disco

Distributed Computer Systems Lab

http://disco.cs.uni-kl.de

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Worst Case Analysis of Distributed Systems

Prof. Dr.-Ing. Jens B. Schmitt
(jschmitt@cs.uni-kl.de)

Murphy's Laws

1. In any field of endeavor, anything that can go wrong, will go wrong.
2. Left to themselves, things always go from bad to worse.
3. If there is a possibility of several things going wrong, the one that will go wrong, is the one that will cause the most damage.
4. Nature always sides with the hidden flaw.
5. If everything seems to be going well, you have obviously overlooked something.

Overview of Lecture

1. **Introduction**
   - Motivation, Background
2. Min-plus Algebra Primer
3. Network Calculus Basics
   - Arrival & Service Curve
   - Basic Bounds
   - Concatenation
   - Packetizer
4. Advanced Network Calculus
   - Scheduling
   - Network Analysis
   - Tool Support: The DISCO Deterministic Network Calculator
   - Flow Transformations
   - Feedback Systems

Performance in Distributed Systems

- Performance is key criterion for any system
  - quest for highest performance at lowest cost
  - any computer scientist/engineer needs basic knowledge on how to evaluate, dimension, control system performance
- **Distributed** systems **share** resources
  → performance is harder to model and analyse
  → performance is harder to control
- Thorough performance planning is key success factor
  - for example, Internet providers need to know when to expand their transport capacity
  - data center resources need to be dimensioned correctly
  - cellular networks need coverage planning
  - …

Performance Models

- Models are abstractions from real systems

- Models as views on a system
  - Functionality
  - Security
  - Cost
  - **Performance 🡐**

- What are they good for?
  - **Performance Evaluation**
    - Comparison of different system alternatives
  - **Performance Dimensioning**
    - Planning of future system to satisfy certain performance targets
  - **Performance Control**
    - Online control to meet certain performance assurances
    - Tuning of performance parameters

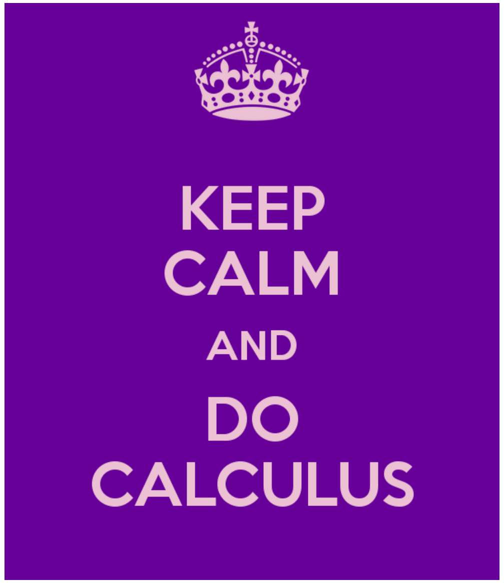
Common Pitfalls in Performance Modeling

- Modeling without understanding the problem
- Using the wrong method
- Inappropriate level of detail
- Wrong selection of factors during abstraction
- Missing sensitivity analysis
- Assuming no change in future
- Ignoring errors in input
- Only mean behavior is modeled
- Omitting assumptions and limitations

Worst Case Performance Analysis

- Often knowledge on average/equilibrium case behaviour is not enough
- How does the system behave in the worst case?
- What is the worst case
  - maximum possible load
    → some **regulation** takes place
  - minimum service guarantees
    → system follows a deterministic **schedule**
- A (system) theory for worst case analysis is **Network Calculus**, it abstracts from
  - specific regulation schemes
  - specific scheduling algorithms

  NC is an algebraic method to virtually construct
  the worst case performance in a network of queues



Worst Case Analysis: Pros and Cons

- Why would the worst case **not be** interesting?
  - It rarely happens
  - Dimensioningwise it is wasteful
  - Many applications do not care about worst case
  - It can be trivial and thus useless

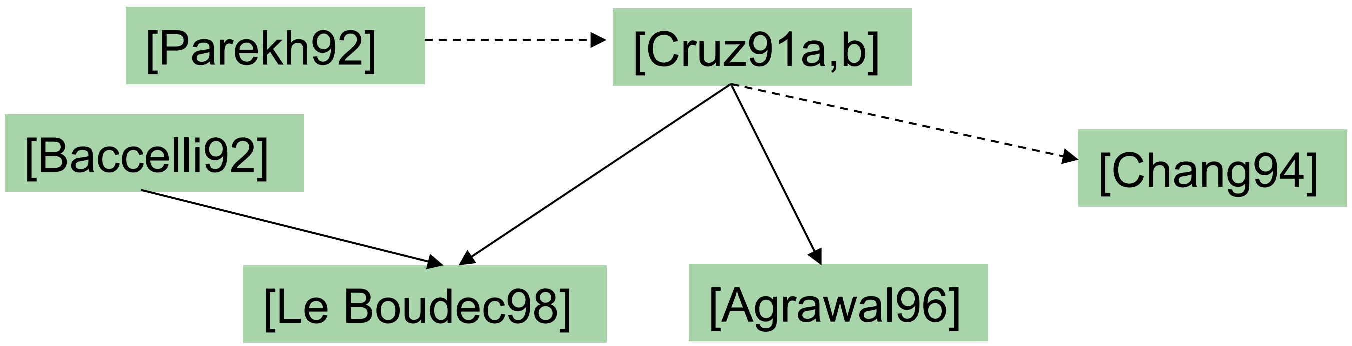
- Why would the worst case **be** interesting?
  - Because it *can* happen
  - It is a figure of (relative) merit
  - Real-time / mission-critical systems care *only* about worst case
  - It is „easy“ to find
  - It requires little assumptions

Origins of Network Calculus / Genealogy

- Network calculus does not deal with arrivals and departures but bounding processes
- Origins in traffic control for IP and ATM QoS

  „traffic policing/shaping+packet scheduling+flow admission control = guarantees“
- Compared to queueing theory still in its very infancy

- Genealogy of network calculus



Case Study: Cisco LLQ (1)

- Cisco *Low Latency Queueing (LLQ)*
  - aka *Strict Priority Queueing (SPQ)*
- Goal
  - evaluate delay differentiation capabilities for a single Cisco router using LLQ vs. FIFO
- Performance Analysis ← no Cisco router available ☺
  - Queueing Theory and Network Calculus to analyse average case respectively worst case behaviour of single node
  - effort: one day literature research and some thinking
  - the average waiting time for class *i* is given by (ass. Poisson arrivals)

$$E(W_i) = \frac{E(T_0)}{\left(1 - \sum_{j=1}^{i} \rho_j\right)\left(1 - \sum_{j=1}^{i-1} \rho_j\right)} \text{ with } \rho_j = \frac{\lambda_j}{\mu_j}, \text{ and}$$

$$E(T_0) = \lambda E(\tau^2)/2 = \sum_{j=1}^{n} \lambda_j E(\tau_j^2)/2 = \sum_{j=1}^{n} \lambda_j \frac{\sigma_j^2 + (1/\mu_j)^2}{2} \text{ with } \lambda = \sum_{j=1}^{n} \lambda_j$$

Performance Evaluation: Cisco LLQ (2)

- The worst case delay per traffic class *i* is

$$h\left(\gamma_{r_i,b_i},\beta_i^P\right)=\left(\sum_{j=1}^{i} b_j + L_i\right)\Big/\left(C-\sum_{j=1}^{i-1} r_j\right)$$