Skalierbarkeit einer Plattform für föderierte ortsbezogene Dienste

ITG Fachgruppe 5.2.4 Workshop

Michael Scharf
scharf@ikr.uni-stuttgart.de

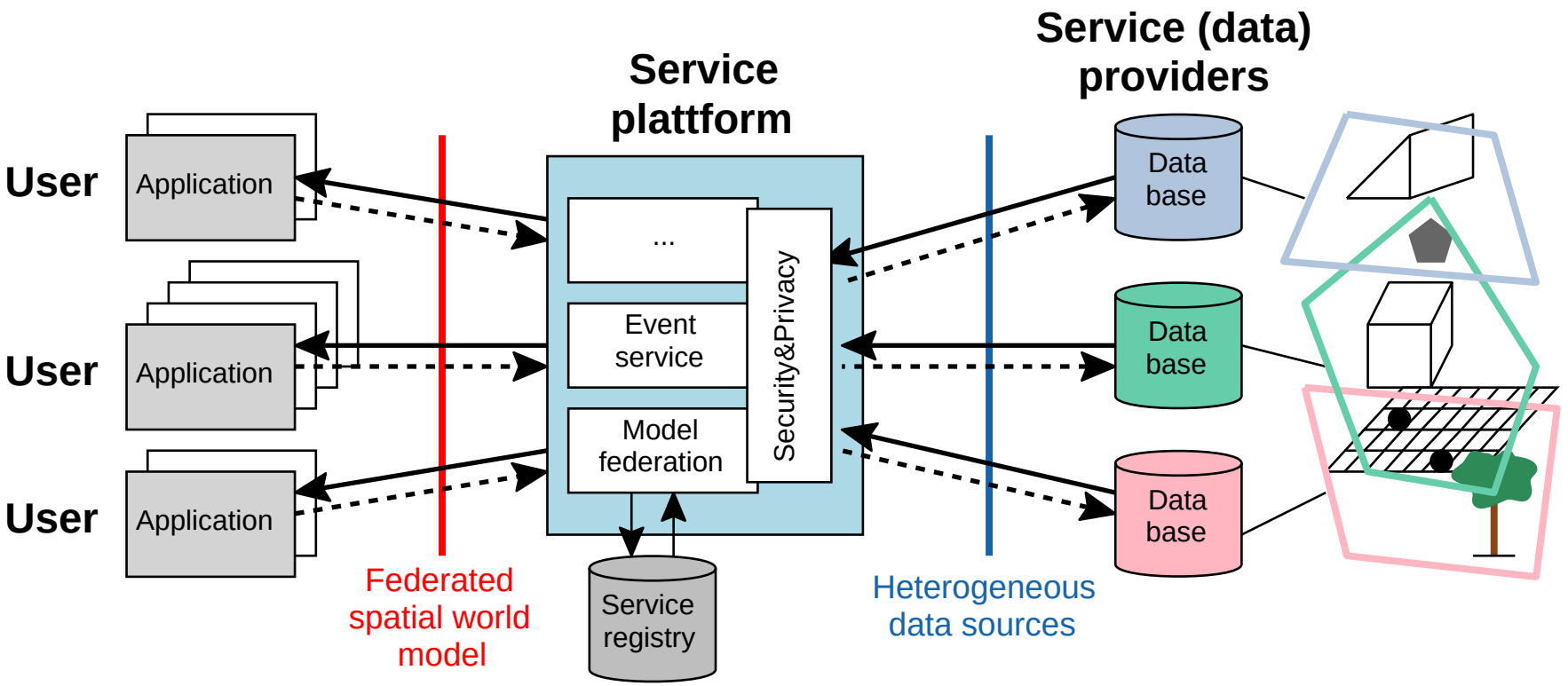
Stuttgart, 9. Juni 2005

Outline

- **Federation of service providers**
- **Evaluation model**
- **Evaluation**
  - Analytical approximation
  - Analytical analysis for exponential case
  - Heavy-tailed server response times
- **Enhancements**
- **Conclusions**

Federation of service providers (1)

Nexus vision



- Service platform for location-based applications
- Federation: Integration of different databases, operated by different providers
- Special case of "Composite Web Services"

Federation of service providers (2)

