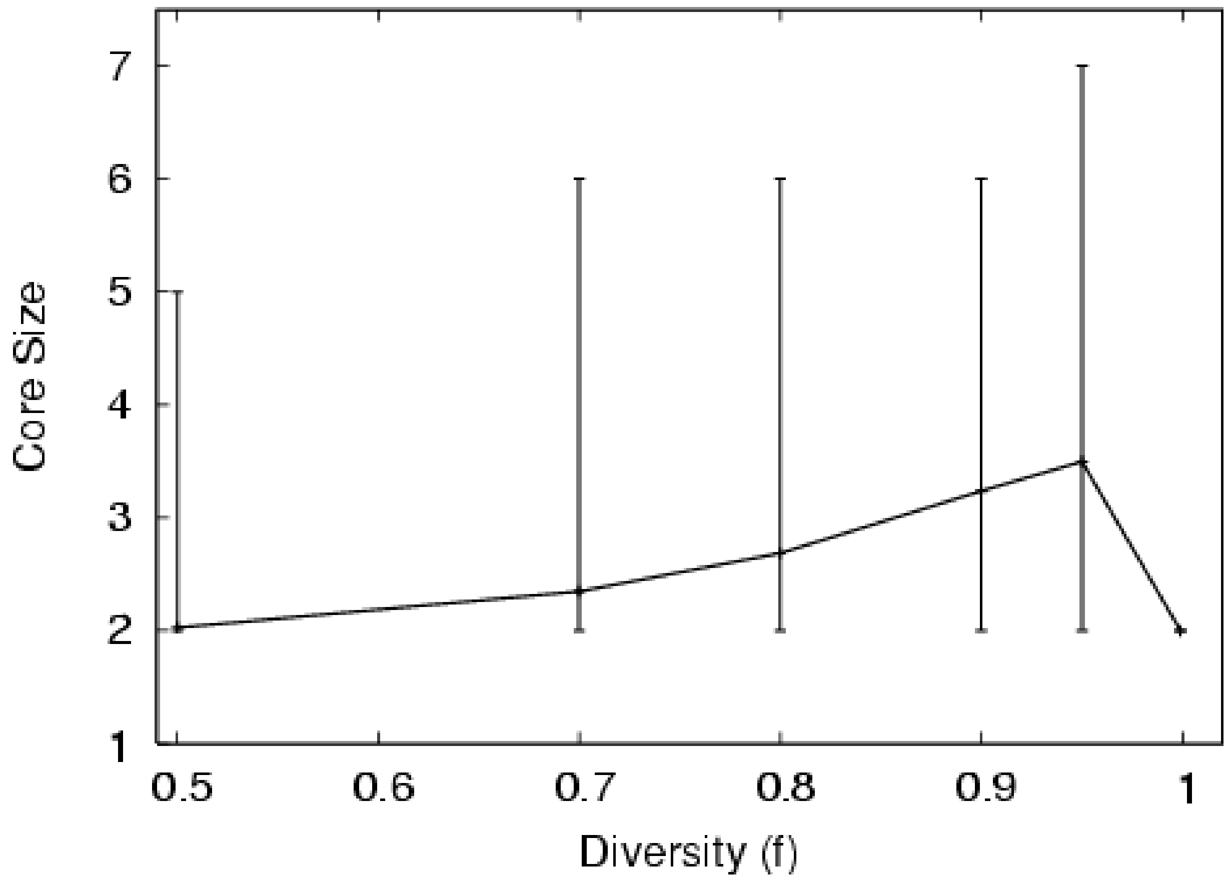
```
1. Make a list $L$ of hosts orthogonal to $h_i$
2. If $L$ is not empty
    1. Choose a host $h_j$ s.t  $h_j \in L$;
    2. Return $\{h_i, h_j\}$;
3. Else
    1. $R \leftarrow \{h_i\}$;
    2. Make a list $L'$ of hosts that have different attribute configurations;
    3. For each attribute $a$ in $A$, choose randomly a host $h_j$ in $L'$ s.t. $h_j$ has a
       different value for $a$;
    4. $R \leftarrow R \cup \{h_j\}$;
    5. Repeat  2 and 3 until $R$ covers all attributes or $L'$ is empty;
    6. Return $R$.
```

