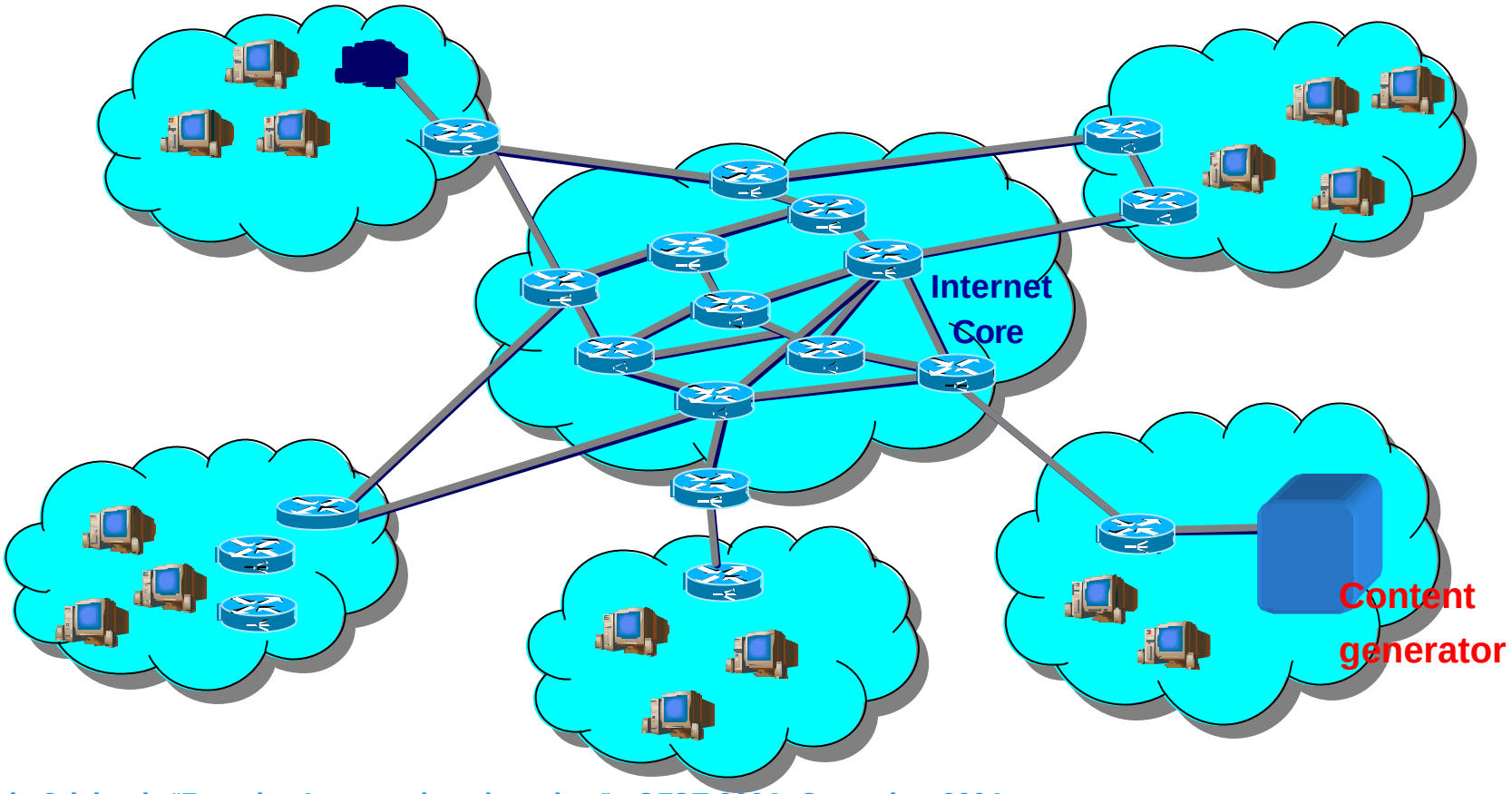
QEST 2004, University of Twente

Emerging Internet-based services: New frontiers for performance models and applications

Michele Colajanni
University of Modena, Italy

Internet-based services

- **Remote interaction with some content provider (repository/generator)**

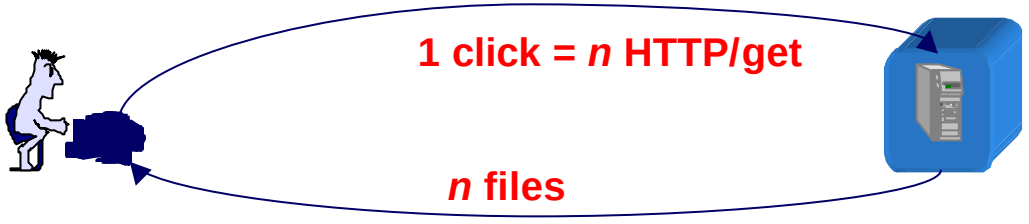


Internet-based services
(driven forces for performance studies)

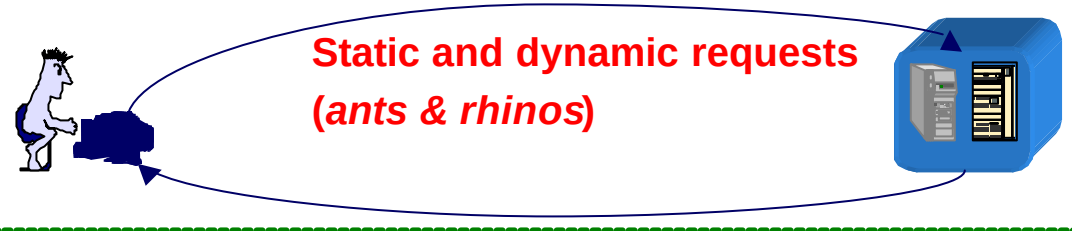
- **Remote interaction with some content provider (repository/generator)**
  - Dynamically generated contents
  - Multimedia services

- **QoS-based services**
- **Adaptation**
  - **System adaptation**
  - **Content adaptation**
- ***Security***

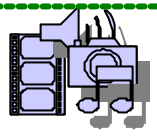
15 years of Web history in one slide



User's clicks may be assumed independent, not so the *n* HTTP/get requests to the server

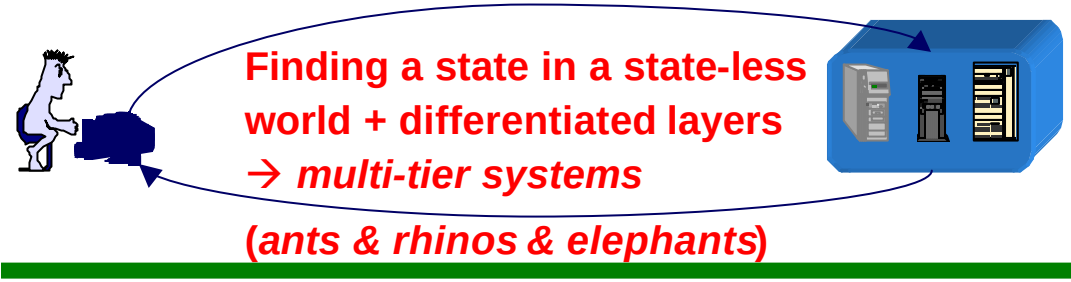


User's clicks become different: Service times of requests for static and dynamic files differ for one-two orders of magnitude



