Dependability Scalability: A Technical and Socio-Technical Challenge

Jean-Claude Laprie

ITI Workshop - Urbana-Champaign, 3 December 2004
High dependability for safety-critical or money-critical systems

- Avionics, railway signalling, nuclear control, etc.
- Transaction processing, back-end servers, etc.

**Scalability of dependability?**

Continuous complexity growth

(web-based applications, networked embedded systems; fixed or mobile)
NetCraft — Web Sites Uptime statistics

Top 50 most requested sites

24 November 2004

Time to reboot (h)

Requests

www.georgewbush.com
www.google.com
wwwdir.telia.com
java.versalite.com
microsoft.com
www.johnkerry.com
www.princeofwales.gov.uk
www.microsoft.com
www.johnkerry.com
wwwdir.telia.com
java.versalite.com
microsoft.com
www.google.com
www.georgewbush.com

avg
max
High dependability for safety-critical or money-critical systems

Avionics, railway signalling, nuclear control, etc.

Transaction processing, back-end servers, etc.

Scalability of dependability?

Continuous complexity growth
(web-based applications, networked embedded systems; fixed or mobile)

In addition to Fault Avoidance (fault prevention and fault removal),

Generalization of Fault Acceptance
[fault tolerance and fault forecasting for Resilience and Survivability]

Occurrences

Non malicious causes 71%

Malicious causes 29%

Occurrence impact

Risk perception

Non malicious causes 29%

Malicious causes 71%

3 year trends
stable
increase
decrease
Scalability of dependability?

Continuous complexity growth
(web-based applications, networked embedded systems; fixed or mobile)

In addition to Fault Avoidance (fault prevention and fault removal),
Generalization of Fault Acceptance
[fault tolerance and fault forecasting for Resilience and Survivability]

Physical failures
Residual software defects
Intrusions, incl. denials of service
Vulnerabilities [some unavoidable for usability]

Human interaction errors [Administration, configuration, maintenance]
Fault tolerance

Physical failures
- Rejuvenation
  - Checkpointing
    - Intermittent faults
- Error detection
  - Wrapping

Residual software defects
- Fail-fast & reconfigure

Intrusions
- Information dispersal
- Platform diversity
- Recovery after intrusion

Human interaction errors
- Automation (‘autonomic systems’)
  - Automation paradox
From N. Bowen, ‘Software availability in an on demand world’
Fault tolerance assessment

Coverage demonstration
- by analysis (incl. formal) and by experiments (representative fault injection)
- dynamic, for evolutions

Fault acceptance policies and model development
interconnection

integration

unavoidability of faults

funnel effect

dependence

Decreasing natural robustness

Vanishing substitutes for informatics

Ill-mastered complexity

Performance

Interconnection