Mesure et analyse de données pour l'évaluation de performances de réseaux et de systèmes

Master 2R SL module MD

Jean-Marc Vincent and Arnaud Legrand[1]

[1] Laboratory ID-IMAG
MESCAL Project
Universities of Grenoble
{Jean-Marc.Vincent,Arnaud.Legrand}@imag.fr

Outline

1. Scientific context
2. Methodology
3. Master course
4. Performance indexes
5. Workload characterization

Outline

1. Scientific context
2. Methodology
3. Master course
4. Performance indexes
5. Workload characterization

Research activities in performance evaluation

Teams in Grenoble

- Mescal project: large systems (clusters and grids)
- Moais project: interactive parallel systems
- Drakkar team: networking
- Sardes: middleware
- Hadas: data-bases
- etc

Industrial collaborations

- France-Télécom R & D: load injectors, performances of middlewares
- HP-Labs: cluster computing, benchmarking
- Bull: benchmarking, performances analysis

Application context (1)

Complexity of computer systems

- **Hierarchy**: level decomposition: OS / Middleware / Application
- **Distribution**: asynchronous resources: memory, CPU, network
- **Dynamicity**: architecture and environment (reliability, mobility,...)
- **Scalability**: number of components (autonomous management)

Typical problems

- Minimize losses in routing policies
- Minimize active waiting in threads scheduling
- Maximize cache hits
- Optimise block sizes in parallel applications
- Maximize troughput of communication systems
- Fix time-outs, reemission periods, ...
- Fix the granularity: pages, blocks, tables, message sizes...

Application context (2)

Typical “hot” applications

- **Peer to peer systems**: dimensionning, control
- **Mobile networks**: ad-hoc networking, reactivity, coherence
- **Grids**: resources utilization, scheduling
- etc

Other application domains

- production systems: production lines, logistic,...
- embedded systems
- modelling of complex systems: biology, sociology,...
- etc

Outline

1. Scientific context
2. Methodology
3. Master course
4. Performance indexes
5. Workload characterization

Development of parallel/distributed aplications

- **Qualitative specifications:** Is the result correct ?

  - properties verifications: formal/automatic proofs

  - testing: critical dataset

- **Quantitative specifications:** Is the result obtained in an acceptable time ?

  - performance model

  - performance measurements

- **Problem identification**

  - debugging, log analysis

  - performance statistical analysis

- **Modification**

  - source code / libraries / OS / architecture

  - parameters of the system: dimensionning

  - control algorithms: tuning

Dual analysis

Understand the behavior of a distributed application

1. identification of distributed patterns, states of the system
2. pattern verification
3. time evaluation
4. global analysis of the execution and performance synthesis
5. system monitoring
6. global cost evaluation for the application user

Understand resources utilization

1. hierarchical model of resources
2. evaluation of utilization at:
   application level; executive runtime;
   operating system; hardware architecture
3. global cost evaluation for the resources manager

Evaluation methods

From abstraction to physical reality

Model **Method**

Remarks:

Hybrid methods (emulation, load injectors, synthetic programs,...)

Dynamical process of evaluation

Experimentation ⇒ Planning experiments methodology

Evaluation methods

From abstraction to physical reality

Model		**Method**
Mathematical	⟶	Analysis (formal, numerical, approximation)

Remarks:

Hybrid methods (emulation, load injectors, synthetic programs,...)

Dynamical process of evaluation

Experimentation ⇒ Planning experiments methodology

Evaluation methods

From abstraction to physical reality

Model		**Method**
Mathematical	→	Analysis (formal, numerical, approximation)
Software	→	Simulation (discrete events)

Remarks:

Hybrid methods (emulation, load injectors, synthetic programs,...)

Dynamical process of evaluation

Experimentation ⇒ Planning experiments methodology

Evaluation methods

From abstraction to physical reality

Model		**Method**
Mathematical	→	Analysis (formal, numerical, approximation)
Software	→	Simulation (discrete events)
Prototype	→	Observation (measures)

Remarks:

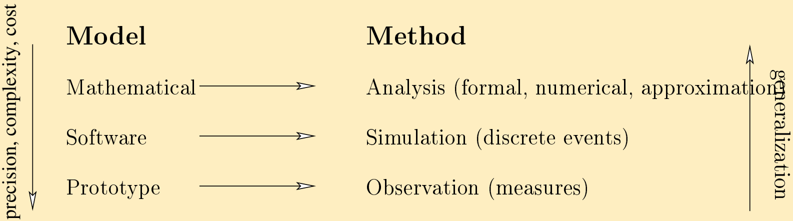
Hybrid methods (emulation, load injectors, synthetic programs,...)