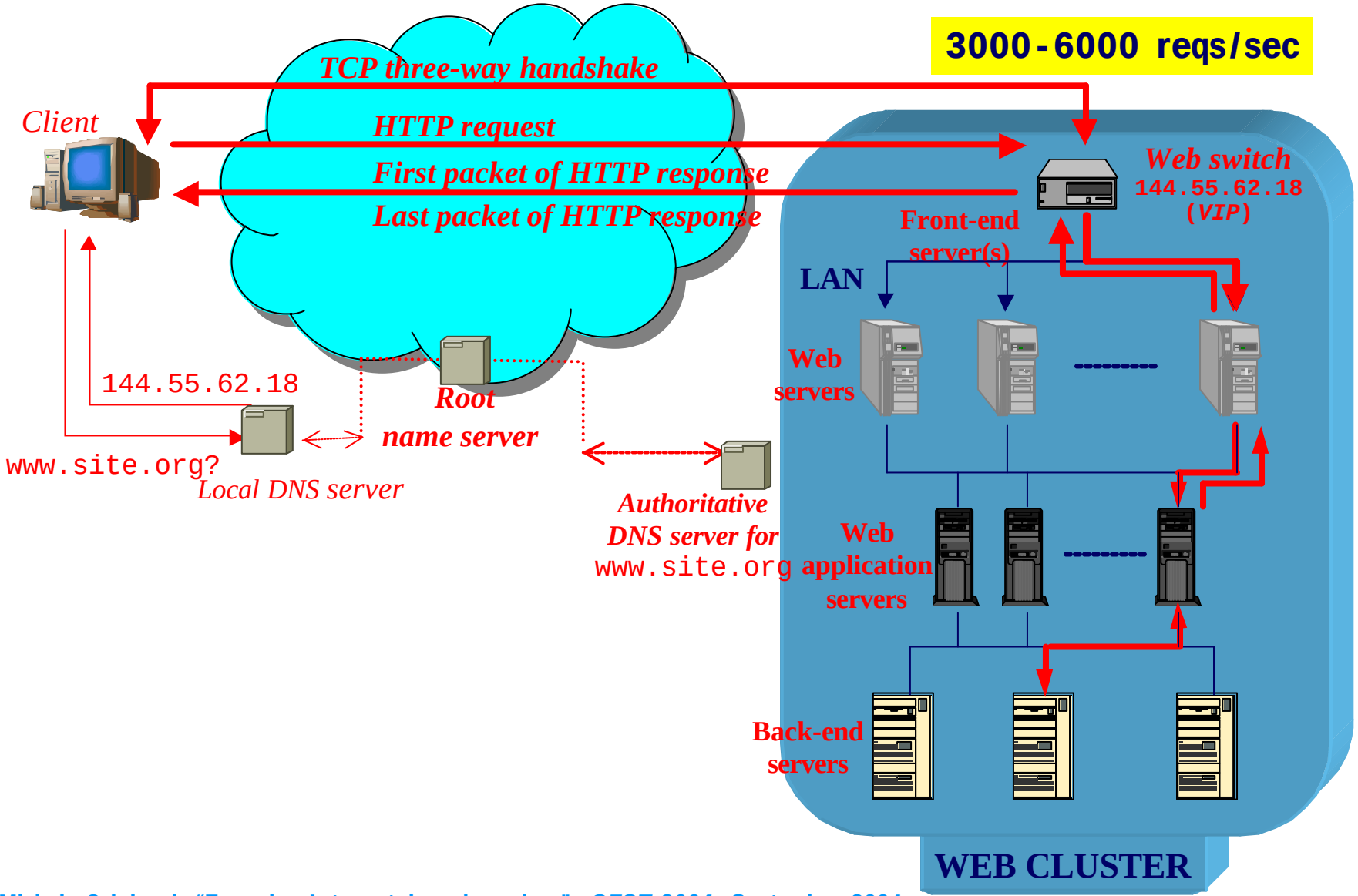
The advent of multimedia content as a Web resource
(*it was and it is more a network problem than a server problem*)

Big impact on service times (both server side and network side)



User's clicks become badly different: Requests may differ for two-three orders of magnitude, have quite diffent impact on system resources (memory, processes, disk, etc.)

Basic server infrastructure: *Web cluster*

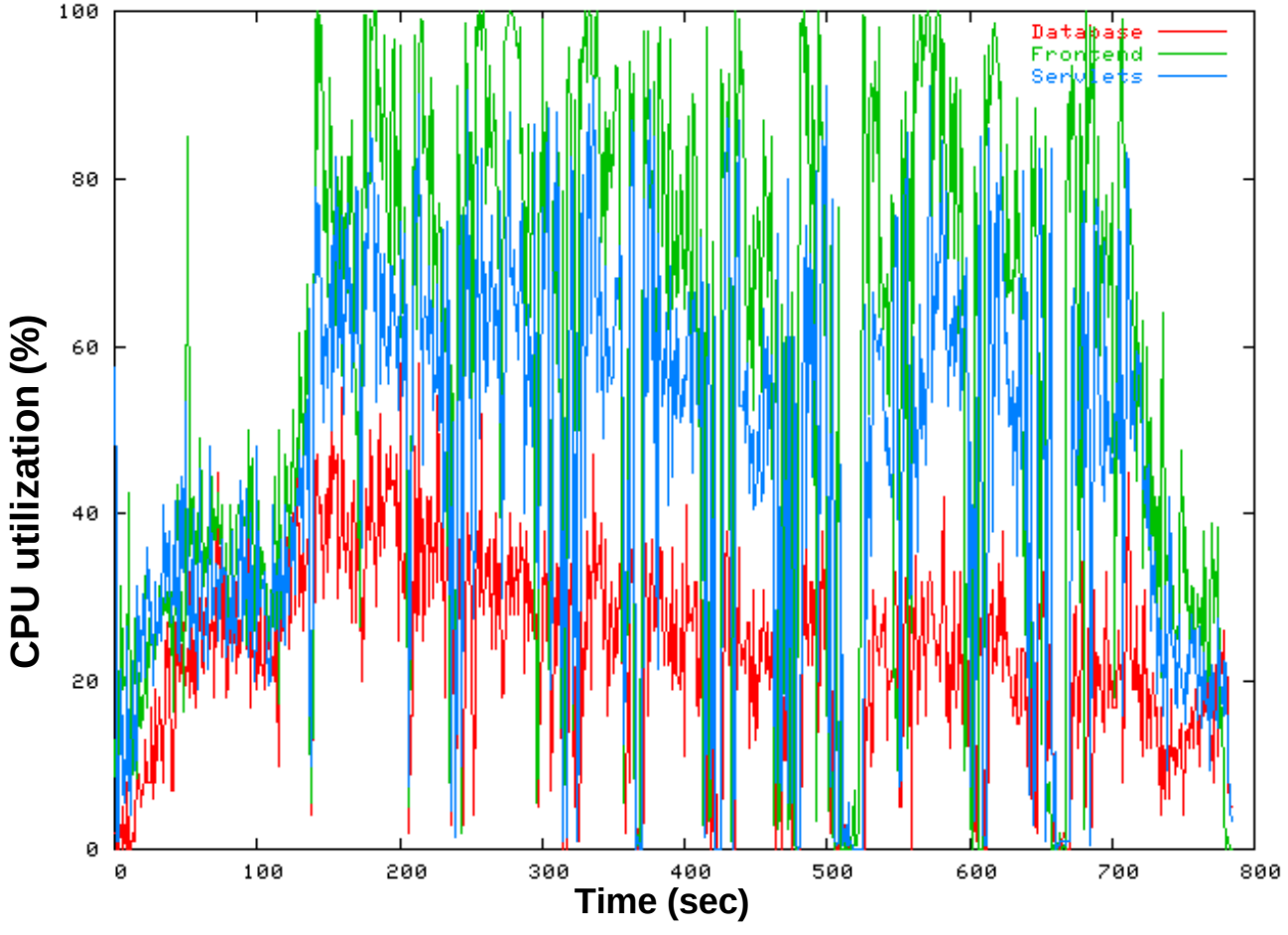


Self-similarity

- **Will Leland, Murad Taqqu, Walter Willinger, and Daniel Wilson, IEEE/ACM Transactions on Networking, Vol. 2, No. 1, Feb. 1994**
  - Identified presence of self-similarity property in aggregate Ethernet traffic
  - Defined methodology for testing for the presence of self-similarity

- **Several papers since then have identified self-similarity in other types of traffic (Internet, Web, multimedia)**

Scenario 1 (*ants-oriented*)