Challenges

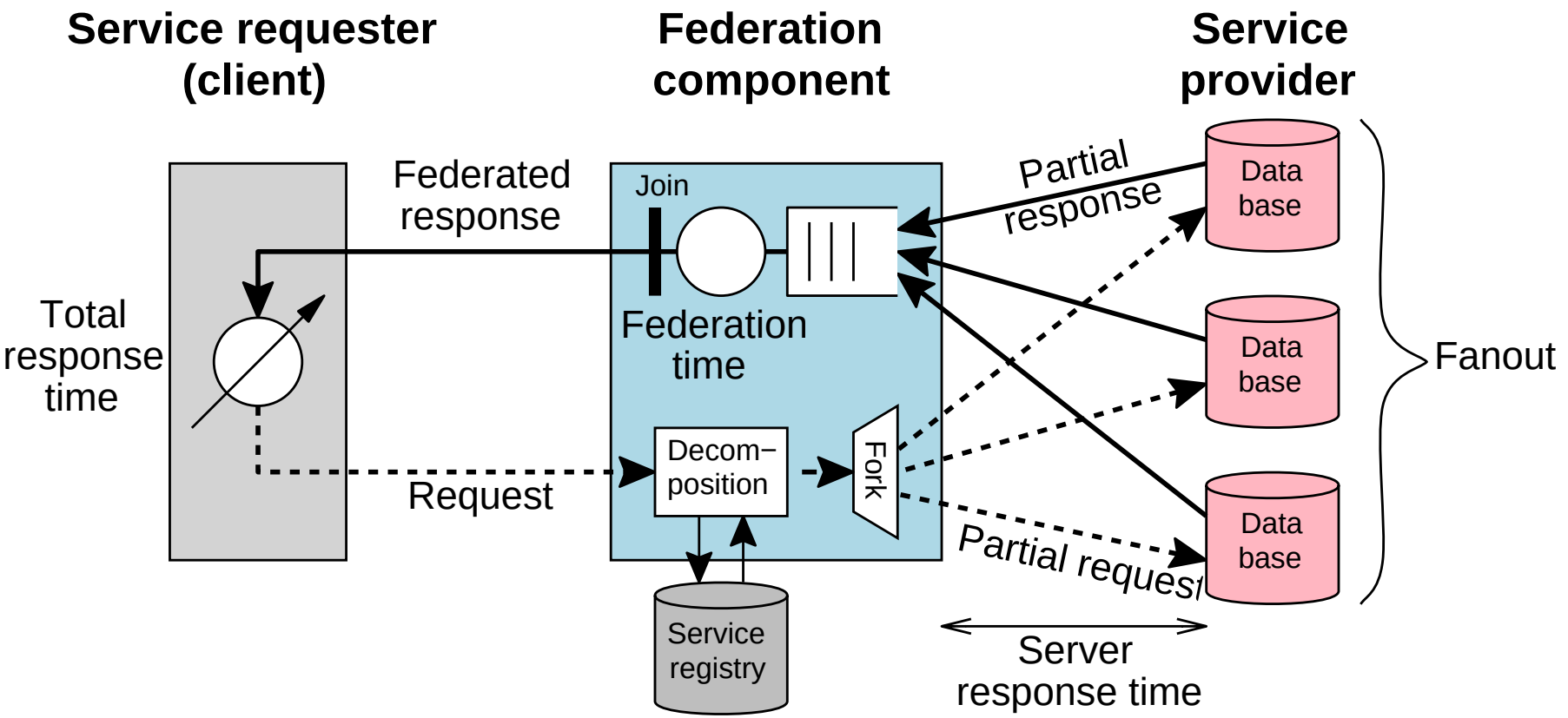
- **Integration of (many) heterogeneous data sources**
- **Interoperation of (many) different providers and infrastructures**
- **Scalability to large-scale scenarios**

System scalability

- **What does *scalability* mean?**
  - No unique definition available
  - E. g.: Scalability metric of Jogaleka&Woodside: $\frac{\text{throughput}}{\text{responsetime} \cdot \text{cost}}$
  - ➥ Scability is not only a technical issue
- **Performance of system architectures**
  - Complex problem, in particular if distributed databases are involved
  - Technology-dependent
  - ➥ Difficult to handle with "classical" performance evaluation methods

Evaluation model (1)

