Availability: the game of 9’s

Ratul Mahajan
CSE 590is
Winter ‘03

Availability

 Availability (əˈvɑːləˈbiːləti) [noun]:
• the quality or state of being available

 Available (əˈvɑːləˈbli) [adjective]:
• present or ready for immediate use
• present in such chemical or physical form as to be usable

 Measured in number of 9’s

 Not binary
• Degraded service

 Internet Services = network + sites
Outline

- How many 9’s?
  - Statistics

- Why not more 9’s?
  - Causes

- How can we get more 9’s?
  - Fixes

- But, how many 9’s do I want?

How many 9’s in the Internet?

- Hard to quantify
  - How to make representative measurements?
  - How to treat degraded service (application-dependent)?

- Between 2 and 3 is the conventional wisdom
- Hasn’t moved in the last 5-7 years

- Intra-domain is more reliable than inter-domain
- No conclusive data on whether the “edge” or the “core” is more reliable
How many 9’s in the sites?

- Between 4 and 5

- Internet becomes the bottleneck
  - Has important consequences as we try to improve user experience

The unsurprising facts

- Most failures are short-lived
  - Median was 100 seconds for the network [Dahlin]

- But most of the downtime comes from long failures
  - More than half came from 100,000+ second failures [Dahlin]

- Consequences for techniques to increase availability
### 2000 "Downtime Costs" (per Hour)

<table>
<thead>
<tr>
<th>Service</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brokerage operations</td>
<td>$6,450,000</td>
</tr>
<tr>
<td>Credit card authorization</td>
<td>$2,600,000</td>
</tr>
<tr>
<td>Ebay (1 outage 22 hours)</td>
<td>$225,000</td>
</tr>
<tr>
<td>Amazon.com</td>
<td>$180,000</td>
</tr>
<tr>
<td>Package shipping services</td>
<td>$150,000</td>
</tr>
<tr>
<td>Home shopping channel</td>
<td>$113,000</td>
</tr>
<tr>
<td>Catalog sales center</td>
<td>$90,000</td>
</tr>
<tr>
<td>Airline reservation center</td>
<td>$89,000</td>
</tr>
<tr>
<td>Cellular service activation</td>
<td>$41,000</td>
</tr>
<tr>
<td>On-line network fees</td>
<td>$25,000</td>
</tr>
<tr>
<td>ATM service fees</td>
<td>$14,000</td>
</tr>
</tbody>
</table>

Source: "A simple way to estimate the cost of downtime" (Patterson)

Estimated Average Cost of hour of downtime =

Employee Costs per Hour * Fraction Employees Affected by Outage +
Average Revenue per Hour * Fraction Revenue Affected by Outage

### Causes

- Hardware
- Software
- Operators
- Overload
- Attacks
- Environment
- Natural disasters
- ......
Causes [RoC]

- Percentage of failures

  ![Sources of Failure]

  - Internet site hardware/software is less reliable
  - Overload/attacks don’t contribute much to site failures

Causes [Labovitz]

- Intra-domain for a regional ISP

<table>
<thead>
<tr>
<th>Outage Category</th>
<th>Number of Occurrences</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance</td>
<td>272</td>
<td>10.2</td>
</tr>
<tr>
<td>Power Outage</td>
<td>273</td>
<td>10.0</td>
</tr>
<tr>
<td>Fiber Cut/Circuit/Carrier Problem</td>
<td>201</td>
<td>7.2</td>
</tr>
<tr>
<td>Unreachable</td>
<td>225</td>
<td>8.5</td>
</tr>
<tr>
<td>Hardware problem</td>
<td>134</td>
<td>5.0</td>
</tr>
<tr>
<td>Interface down</td>
<td>105</td>
<td>4.2</td>
</tr>
<tr>
<td>Routing Problems</td>
<td>104</td>
<td>6.1</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>86</td>
<td>5.9</td>
</tr>
<tr>
<td>Unknown/Undetermined/No problem</td>
<td>32</td>
<td>5.0</td>
</tr>
<tr>
<td>Congestion/Sagging</td>
<td>65</td>
<td>4.6</td>
</tr>
<tr>
<td>Malicious Attack</td>
<td>26</td>
<td>1.5</td>
</tr>
<tr>
<td>Software problem</td>
<td>23</td>
<td>1.0</td>
</tr>
</tbody>
</table>