Back to the first example

- Attributes
  - Operating system:{ , }
  - Web server:{ , }
  - Web browser:{ , }
- Cores
  - Red and Green
  - Red, Yellow, and Blue



Core size for scenario 8/2

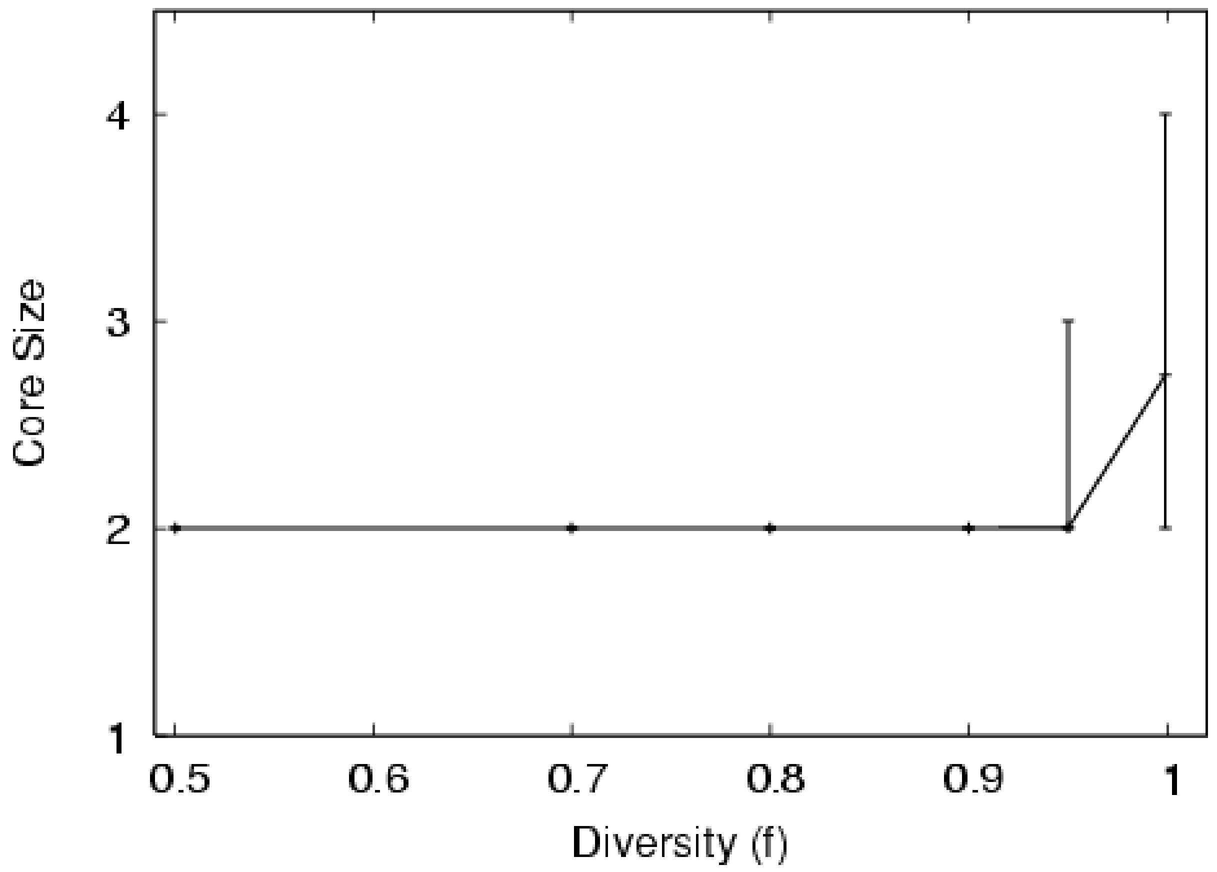


- 1,000 hosts
- 8 attributes
- 2 values per attribute

"Linux vs. Windows"