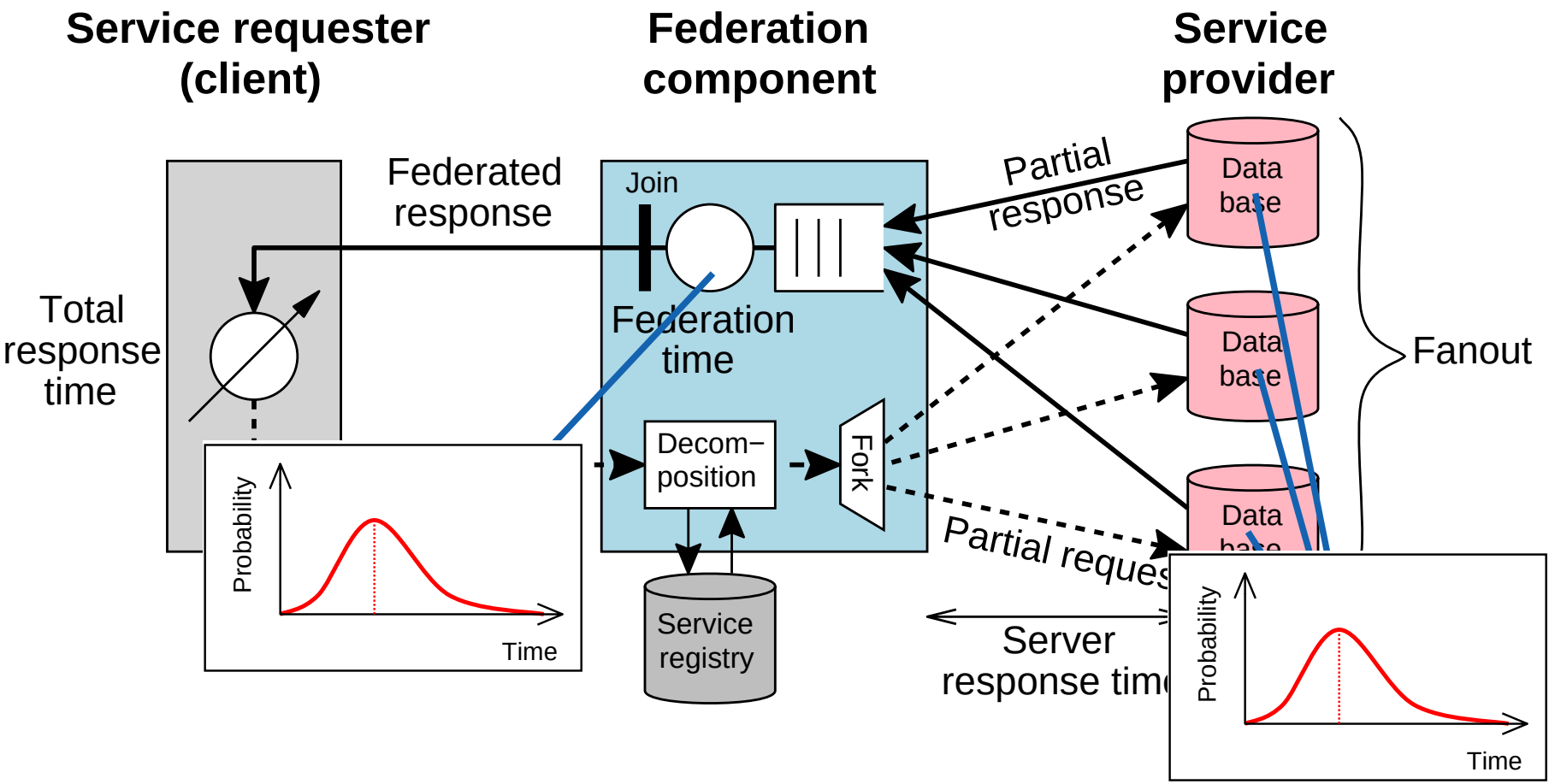
Model



➥ **Example for a fork-join queueing network**

Evaluation model (2)

Model



Simplifications:

- ❒ No resource contention by concurrent requests
- ❒ All servers have same characteristics

Evaluation model (3)

Parameters

1. **Fanout factor N: Number of servers queried in parallel**
2. **Response time distribution of servers**
3. **Processing time distribution of federation component**

➥ **Relative performance of federation:**

$$\kappa = \frac{\text{mean response time of a server}}{\text{mean processing time of federation}}$$

Main performance metric

Normalized federation response time ("federation slowdown"):

$$S = \frac{\text{mean total response time}}{\text{mean response time of a server}}$$

"How much longer do I have to wait if several servers are queried?"