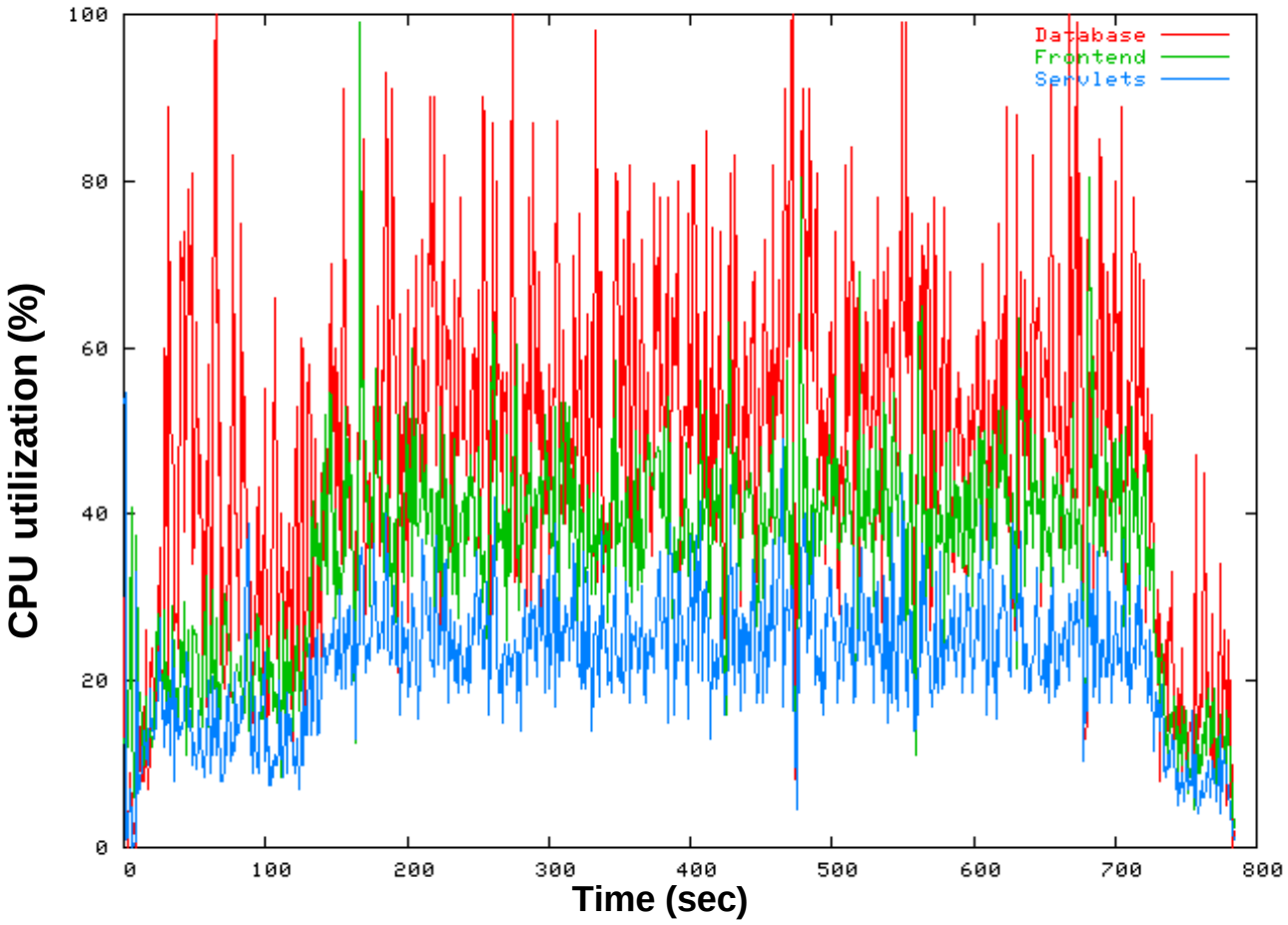


Web server

Servlets server

Database server

Scenario 2 (*elephant-oriented*)



Web server

Servlets server

Database server

System potential and *network* limits ("first mile" problem)

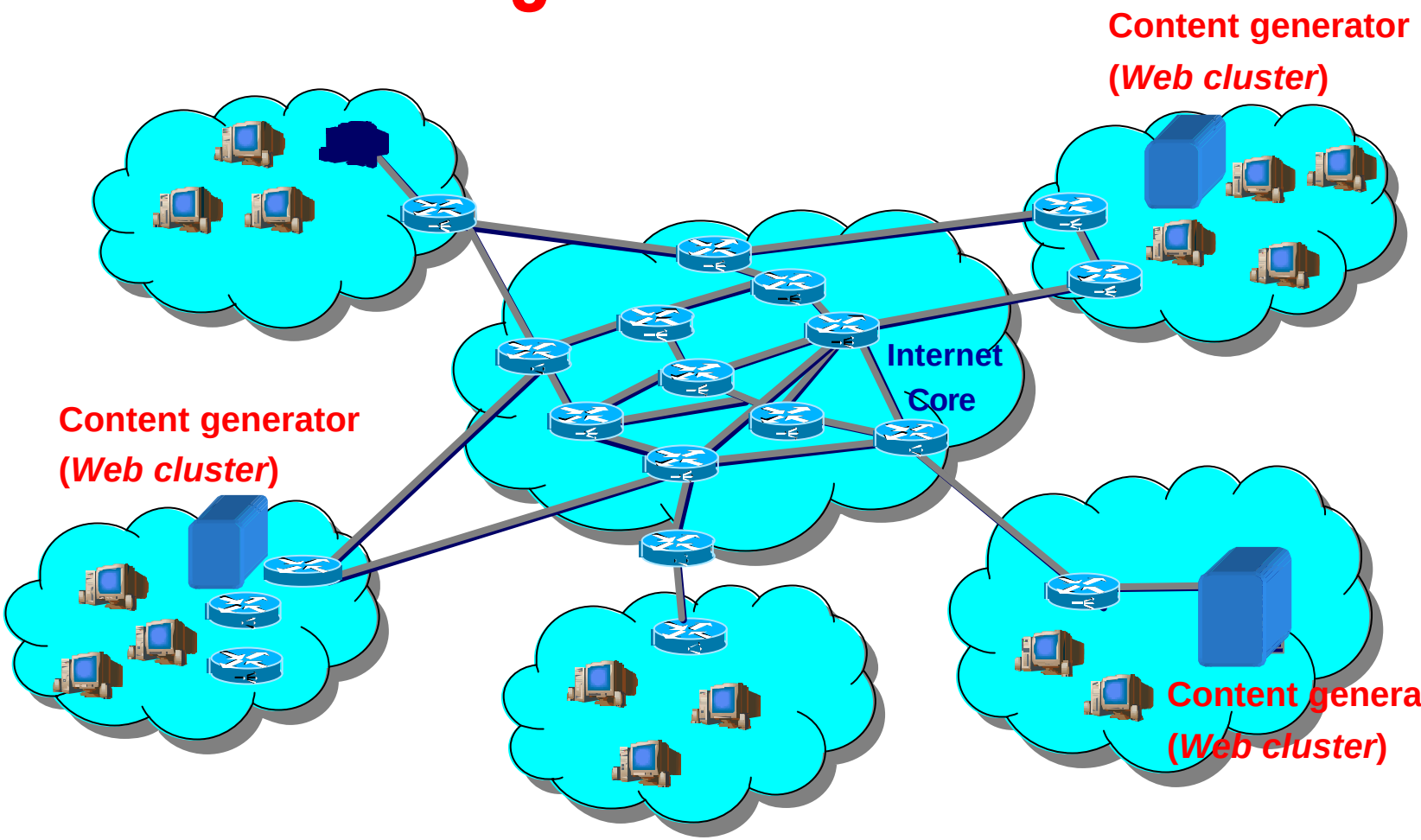
- **Web cluster throughput:**

  5 Mbps → 85 Mbps → W

- **Network throughputs:**
  - **T1** - **T2**: Large company to ISP → 3.1/6.3 Mbps
  - **T3** - **OC1**: ISP to Internet infrastructure → 44.7-51.8 Mpbs
  - **OC3**: Large company backbone → 155.5 Mbps
  - **OC12** - **OC256**: Internet backbones → 0.62-13.2 Gbps