- This is not so operator heavy; hardware and software are not big factors either.
**Trends [RoC]**

- Human error is largest single failure source
  - HP HA labs: human error is #1 cause of failures (2001)
  - Oracle: half of DB failures due to human error (1999)
  - Gray/Tandem: 42% of failures from human administrator errors (1986)
  - Murphy/Gent study of VAX systems (1993):

![Diagram of causes of system crashes]

**Improving availability**

- Availability = $\frac{MTTF}{MTTR+MTTF}$
  - Decrease MTTR
  - Increase MTTF
    - Of components -- hardware, software, operators
    - Of service -- masking component failures from "users"
- 10x decrease in MTTR = 10x increase in MTTF
  - MTTR >> MTTF [Fox]
    - MTTF is hard to verify - time, environmental factors
    - Indirect costs of a high MTTR are higher
      - 1 day per year is worse than 10 mins per day
  - Cannot neglect MTTF
    - MTTF >> typical "session"
  - Is it an XOR scenario?
MTTR [RoC]

1. Time to detect
   ◆ Better interfaces, active measurement
2. Time to pinpoint
   ◆ Error checking at edges, log past experience
3. Time to find a fix
   ◆ Past experience, easy undo
4. Time to fix
   ◆ Find the culprit workload, interfaces that enable testing
   ◆ All of the above is hugely complicated when multiple administrative entities are involved
   ◆ Routing protocols try to do this automatically

Component MTTF

◆ Hardware (and software?) is already > 5 9’s
   • For both sites and network

◆ The big remaining piece is the operator (slips, mistakes)
   • Configuration checking -- consistency and system constraints
   • Holistic configuration -- configure the network, not the router
   • Better interfaces -- guard against “fat fingers”
   • High level languages
   • Revolutionary operator fault-tolerant design?
   • “Automation irony”

◆ What about BGP?
   • Plagued by a high MTTR
Service MTTF

• Two basic techniques
  • Redundancy (no replicated state)
    § Stand-by or partitioning
  • Replication

• Issues
  • Failure detection
  • Failing over
  • May not be 100% effective -- degraded service
  • Consistency in case of replication

• Not always possible
  • Too expensive
  • Consistency and performance are important

Redundancy

• Examples
  • Hardware and software
  • Network paths
    § ISPs, physical links, conduits

• Fail-over may not be straightforward in the Internet
  • Tension between hierarchy, efficiency, scaling, and policy
  • RON targeted this aspect
    § Value depends on fraction of “middle of the network” failures
    § But it could overcome almost all failures with ~40 nodes
    § Can also deal with degraded service
  • Roughly factor of 2 improvement in Dahlin’s data
Replication

◆ A few tough decisions
  • What to replicate?
    § The whole site or a subset (resource consumption)
    § Data or code+data (untrusted code)
  • When to replicate and refresh?
    § When first requested, let it expire (cache)
    § When first requested, keep it fresh (push-updates)
    § Even before the first request (pre-fetching)
  • Where to replicate?
    § Close to users (need more replicas), another place

Benefits of replication

◆ Speed-up = (Original unavailability)/(New unavailability)

<table>
<thead>
<tr>
<th>Replication technique</th>
<th>Caching</th>
<th>Pushed Cache</th>
<th>Pre-fetching</th>
<th>Active pre-fetching</th>
<th>Site replication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed-up</td>
<td>1-1.5</td>
<td>1-1.5</td>
<td>2-6 (4-12)</td>
<td>2-6 (100%)</td>
<td>2</td>
</tr>
</tbody>
</table>

◆ Replication combined with re-routing can provide 2 more 9’s
How many 9’s are good enough?

- The answer lies in economic balance
  - As you try to get more 9’s
    - Cost rises exponentially
    - Benefits decrease exponentially


- But we all share the same network
  - Pay for quality of service
  - Pay your ISP to reduce “near-server” failures
  - Pay Akamai