- Average core size is small even for highly skewed diversity

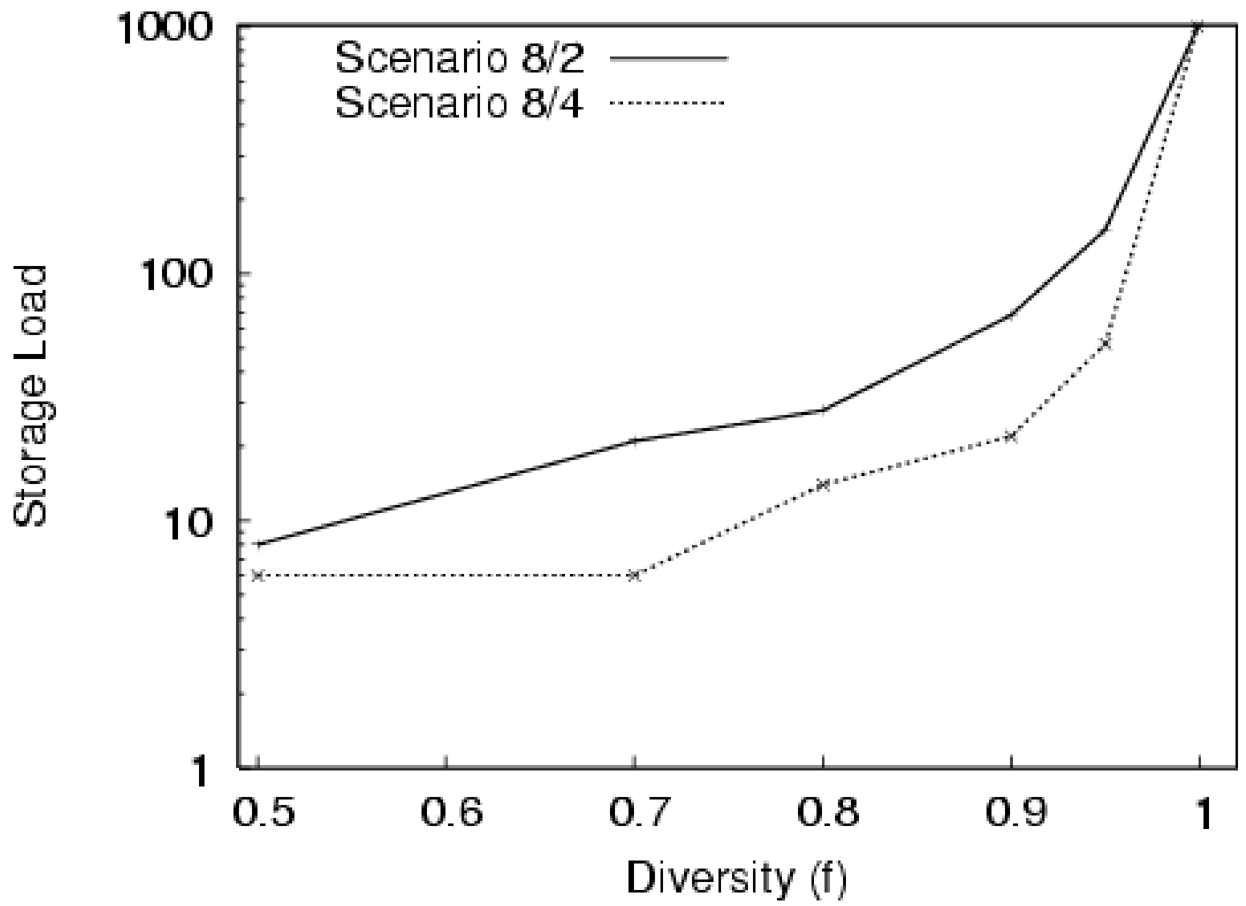
Core size for scenario 8/4



- 1,000 hosts
- 8 attributes
- 4 values per attribute

- More attribute values reduces core size variation

Storage load



- 1,000 hosts
- For highly skewed diversity, storage load can be high

System design issues

- Fully-distributed system
  - No single point of failure
  - Leverage research on P2P systems
- Announcing available configurations
  - DHT-based approach
- Encryption scheme to protect against data corruption
- Recovering from a catastrophe
  - Time to recover is not critical
  - Coping with a large number of requests
    - Threshold on the number of accepted requests
    - Exponential backoff

Conclusions

- Failures are not independent
- Computing a threshold is not practical
- Model of dependent failures based on shared vulnerabilities
- Storage overhead is small even for highly skewed diversity
- Storage load can be large
  - Has to be considered by the heuristic that finds cores
  - Increase average core size

Future work

- How do we determine the attributes?
  - Resilience depends on the attributes
  - Vulnerability databases
  - Dynamic attributes:new attributes and values
- How many attributes do we need?
  - The number of attributes impact on storage overhead
- What is a good level of granularity for the attributes?
  - E.g. {Windows} vs. {Win_95, Win_98, Win_2000, Win_XP}
- Other challenges
  - Heuristics for finding cores: storage overhead and storage load
  - Efficacy
    - How do we assess the efficacy of a prototype?
    - Major Internet incidents are not so frequent