→ *Caching* and *replication* over a geographical area

Multiple content repositories and generators



E-delivered services are a reality

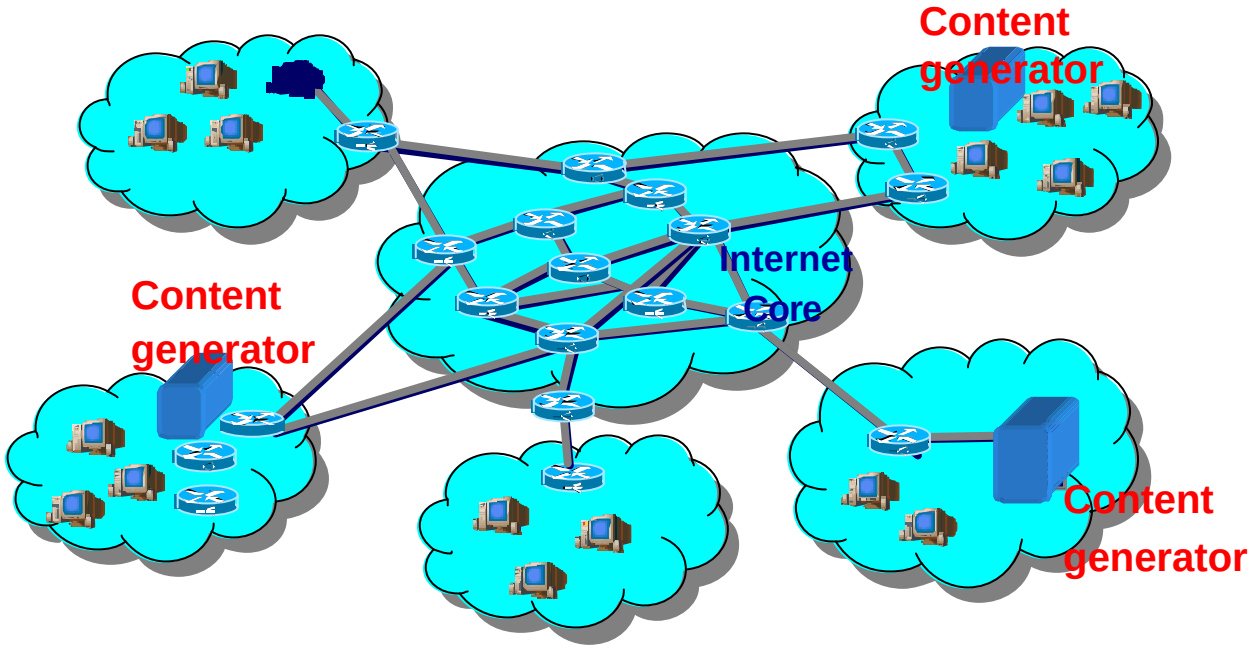
- **Sidney 2000 Olympics Web site was able to deliver 1.2M hits/minute**
- **International companies normally use geographically distributed information systems that are interconnected through their VPNs**
- **Sites servicing 100M requests/day are not unusual**

There are many complex Internet-based services that are not "emerging" anymore

A rhetorical question

Is there any space for performance modelers out there?

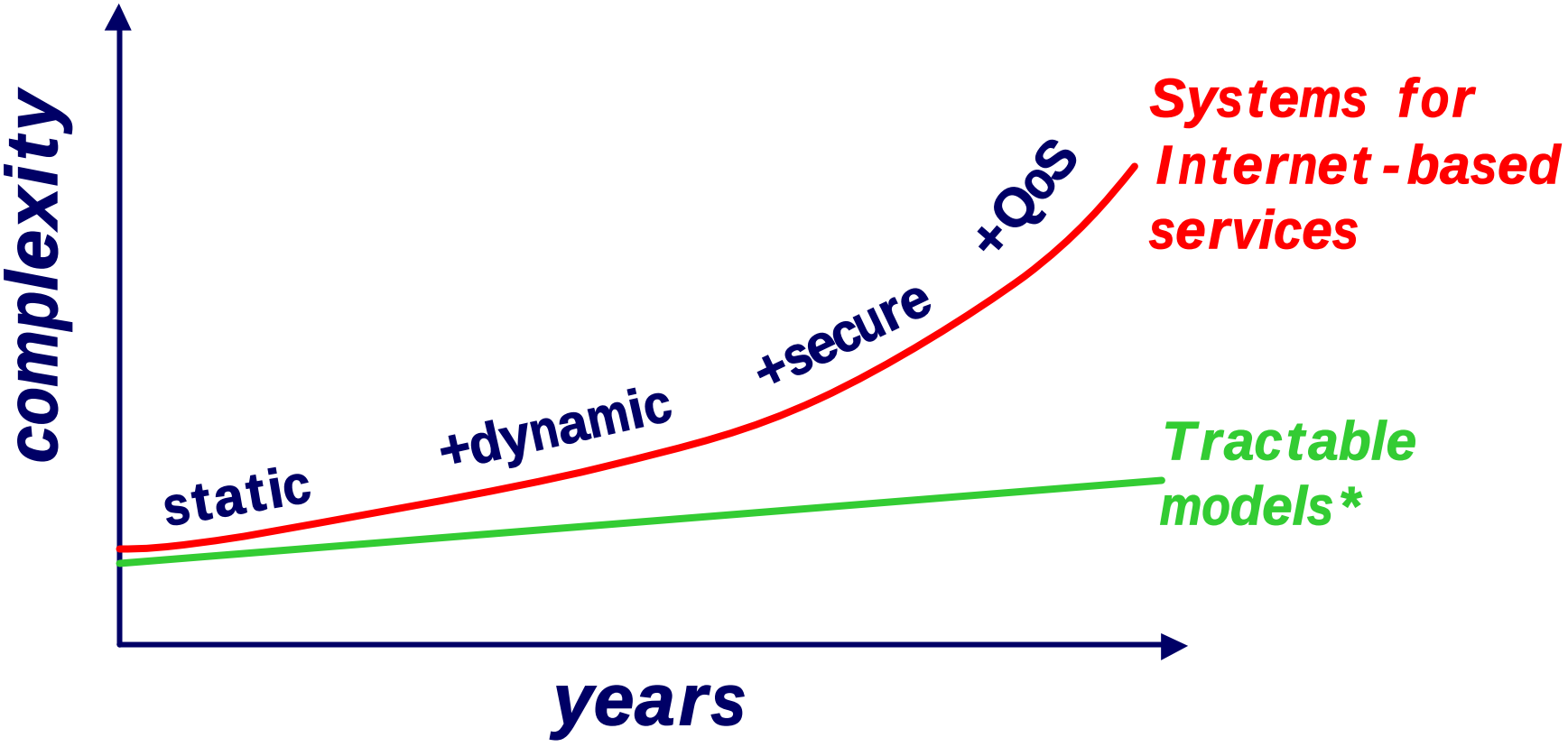
YES. A lot of needs!



- Efficient dispatching and load balancing algorithms
- Distributed mechanisms for monitoring the system components
- Algorithms and models for taking a large number of off-line and on-the-fly decisions
- Capacity planning and testing

BUT...

An increasing GAP (?)



* = Models that we like to treat

Expectation list: “The system must …

- … work”
- … work”
- … work most of the times”
-
-
- … work always with acceptable performance”
-
- … work always with good performance”
- … work always with optimal performance”

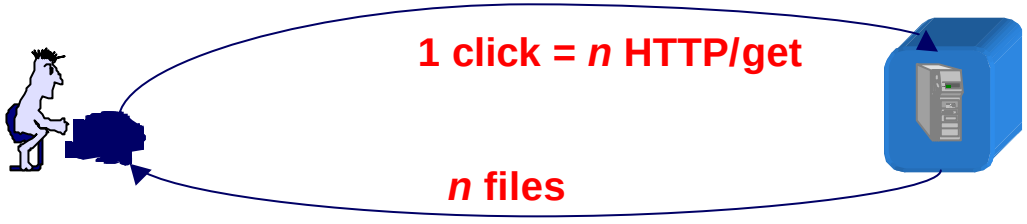
[Inspired by Ignazio Silone, “Fontamara”]

Some motivations

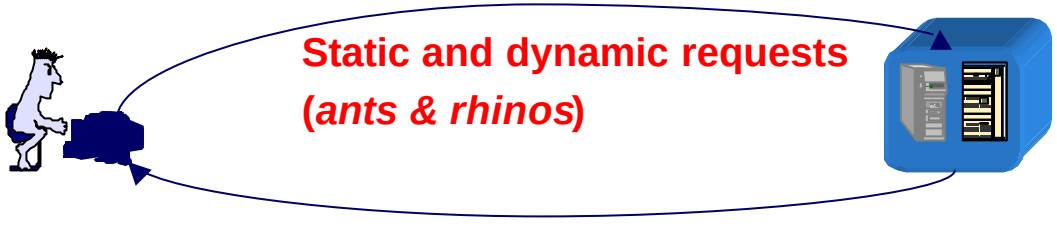
- Technology is abundant
- Time-to-market is short
- Defects are “socially” acceptable

- The systems supporting Internet-based services are quite complex
- Most solutions require time, optimizations require a lot of time, but the time is short
- We control a tiny fraction of the system

