SLA-Driven Planning and Optimization of Enterprise Applications

H. Li[1], G. Casale[2], T. Ellahi[2]

[1] SAP Research, Karlsruhe, Germany
[2] SAP Research, Belfast, UK

Presenter: Giuliano Casale

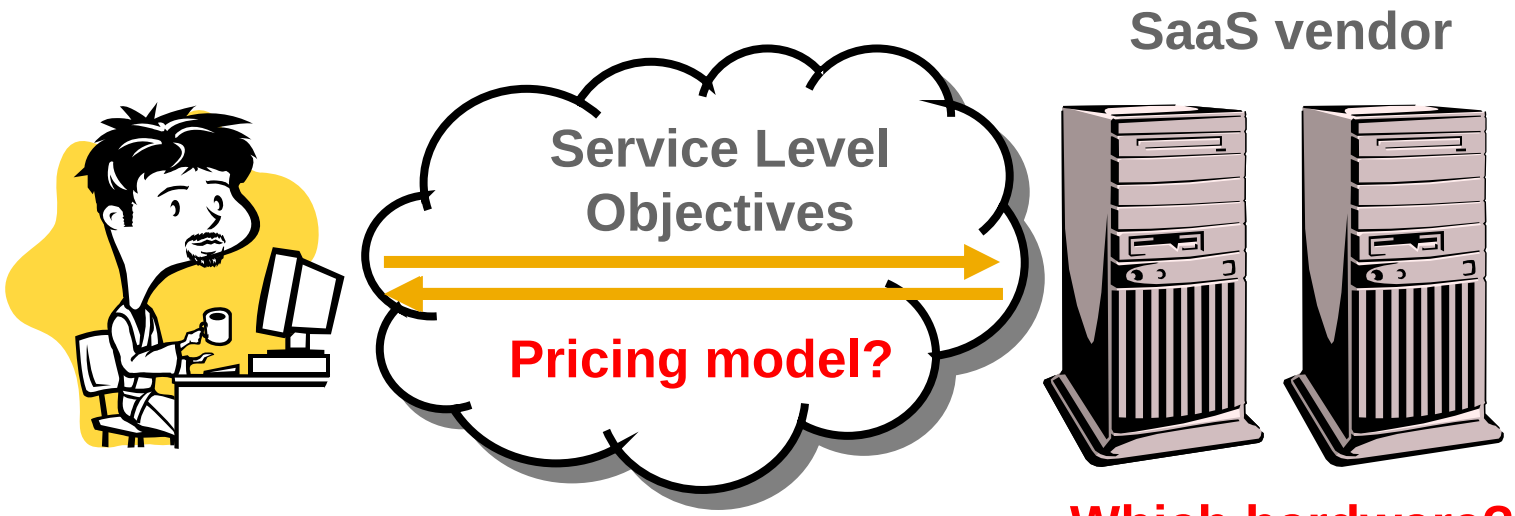
WOSP/SIPEW Conference
San Jose, Jan 29th, 2010

Sizing enterprise applications

- Capacity planning
  - Periodic scalability assessment
  - Focus on performance and costs
- Service Level Agreements (SLAs)
  - Constraints on responsiveness (throughput, resp. time, …)
- Total Cost of Ownership (TCO)
  - HW provisioning
  - SW upgrades
  - Infrastructure management
  - Power consumption

Motivation: SaaS & Sizing

- Software-as-a-Service (SaaS) increasingly more popular



Which hardware?
How many virtual cores?
Optimal SW tuning?