Dynamical process of evaluation

Experimentation ⇒ Planning experiments methodology

Evaluation methods

From abstraction to physical reality



precision, complexity, cost

Model | **Method**

Mathematical → Analysis (formal, numerical, approximation)

Software → Simulation (discrete events)

Prototype → Observation (measures)

generalization

Remarks:

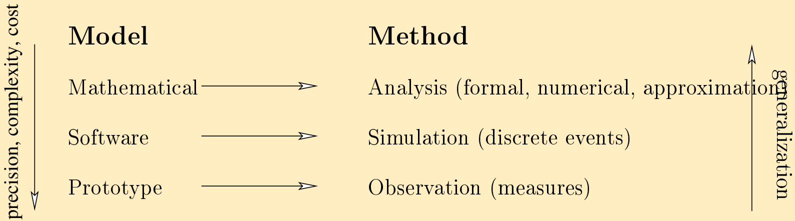
Hybrid methods (emulation, load injectors, synthetic programs,...)

Dynamical process of evaluation

Experimentation ⇒ Planning experiments methodology

Evaluation methods

From abstraction to physical reality



precision, complexity, cost

Model → **Method**

Mathematical → Analysis (formal, numerical, approximation)

Software → Simulation (discrete events)

Prototype → Observation (measures)

generalization

Remarks:

Hybrid methods (emulation, load injectors, synthetic programs,...)

Dynamical process of evaluation

Experimentation ⇒ Planning experiments methodology

Steps for a Performance Evaluation Study (Jain)

1. State the goals of the study: level of decision, investment, optimization, technical,...
2. Define system boundaries.
3. List system services and possible outcomes.
4. Select performance metrics.
5. List system and workload parameters
6. Select factors and their values.
7. Select evaluation techniques.
8. Select the workload.
9. Design the experiments.
10. Analyze and interpret the data.
11. Present the results. Start over, if necessary.

Outline

1. Scientific context
2. Methodology
3. Master course
4. Performance indexes
5. Workload characterization

Aim of the course

Objective

1. Be able to analyze and predict performances of parallel/distributed systems
2. Be able to build a software environment that produces the performances indexes.

Methods

1. Specification and identification of problems: modelling
2. Analysis of quantitative models: formal, numerical, simulation (2nd year)
3. Experimentation and statistical data analysis.

Organisation of the course

8 lectures 1h30

1. Performances of computer systems: quality of service, performance indexes,...
2. Avalysis and visualization of univariate statistical data
3. Performance measurements of computer systems: tools and analysis
4. Trace and analysis of distributed applications
5. Simulation of computer systems
6. Factorial analysis and experimental planning
7. Performance evaluation environments
8. Validation of experimental methods

Applications: networking, distributed applications, parallel/grid computing

References: text books

- **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling.** Raj Jain *Wiley 1991 (nouvelles versions)*
  Covers the content of the course, a complete book
- **Performance Evaluation** Jean-Yves Le Boudec EPFL electronic book
  http://ica1www.epfl.ch/perfeval/lectureNotes.htm
  Covers the statistical part of the course
- **Measuring Computer Performance: A Practitioner's Guide** David J. Lilja *Cambridge University press* 2000
  Covers the practical part of measurement and benchmarking
- **Discrete-Event System Simulation** Jerry Banks, John Carson, Barry L. Nelson, David Nicol, *Prentice Hall*, 2004
  Covers the part on simulation

References: journals and conferences

- **General**: JACM, ACM Comp. Surv., JOR, IEEE TSE,...
- **Specialized**: Performance Evaluation, Operation research, MOR, ACM TOMACS, Queueing Systems, DEDS, ...
- **Application**: IEEE TPDS, TC, TN, TAC, Networks,...
- **Theoretical**: Annals of Probability, of Appl. Prob, JAP, Adv. Appl. Prob,...
- **Conferences on performances**: Performance, ACM-SIGMETRICS, TOOLS, MASCOT, INFORMS, ...
- **Conferences on an application domain**: ITC, Europar, IPDPS, Renpar, ...
- **National seminars**: Atelier d'évaluation de performances,...

Evaluation

- Article analysis and synthesis
- Presentation of the article
- Exam

Outline

1. Scientific context
2. Methodology
3. Master course
4. Performance indexes
5. Workload characterization

Networking

Flow performance

- latency, waiting time, response time
- loss probability
- jitter

Operator performance

- bandwidth utilisation
- achievable throughput
- loss rate

Quality of service

contract between user and provider

service guarantees

tradeoff between utilization and QoS

Parallel processing

Program execution

- makespan, critical path
- speedup, efficiency
- active waiting, communication overlapping
- throughput

System utilization

- cpu utilization, idle time
- memory occupancy
- communication throughput

Parallel programming and scheduling

granularity of the application
tradeoff between utilization and makespan

Distributed applications

Application

- response time
- reactivity
- throughput (number of processed requests/unit time)
- streaming rate

System utilization

- service availability
- resource utilization
- communication throughput

System security

- reliability (error-free period)
- availability

Synthesis

User vision:

- optimize performance

Resource vision:

- minimize cost

Bottleneck = resource with the highest utilization rate

Outline

1. Scientific context
2. Methodology
3. Master course
4. Performance indexes
5. Workload characterization