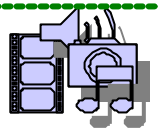
Web history and future in one slide



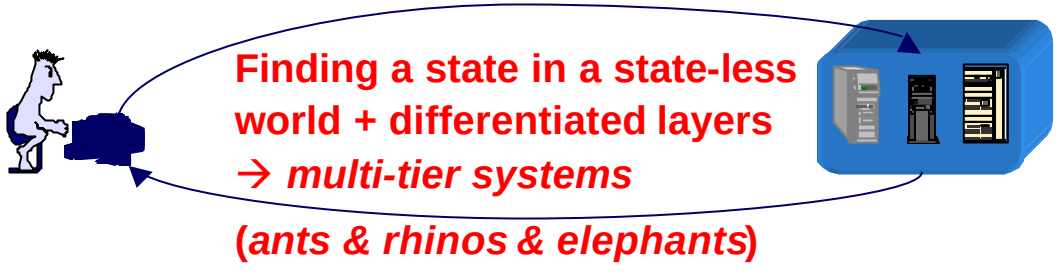
User's clicks may be assumed independent, not so the *n* HTTP/get requests to the server



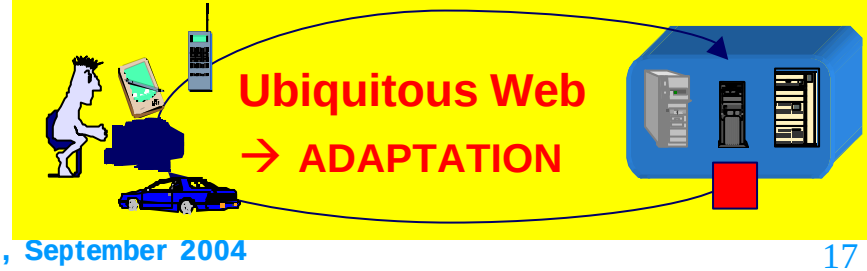
User's clicks become different: Service times of requests for static and dynamic files differ for one-two orders of magnitude



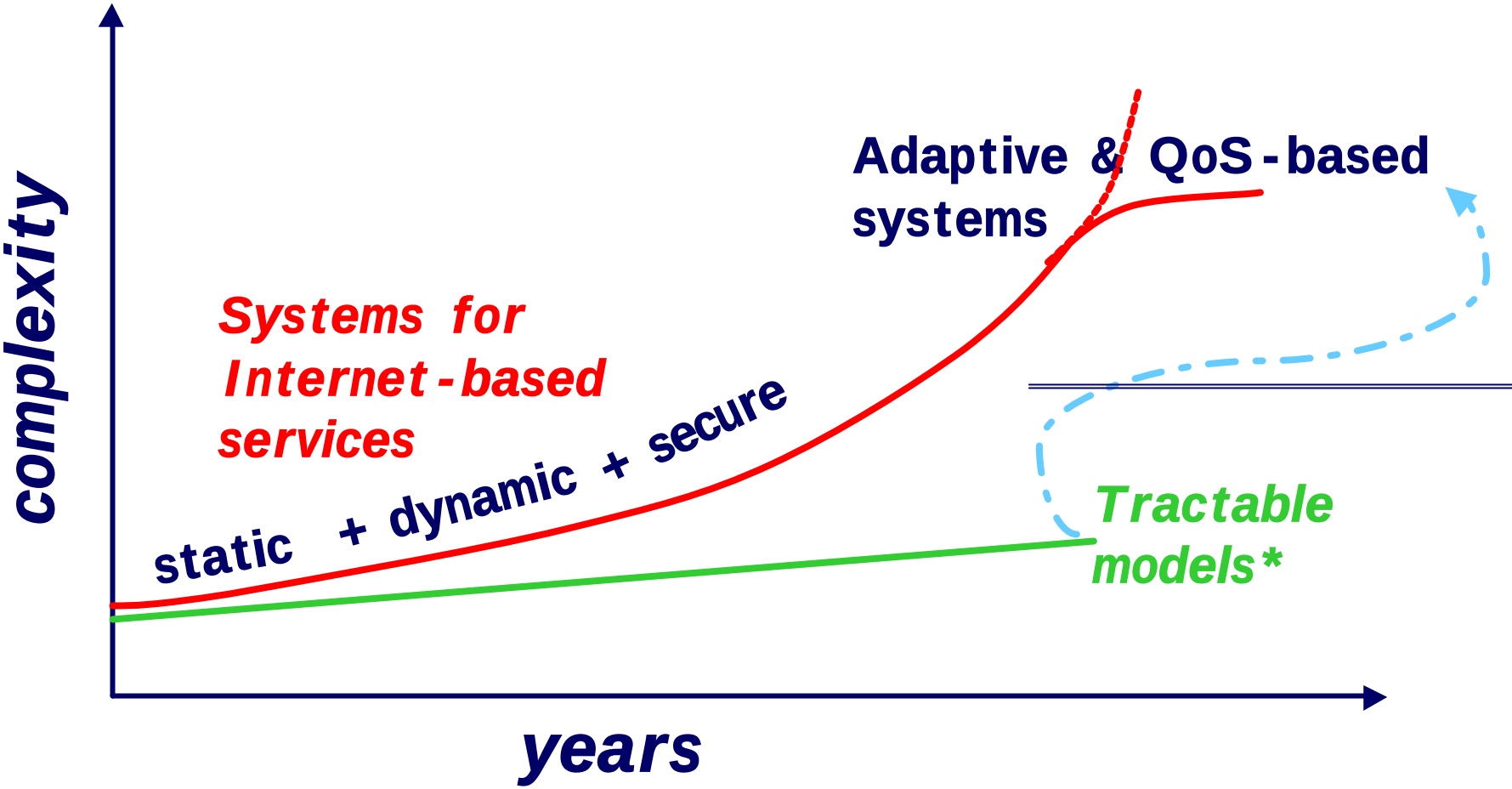
The advent of multimedia content as a Web resource
(*it was and it is more a network problem than a server problem*)



User's clicks become badly different: Requests may differ for two-three orders of magnitude, have quite diffent impact on system resources (memory, processes, disk, etc.)



An optimistic view of the GAP



* = Models that we need to treat to close the gap

Example: Future *e*-service provider

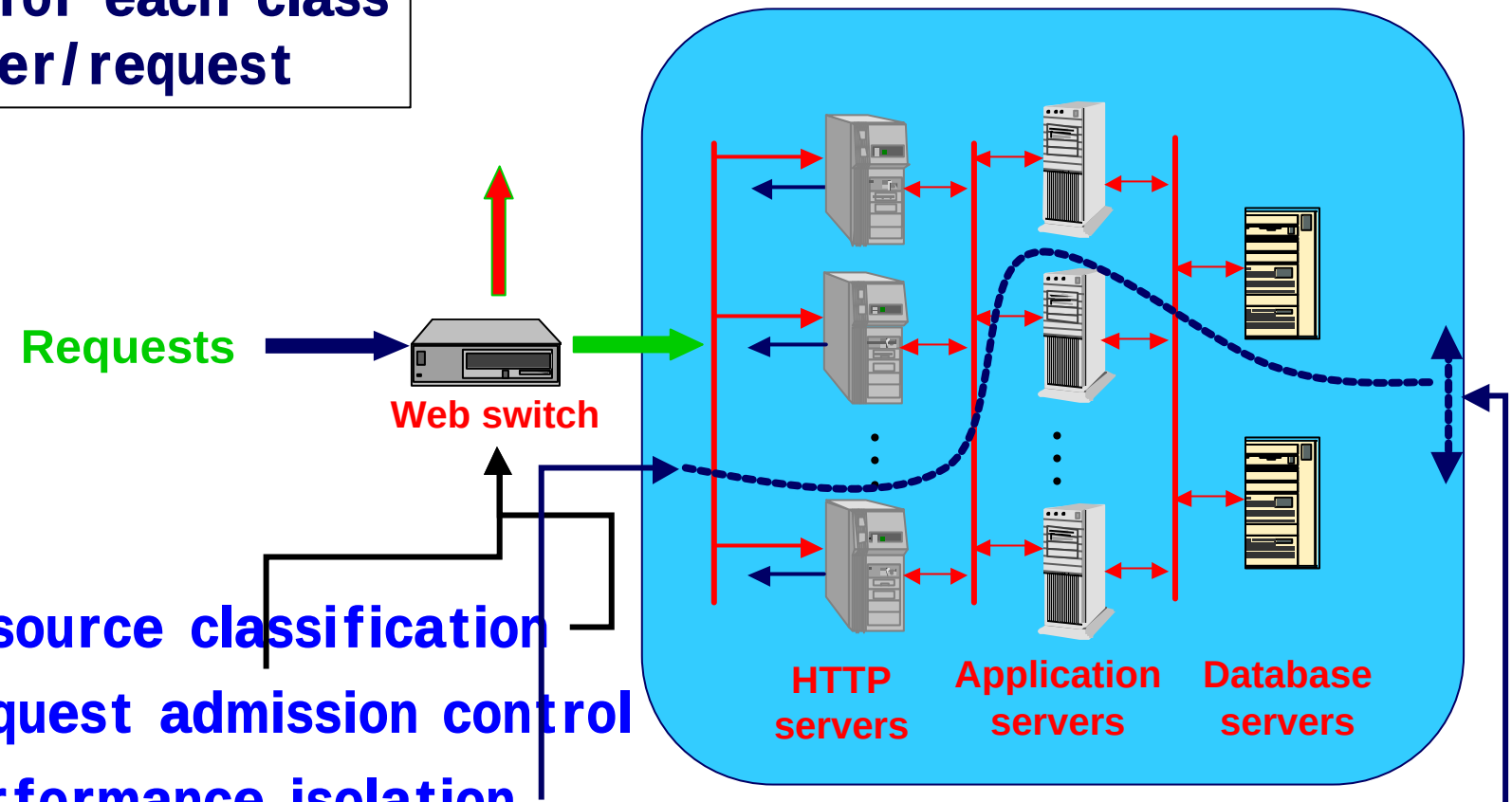
- **A provider that services thousands of customers ranging from large companies to individuals**
- **Offered services:**
  - bandwidth, storage, computational power with different Service Level Agreements (standard, personalized, pay-per-use, flat, etc.)
  - should allow to accommodate all types of customers, while customers' demands vary continuously during the day in an unpredictable way
- **These SLAs impose on the system a large set of performance goals and other objectives including reliability, availability, business constraints, real-time guarantees for interactive services**