- Strong need for sizing frameworks

Contributions to Sizing

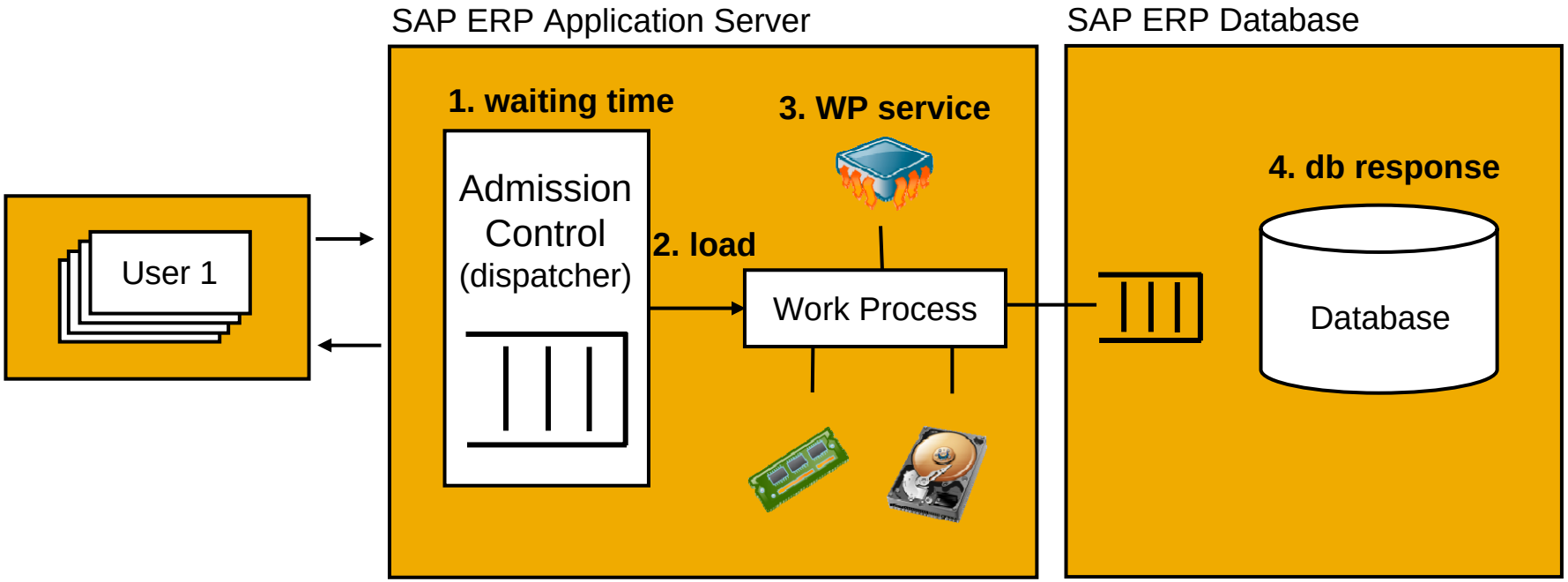
- Performance model of a commercial enterprise application
  - SAP ERP
- Hardware cost model
  - Benchmark-driven approach
- Power consumption cost model
  - Measurement-driven model
- Multi-objective optimization approach to sizing
  - How to simultaneously minimize cost and response time

Performance Modeling

SAP ERP

- ERP: management of business processes and resources
- Workload complexity
  - Sizing based on reference workloads, e.g., sales transact.
- Performance models
  - Commercial application (not a toy system)
    - ERP has 10 times the lines of code of the Windows operating system
  - Can we define models that are both simple and effective?