Possible attributes

- Classes of exposed from the ICAT vulnerability database (http://icat.nist.gov) - 05/13/2003

Exposed component	**2003**	**2002**	**2001**	**2000**
Operating system	54 (15%)	212 (16%)	248 (16%)	152 (15%)
Network protocol stack	2 (1%)	18 (1%)	8 (1%)	14 (1%)
Non-server application	113 (31%)	266 (20%)	309 (21%)	194 (20%)
Server application	177 (48%)	772 (59%)	886 (59%)	555 (56%)
Hardware	17 (5%)	54 (4%)	43 (3%)	15 (2%)
Communication protocol	10 (3%)	2 (0%)	9 (1%)	31 (3%)
Encryption module	0 (0%)	0 (0%)	6 (0%)	23 (2%)
Other	9 (2%)	27 (2%)	5 (0%)	24 (2%)

Introduction

- Backup systems
  - Local techniques: tapes and CDs
  - Commercial remote backup
  - Cooperative remote backup
- Cooperative remote backup
  - A host h relinquishes a fraction x of its disk
  - x/k per user, if h serves k other hosts
- Threshold model
  - Worst-case scenario
  - For dependent host failures
    - Threshold possibly very large
    - k possibly very large and x/k very small
  - Infeasible for such scenarios

Introduction

- Software
- Worms and viruses exploit these vulnerabilities
- Several hosts share the same vulnerability
- E.g. Code Red worm (360,000); Saphire worm (75,000)
- None of these caused any major damage on computers connected to the Internet

… *but It is a matter of time until a major Internet incident occurs*

Replication strategy

- Replicate on hosts that fail independently
- Assumption
  - Hosts executing the same program are likely to fail dependently
  - E.g. Hosts executing the same OS version
- Rationale
  - Shared vulnerabilities
- Derived strategy
  - Replicate on hosts that run distinct sets of programs