Personal view

An approach different from
"overprovisioning+RoundRobin"
is necessary to design and build
Internet-based services
supporting QoS+Adaptation

A Web cluster supporting QoS-based services

SLA for each class of user/request



- **Resource classification**
- **Request admission control**
- **Performance isolation**
- **High resource utilization**

Quality of Internet-based services

High performance systems
(best effort services)

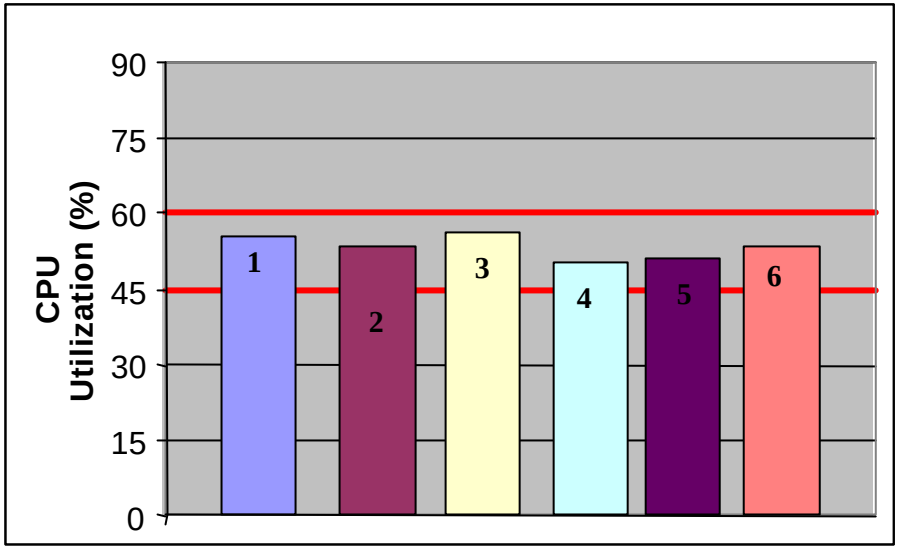
1

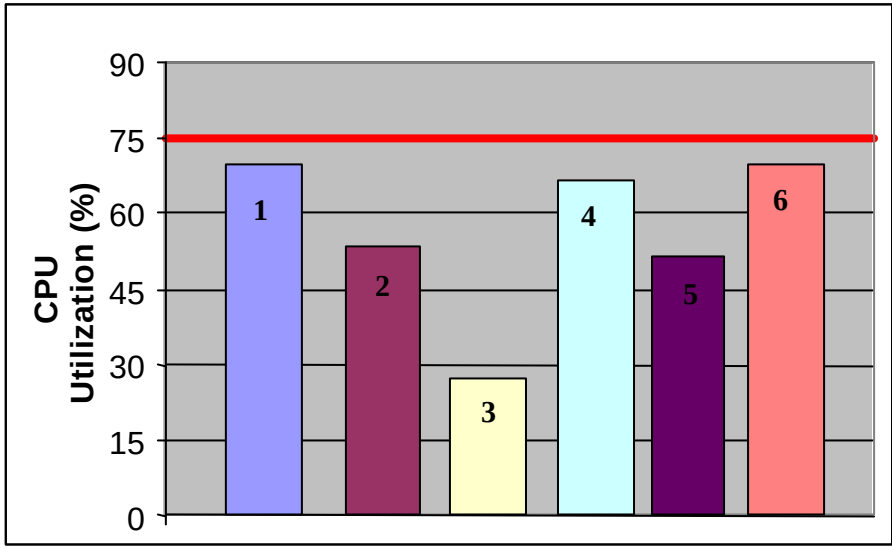
Quality of Service systems
(guaranteed services)

QoS-based systems do not imply more complicated algorithms, but well tuned systems

(over-provisioning is excluded!)

Example: Load balancing *vs* Load sharing ("good enough quality")





Load balancing →
"Equilize the load"

Load sharing →
"Avoid critical load cases"

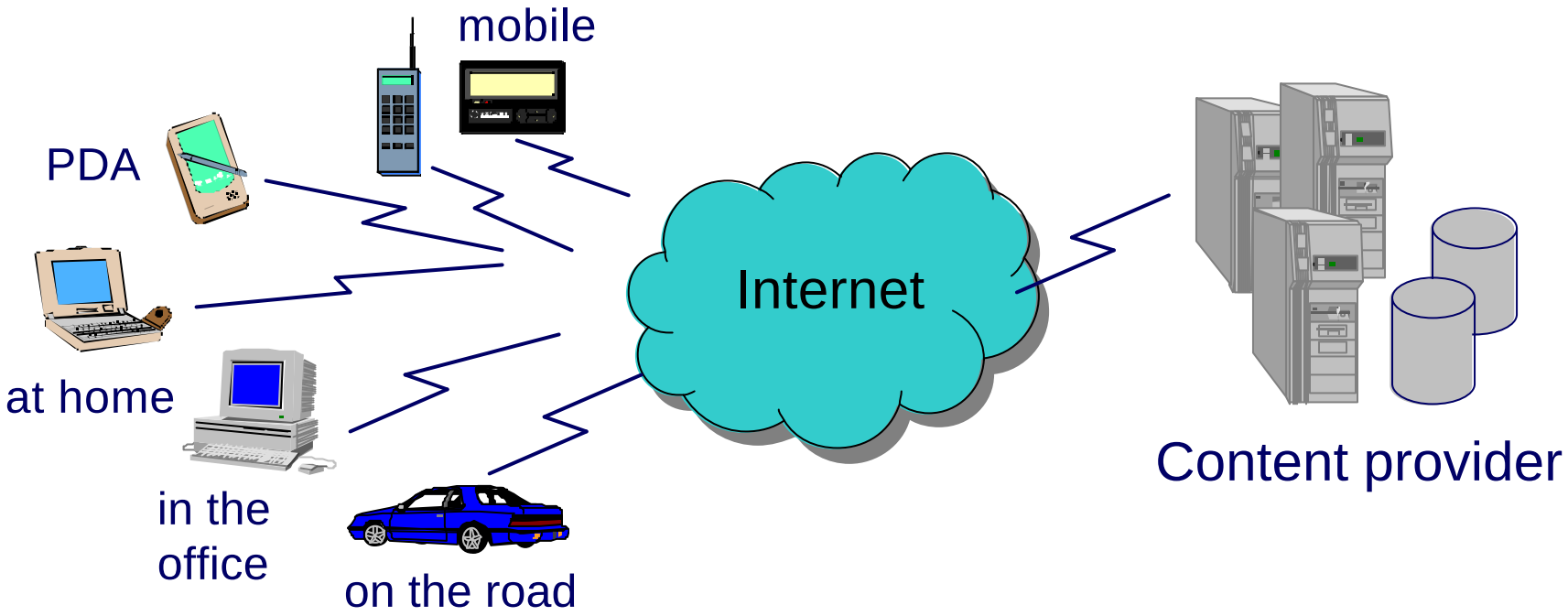
This choice makes the problem tractable, but it does not make the problem naive