End-to-end Performance

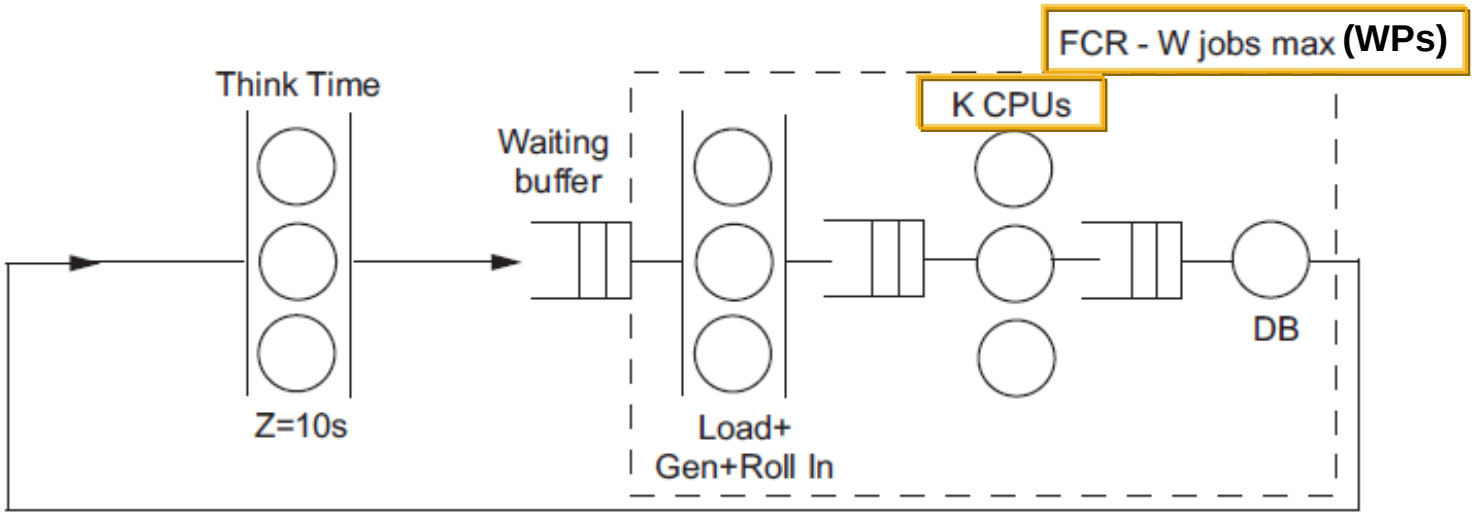


Performance Model

- Limited literature on ERP modeling
  - Rolia et al., ROSSA 2009 – layered queueing model (LQM)
- Queueing networks with finite capacity regions (FCRs)
  - FCR = admission control region
  - More general than MVA queueing networks
  - Less expressive than LQMs (e.g., no async behavior)
- Why FCR queueing networks?
  - Simplest models with admission control
  - Promising for analytical approximations
    - multiclass iterative approximations already available

Performance Model

- FCR queueing network of SAP ERP
  - Performance evaluated with Java Modelling Tools (JMT) simulator



Prediction Accuracy

- Comparison of model and measurement
  - Over-sizing needed when #WPs close to #CPUs
  - Best performance when #WPs much larger than #CPUs

vCPU	WP	usrs	R (Model)	R (Meas.)	U (Model)	U (Meas.)
2	1	300	25.86	28.76	0.30	0.51
2	2	300	9.15	9.32	0.56	0.65
2	4	300	3.57	7.62	0.79	0.66
2	8	300	1.76	2.41	0.92	0.86
2	16	300	1.11	1.37	0.97	0.97
2	32	300	1.05	0.91	0.98	0.99
4	1	300	25.68	26.8	0.15	0.28
4	2	300	9.19	10.6	0.28	0.37
4	4	300	1.61	1.67	0.46	0.51
4	8	300	0.47	1.16	0.52	0.65
4	16	300	0.41	0.43	0.53	0.63
4	32	300	0.37	0.38	0.53	0.62