Analytical approximation

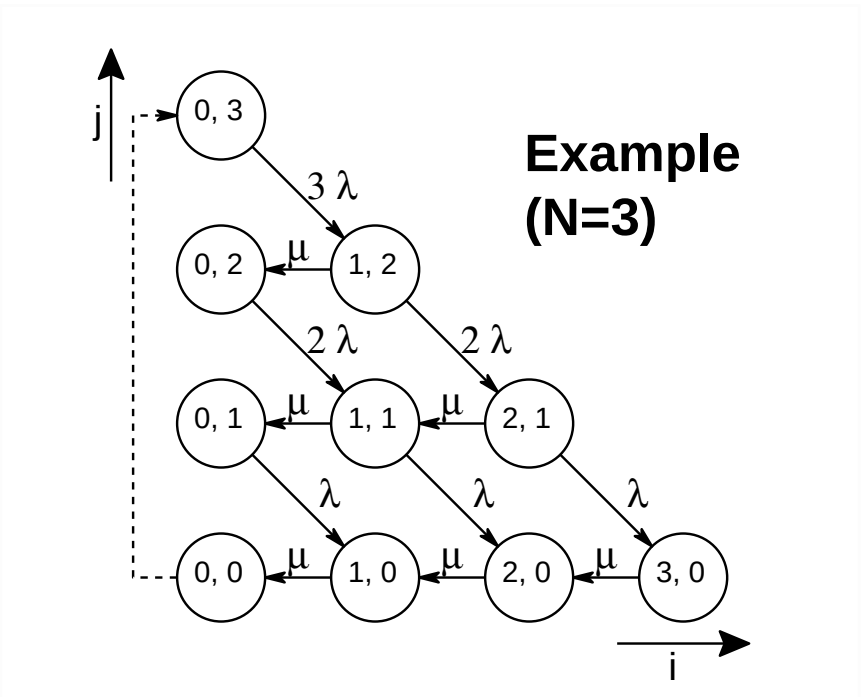
- **Extreme case 1: Very fast federation ($\kappa >> N$)**
  - Synchronization delay dominates
  - Total response time can be approximated by maximum of N i. i. d. RV
- **Extreme case 2: Very slow federation ($\kappa << N$)**
  - Federation processing time dominates
  - Total response time can be approximated by N processing times

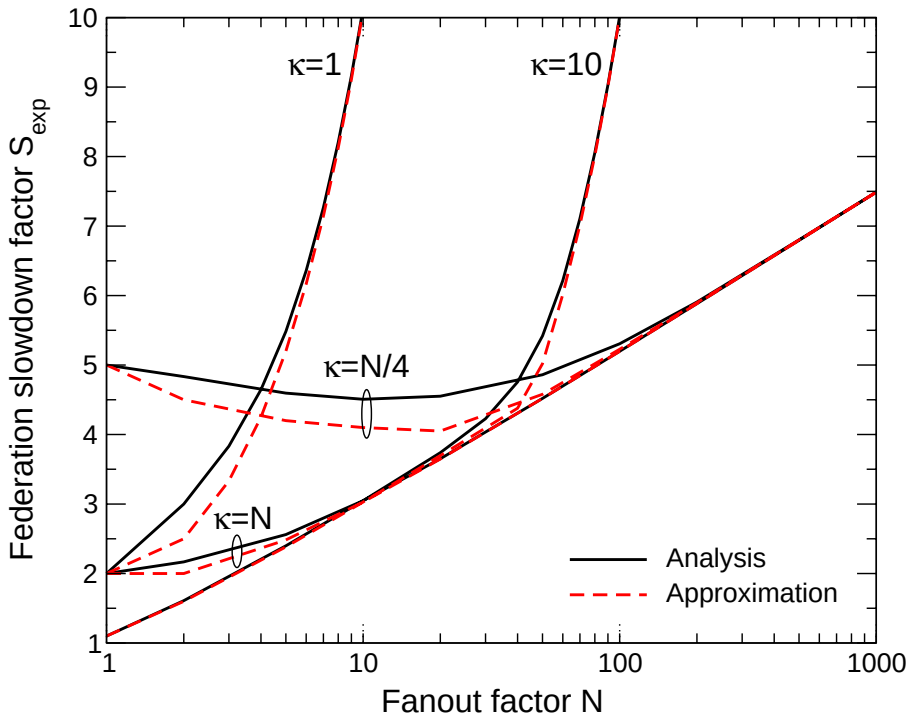
➥ **Combination of both cases gives good approximation**