ADAPTATION

ADAPTATION

- **Content adaptation**
  - Transcoding
  - Personalization
- **System adaptation**
  - Autonomic computing
  - Adaptive enterprise
  - E-business on demand
  - …

Content ADAPTATION

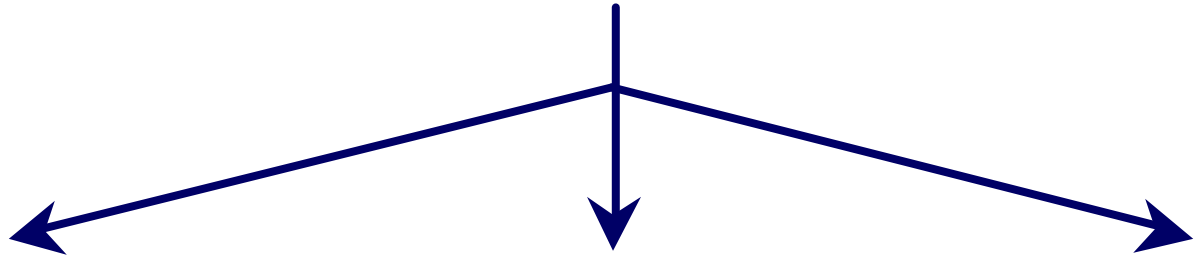


Pervasive computing environment

- **Numerous content types**
- **Various forms in which content can be viewed**
- **Range of available connection bandwidth, processing power, storage, display, memory, and format handling capabilities**

Transcoding

- **Process of converting a multimedia resource from one format to another**



Transformation **across** media types

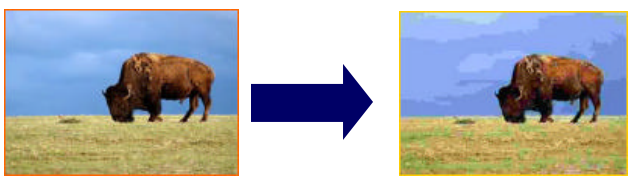
(e.g., from video to set of images, from speech to text)

Transformation **within** media types

(e.g., from JPEG to GIF)

Transformation of resource characteristics

(e.g., reducing number of colors for images)



QF=87 → 40 KB **QF=5 → 8 KB**

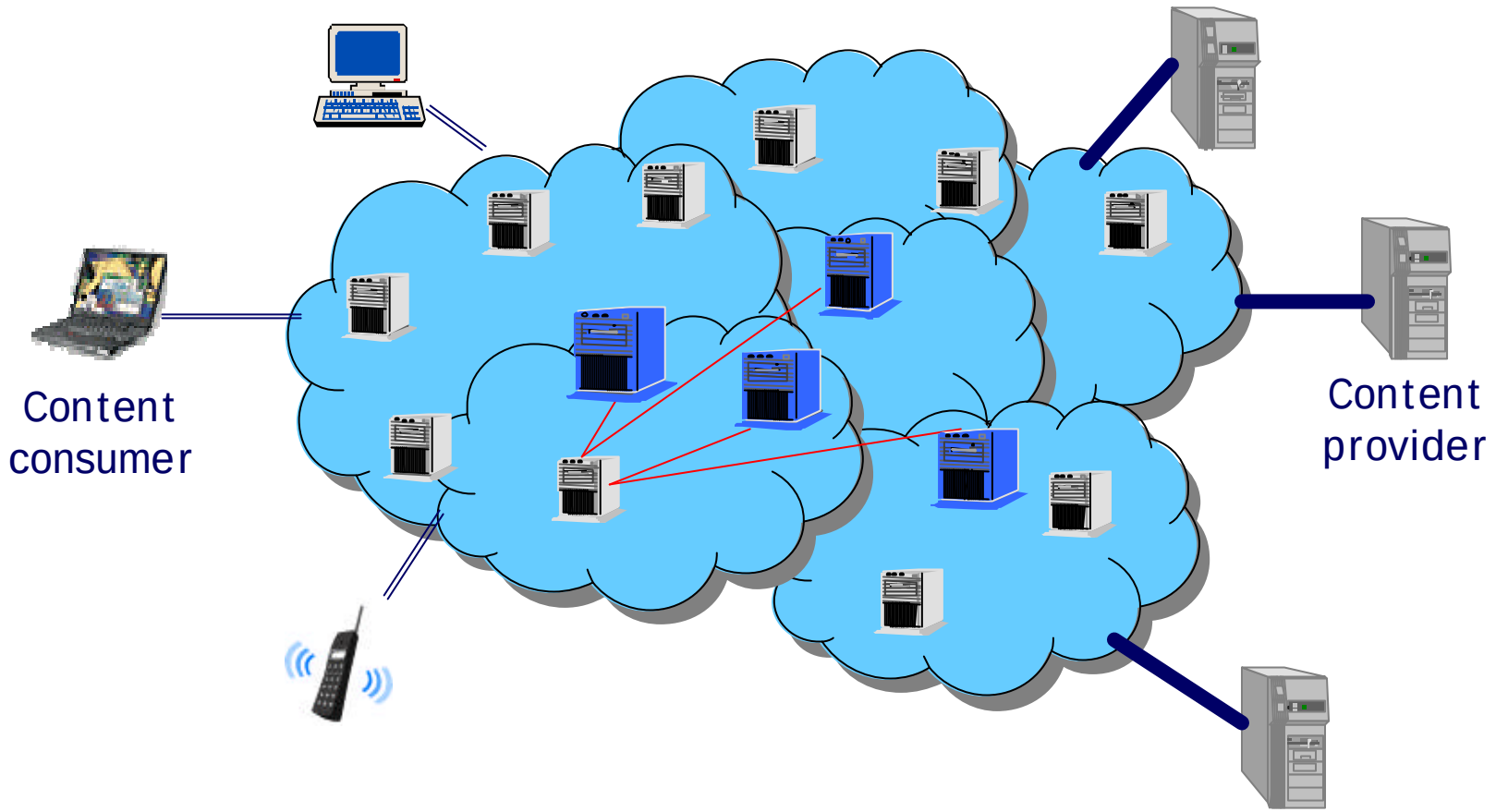
Personalization
("your view of the Web is the Web you want")

- **Examples**
  - Language translation
  - Insertion of personalized contents and banners
  - Adaptation to user interests, to navigation style, etc.
  - Content filtering
  - Delivery through secure channels
  - …
  - Adaptation to user location

Impacts on system infrastructure

- **"Adaptation services" can be computationally expensive (e.g., text2speech, video transcoding, translation, …)**
- **The working set augments (x10)**
- **The possibility of re-using resources (at the basis of Web caching solutions) decreases dramatically**

Internet-based services where intermediary nodes provide content adaptation



Nobody really knows the performance impact of the *Ubiquitous Web* on the system infrastructure

System ADAPTATION
(*autonomic computing*, *adaptive enterprises*)

- **Responsive**: System responsive to a new dynamic business model, to react quickly to unpredictable or unplanned changes in orders, supply, or demand, market changes, client need, partners. Companies must change in hours not days, weeks or months

- **Variable**: Flexible and variable structures to be able to respond quickly. Creating new structures that are driven by speed. Increase productivity, reduce costs