A simple model of diversity

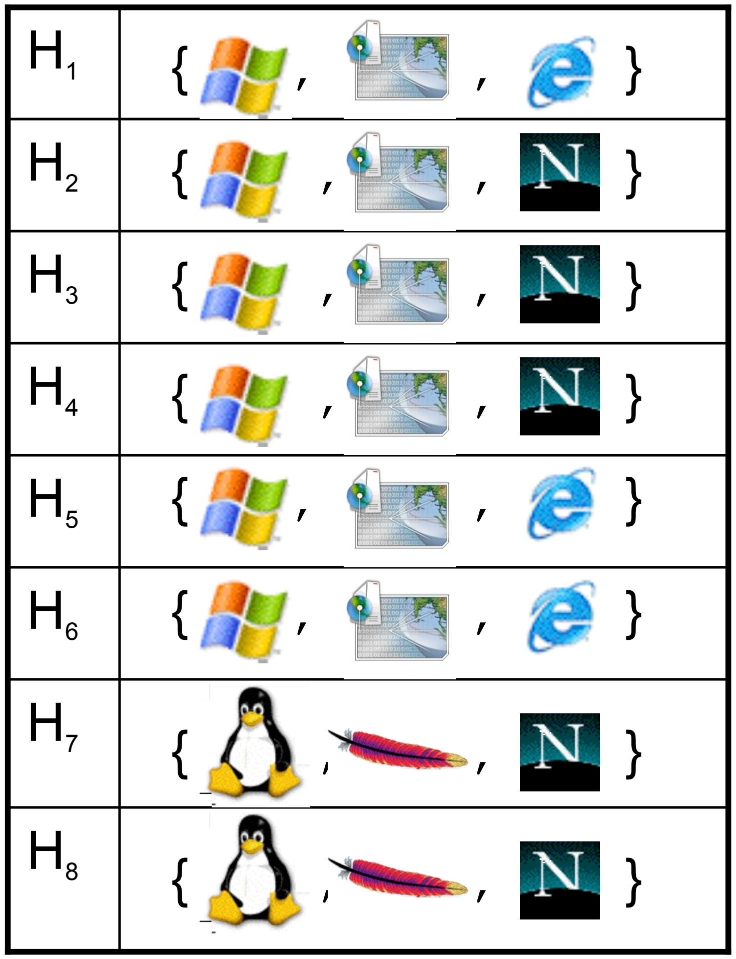
- Rationale:
  - distribution of attribute configurations is often skewed
  - Assess the tradeoffs as diversity becomes more skewed
- f□ $[0.5,1)$: single parameter of the model
  - A share f of the hosts has a share $(1\text{-}f)$ of the attribute configurations
- Given a value of f, find the value of □ that satisfies the following:

$$\frac{1}{y^{□}} \geq (1 □ f) > \frac{1}{y^{□+1}}$$

- Generating a mapping M
  - Fix the value of □ attributes
  - Choose values randomly for the other $|A|$ - □ attributes