Analytical analysis: Exponential distributions

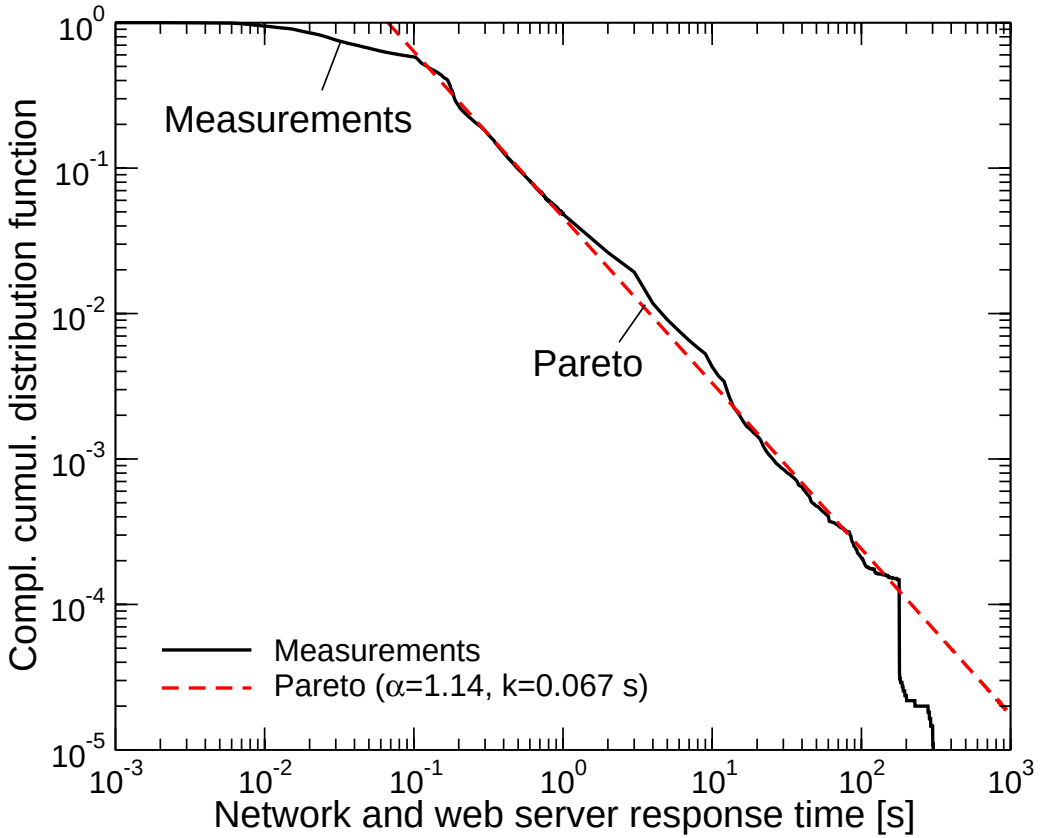
- **Can be modelled by two-dimensional Markov chain**
- **Mean response time can be determined by recursion**
  - Exact solution for small N
  - For large N numerical solution