- **Integrated**: different softwares, different platforms

- **Resilient**: 24/7 services

Contributions we need in QoS and Adaptation

- **Design phase → ?**

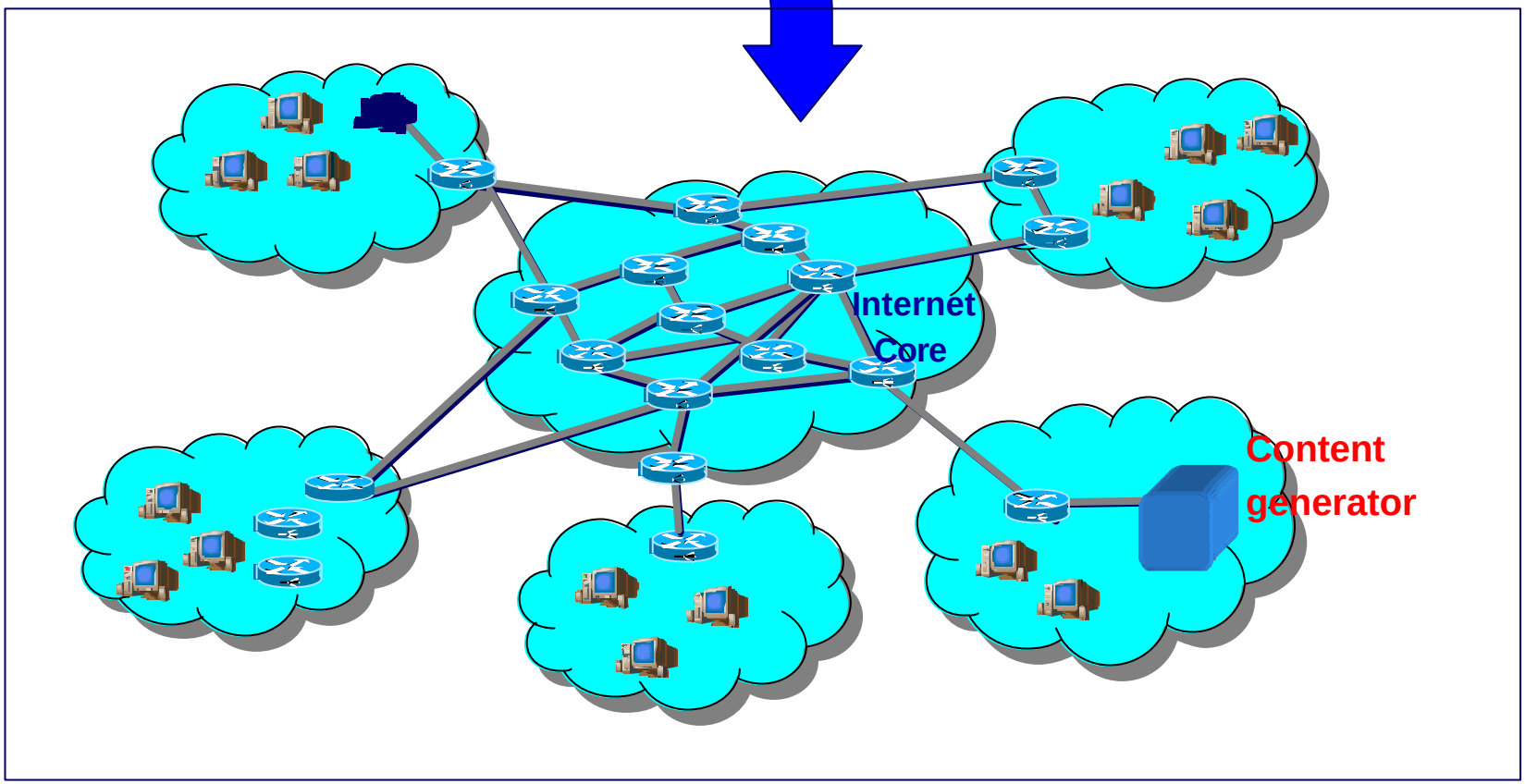
  The complexity of the systems supporting present Internet-based services is too high. Over-simplification is often the only viable alternative

- **Run-time phase → YES**

- **Testing phases → YES**

Not enough time to discuss it, but it is quite important

Performance model



System ADAPTATION (*needs*)

- **Mechanisms for self-adapting the system behavior to different conditions:**
  - **Continuous inspections**
    - Distributed monitoring, measuring, sinthesizing

      (Metrics should capture the most important elements of system behavior and user experience)
  - **Continuous positioning with respect to system capacity**
  - **Forecasting capacity**
  - **Autonomic decisions, e.g. for**
    - Load management, Resource replication, Resource/process migration, Access control

Autonomic decisions

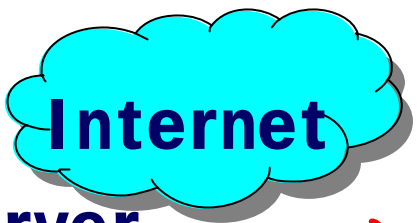
- **The ability of taking autonomous decisions based on:**
  - Service Level Objective rules
  - A measure of the internal system state with respect to system capacity
- **What should the autonomic decision process resemble?**

  ➔ **Whatever you like, but with some constraints**

First problem: Limited control

A typical service requires complex interactions among

- **Client platform (hw/sw)**
- **Client connection (last mile)**
- **DNS**



- **Web server**
- **Middleware**
- **Database server**

Second problem: Facing different orders of magnitude

- **Connection bandwidths**
  - Kbps → Mbps
  - **Mobile → Home → Institutional users**
- **Service times**
  - msec → secs [→ 1 Request » 1000 requests!]
  - **Static → Dynamic (CGI) → Dynamic (Java-based) → Dynamic+Secure services**
- **Workload characteristics**
  - Infrastructure can be susceptible to peaks
  - Web users travel in "herds"
  - URL "flashes" can cause unexpected peaks in traffic

→ Look for the *elephants*, forget the *ants*

Reaction to external forces ("not all resouces are created equal")

→ Forecast where the *elephants* are going

- **Resources degrading *gracefully*** (e.g., CPU)

  They cause a smooth deterioration of system

- **Resources degrading *suddenly*** (e.g., thread pools, memory, process descriptors, connection pool)

  A small change may have tragic consequences (→*chaotic behavior*)

Third problem: Limited TIME

- **We cannot use optimization models and algorithms that do not provide an answer in** time, e.g.
  - Provide dispatching decisions in milli-seconds
  - Monitor Internet weather and react to Internet problems in seconds
  - Respond within seconds to rapid changes in load (local and global load balancing)
  - Respond within minutes to substantial changes in workload (reconfigurations)
  - Respond within hours (not days, weeks, months) to business changes

Autonomic decisions - Bad news

- Models that found themselves on assumptions, such as linearity, independency, convex functions, may be highly inappropriate

- There is a potentially huge number of parameters

Autonomic decisions - Good news

- **Any considered self-system consists of layers, subsystems, hierarchies**
  - Traditional ***divide-et-impera*** schemes still work

- **We may not look for the "optimal solution"**
  - We seek "adequate" solutions

"Good enough quality"

- **Optimization? Yes, when it is possible.**
- **When the complexity is high and the time for taking a decision is short, traditional optimization is**

"Look for adequate solutions", "Be able to drive the system far from the critical zones", i.e.

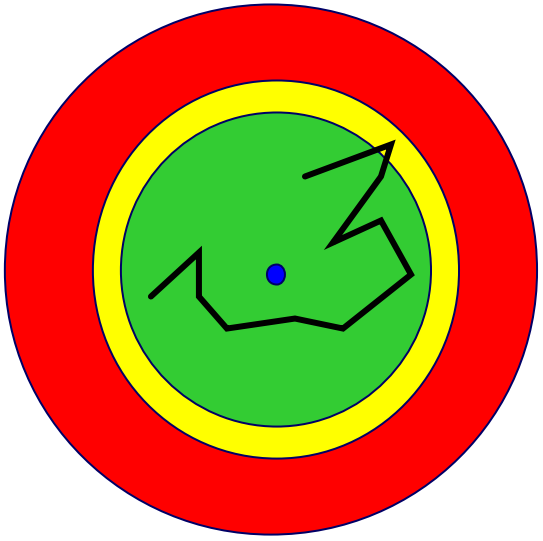
Avoid

→ service interruption

→ congestion

→ resource shortage

→ SLA violation



Whatever your favorite modeling technique is, "playing" with emerging Internet-based services has some common constraints:

1. **You have a (partial) control on a quite small fraction of the overall system**
2. **You have to face different orders of magnitudes (*ant & rhinos & elephants*)**
3. **You have strict time constraints**
4. **You may often forget "optimal performance", because "adequate performance" is good enough**

Emerging Internet-based services: New frontiers for performance models and applications

Michele Colajanni
University of Modena, Italy
colajanni@unimo.it

http://weblab.ing.unimo.it/people/colajanni/talks/qest2004.pdf