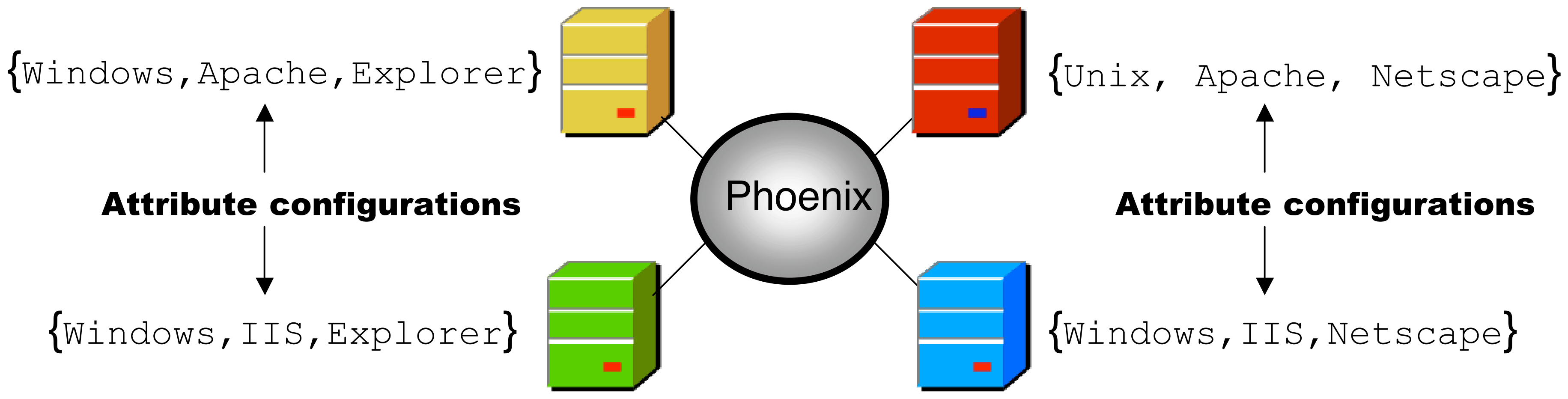
Another example

- Attributes
  - Operating system:{ , }
  - Web server:{ , }
  - Web browser:{ , }
- Operating system and Web browser: most skewed attributes
- 75% of the hosts (6) have 25% of the attribute configurations (2)
  - f = 0.75
  - y = 2
  - □ = 2


H_1
H_2
H_3
H_4
H_5
H_6
H_7
H_8

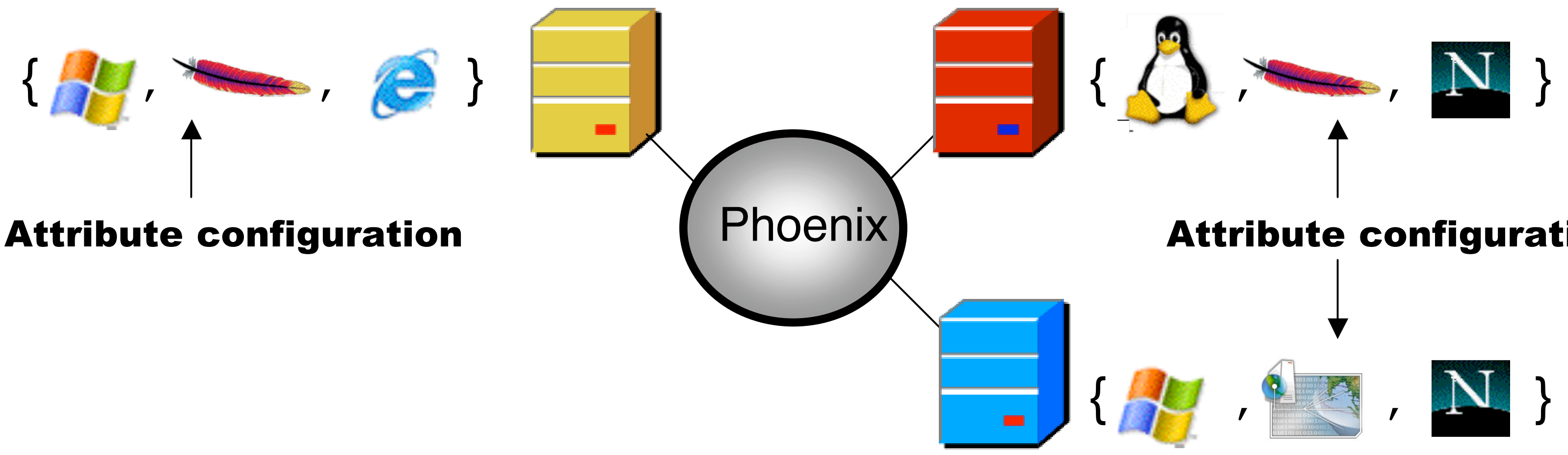
An example

- Attributes
  - Operating system:{Windows,Unix}
  - Web server:{Apache, IIS}
  - Web browser:{Netscape, Explorer}
- Cores
  - Red and Green
  - Red, Orange, and Blue



Back to the first example

- Attributes
  - Operating system:{ , }
  - Web server:{ , }
  - Web browser:{ , }
- Cores
  - Red and Green
  - Red, Yellow, and Blue


{ , , }
Attribute configuration
Phoenix
{ , , }
Attribute configurations
{ , , }