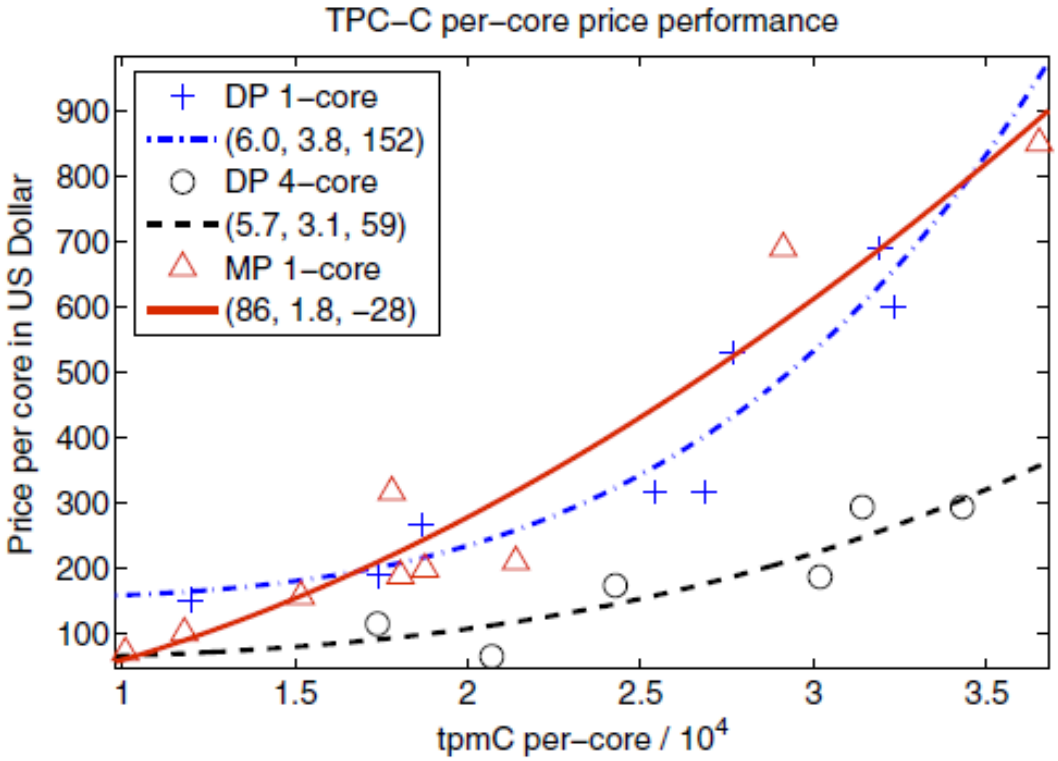
Cost Modeling

Modeling Hosting Costs

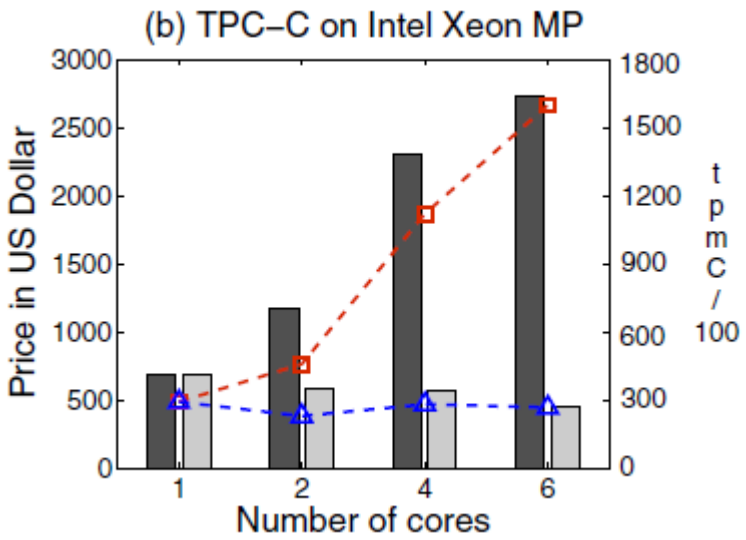
- Focus on technical components of TCO
  - Goal: find relation between service demand and costs
- Hardware costs
  - Parallelism (#cores/CPU)
  - Price (price/core)
- Usage costs
  - Power consumption
- Service demand
  - Nominal performance (tpmC/core: TPC-C results)

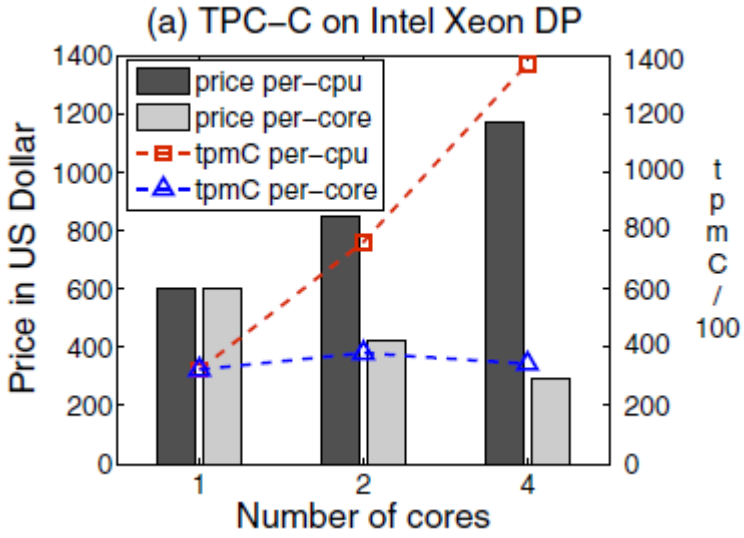
Cost vs Performance

- Publicly available data for Intel Xeon DP/MP
- Tested polynomial, exponential, and power laws
  - Best RSS for power law: $f(x) = c_1 x^{c_2} + c_3$