➥ **Minimal response time that scales logarithmically with fanout N**





Real server response times: WWW example



Data source

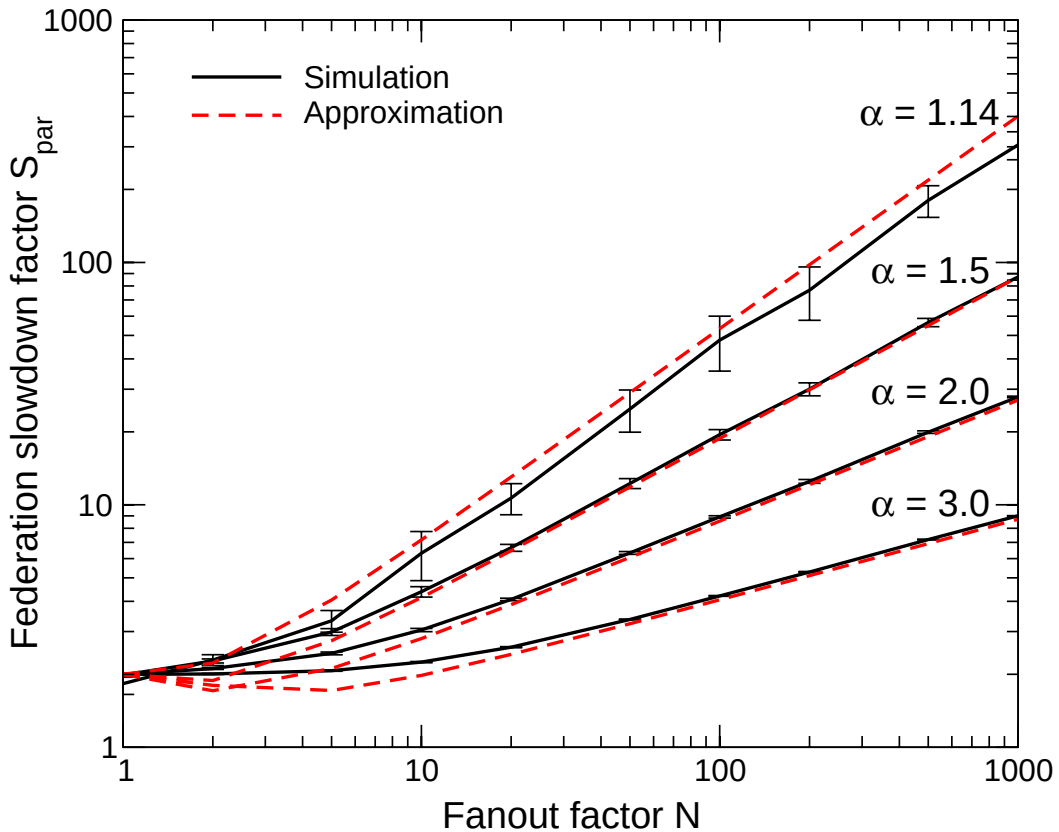
- **Measurements of Web access from student dormitory in Stuttgart, Nov. 2000**
- **Courtesy of J. Färber, IKR**

- **Measurements reveal heavy-tailed response times in the Web**

  Burstiness of requests, heavy-tailed amounts of data, ...

➥ **What does this mean for the federation concept?**

Effect of heavy-tailed server response times



Assumptions

- Server response times Pareto distributed
- Federation processing times neg-exp. distributed
- $\kappa = N$

➥ **Response time scales with $N^{1/\alpha}$**
α : shape factor of Pareto distribution