Cost vs Performance vs Cores





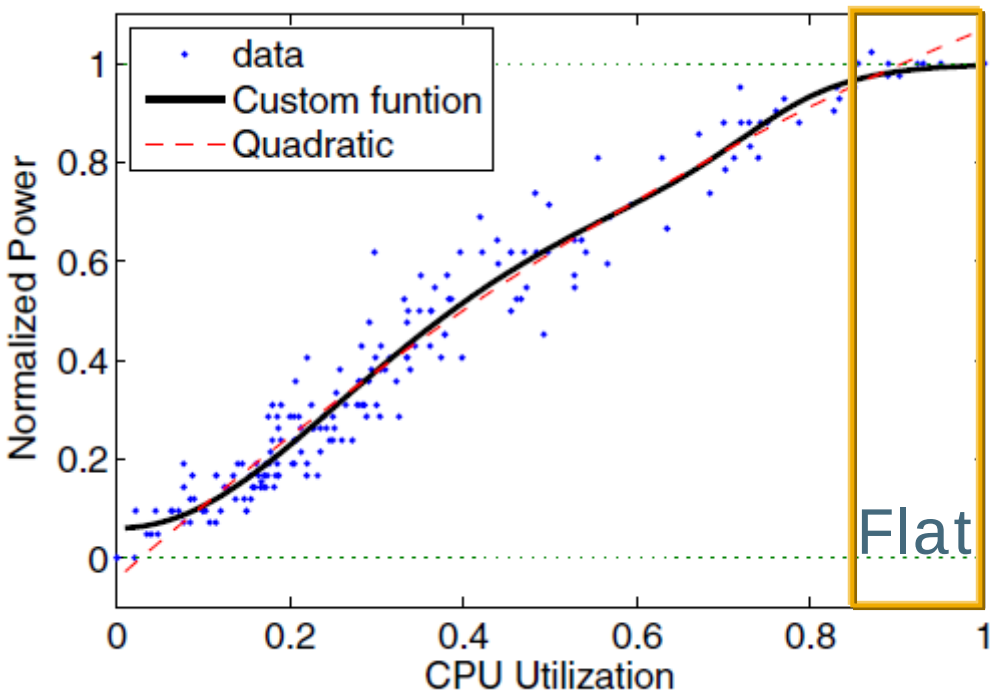
Cost/CPU: $C_{cpu} = N_{core} C_{core}$

Cost/core: $C_{core} = c_1 T_{core}^{c_2} + c_3 N_{core}^{c_4} + c_5$

model param.	c_1	c_2	c_3	c_4	c_5
TPCC/DP	36	2.0	261	-0.9	-105

Modeling Power Consumption

- Focus on hardware costs and power consumption



Normalized power:

$$P_{norm} = \frac{P_{sys} - P_{idle}}{P_{busy} - P_{idle}}$$

Butterworth-type law:

$$h(U) = c_1 U^{c_2} + c_3 U^{c_4} + c_5,$$

$$P_{norm}(U) = 1 - h(U)^{-1}$$

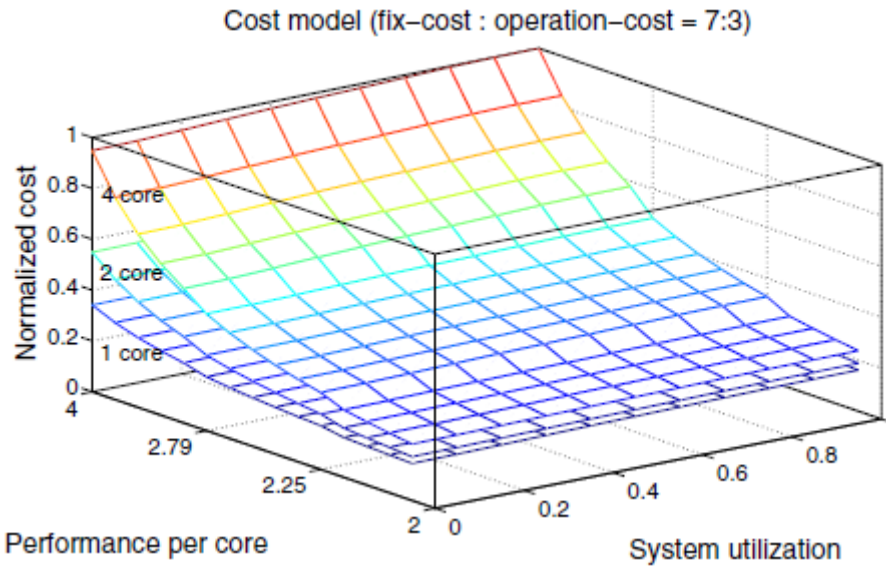
Final server cost model

$$Cost(T_{core}, N_{core}, U, I) =$$

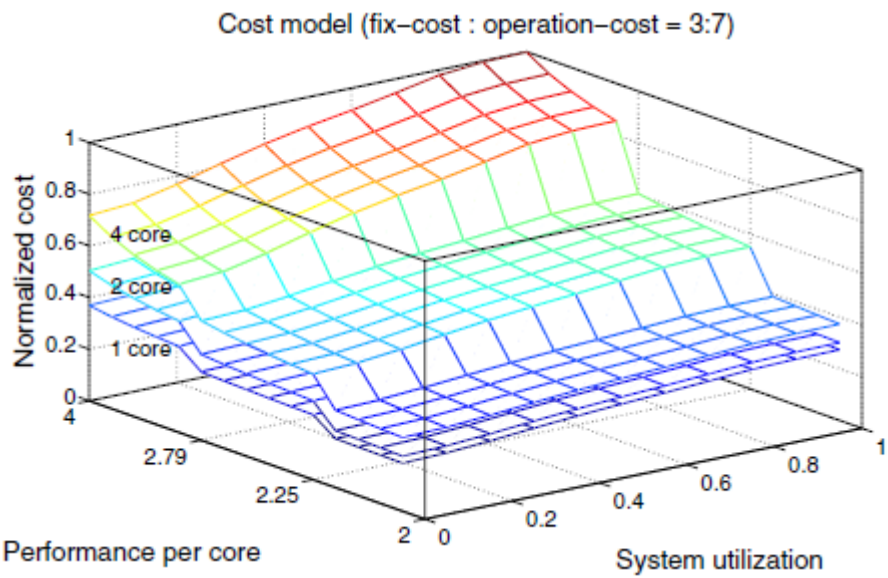
$$p_0 + p_1 C_{cpu} + p_2 \int_{t \in I} P_{sys}(U(t))dt,$$