➥ **Large-scale federation impossible for significant tails (small α)**

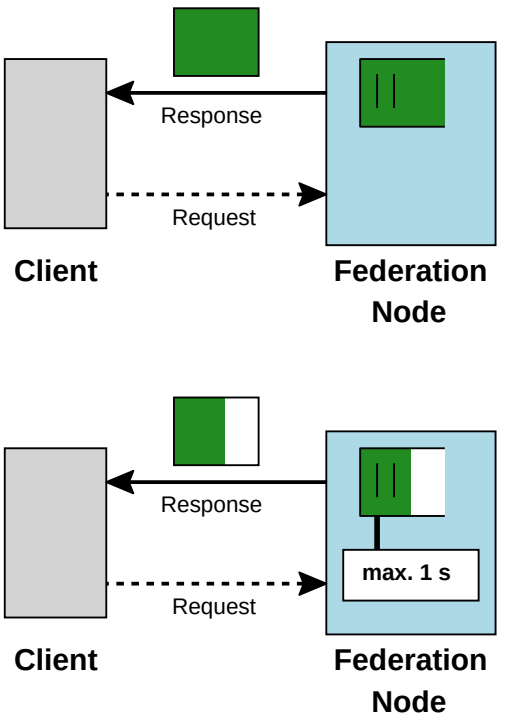
Potential solutions

- **Solution 1: Just avoid the heavy-tails...**

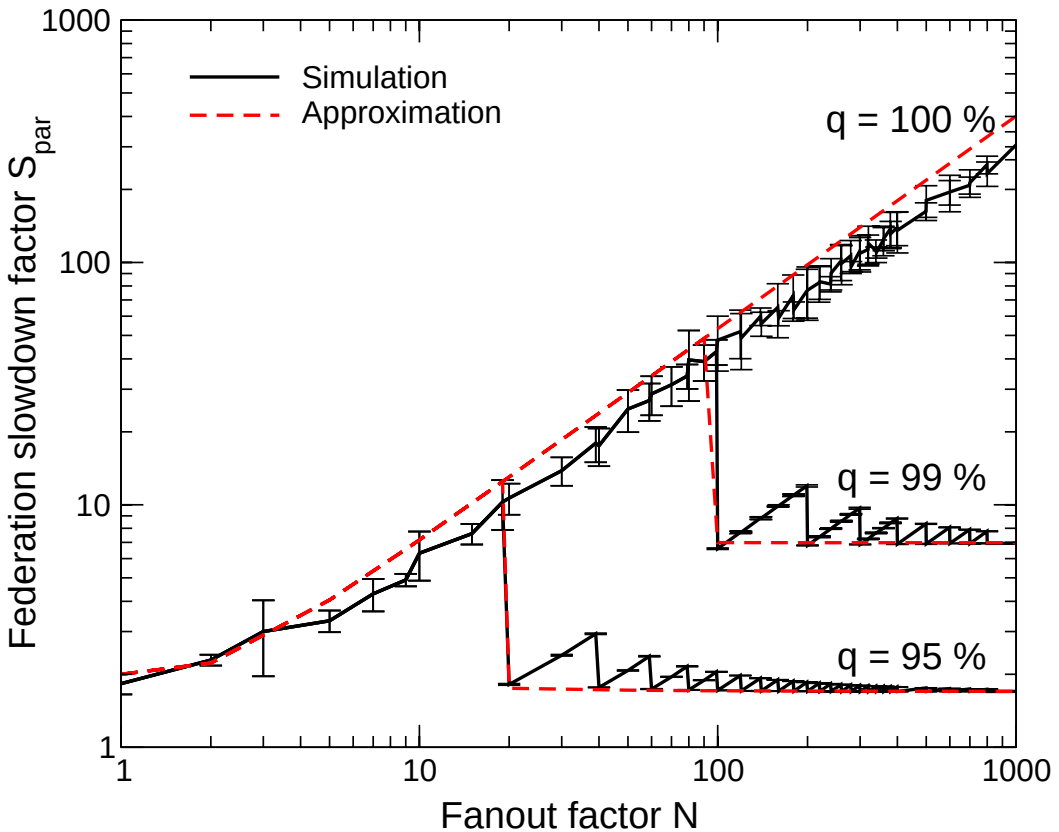
  But: Problem of synchronization delays still exists

- **Solution 2: Do not wait for slow servers**
  - "Do users really need a complete list of all italian restaurants?"
  - Different possibilities for implementation
    - Timeouts ("maximum 1 second")
    - Thresholds ("99% of the servers")
    - Iterations ("query first 10 servers, more later if the result is not sufficient")
  - Mathematically related to order statistics theory

➥ **Tradeoff between performance and result completeness**



Performance improvements by incomplete results



Assumptions

- **Server response times Pareto distributed ($\alpha = 1.14$)**
- **Federation processing times neg-exp. distributed**
- $\kappa = N$

➥ **Significant improvements by omitting a few servers**

Conclusions

- **Future location-based services could benefit from federation of data stored in distributed data bases**
- **Challenge for large-scale federation: Synchronization delays**
  - Criticial if some servers have long response times
  - Heavy-tailed response times can be observed in Web-like systems
- **Queueing model for response time of a federation**
  - Demonstrates synchronization effects
  - Accurate closed-form approximations
- **Incomplete query results as a solution to reduce response times?**

Further reading

Michael Scharf: "On the Response Time of Large-scale Composite Web Services", Proc. ITC 19, Beijing, China, Sept. 2005