static costs — hw costs — power. costs

Traditional data center



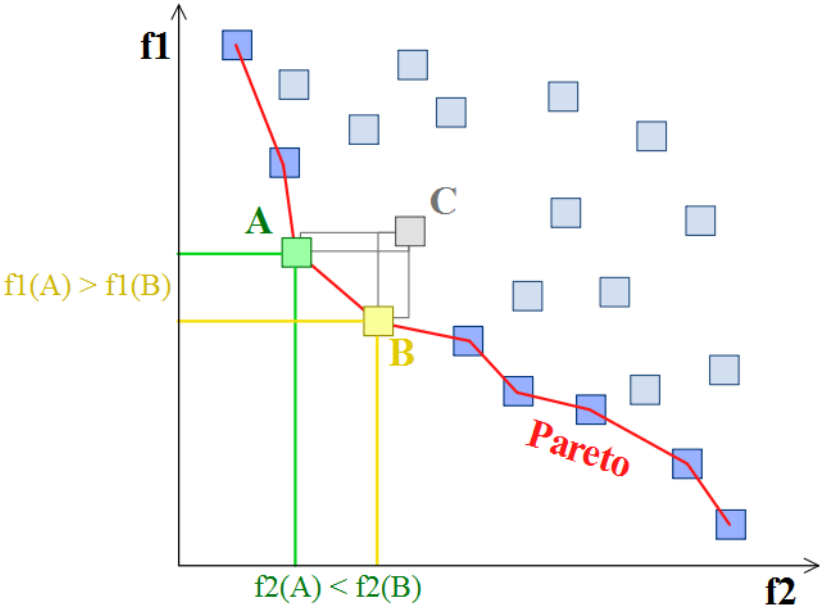
Consolidated data center



Sizing Framework

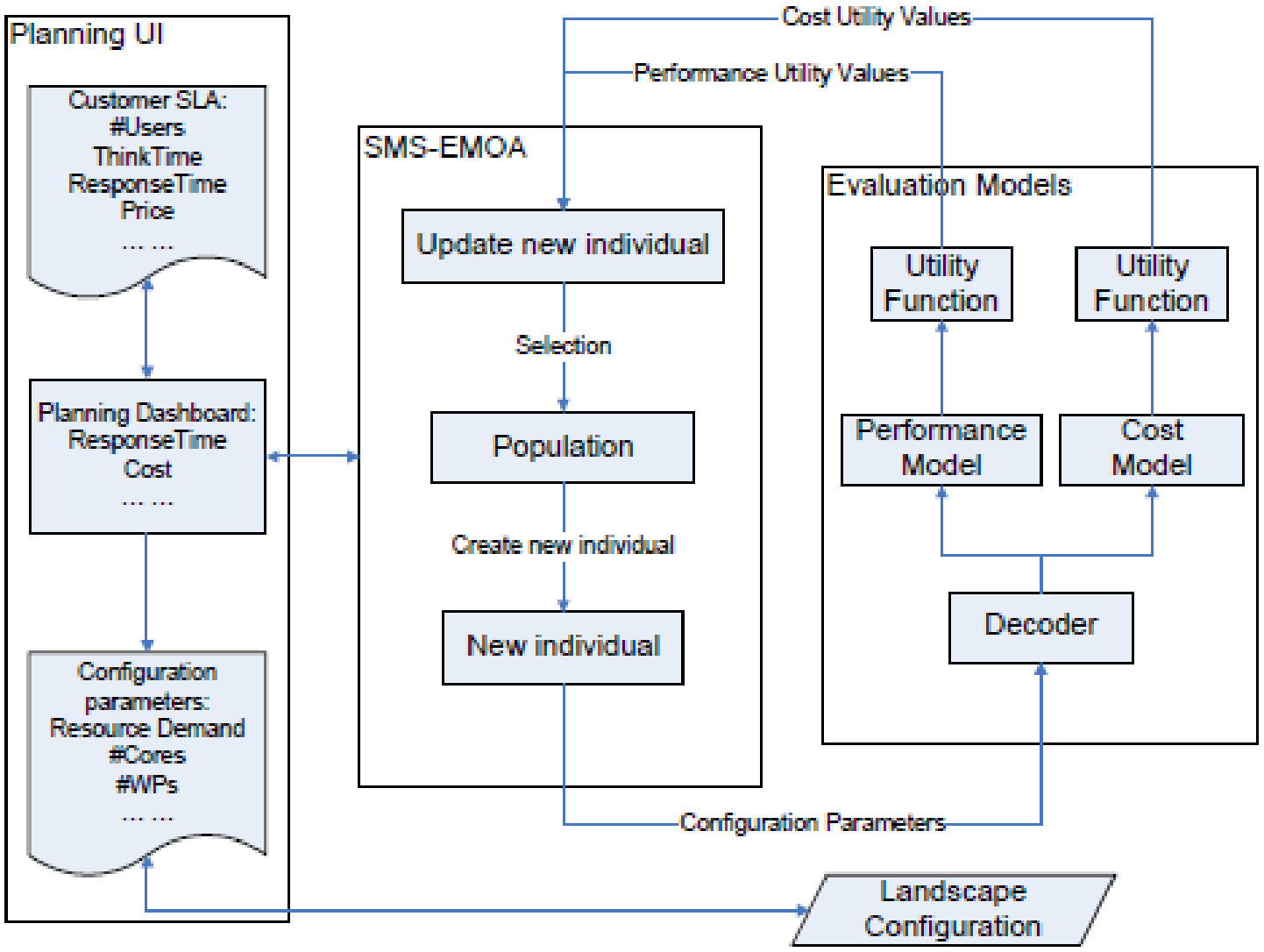
Pareto Front

- Optimization variables
  - Variables: service demands, #cores, software threading lvl
  - Objective functions: min *Cost* , min *Response Time*
- Pareto improvement
  - improve one variable without making any other worse
- Pareto-optimal point
  - no feasible improvements
  - Pareto front



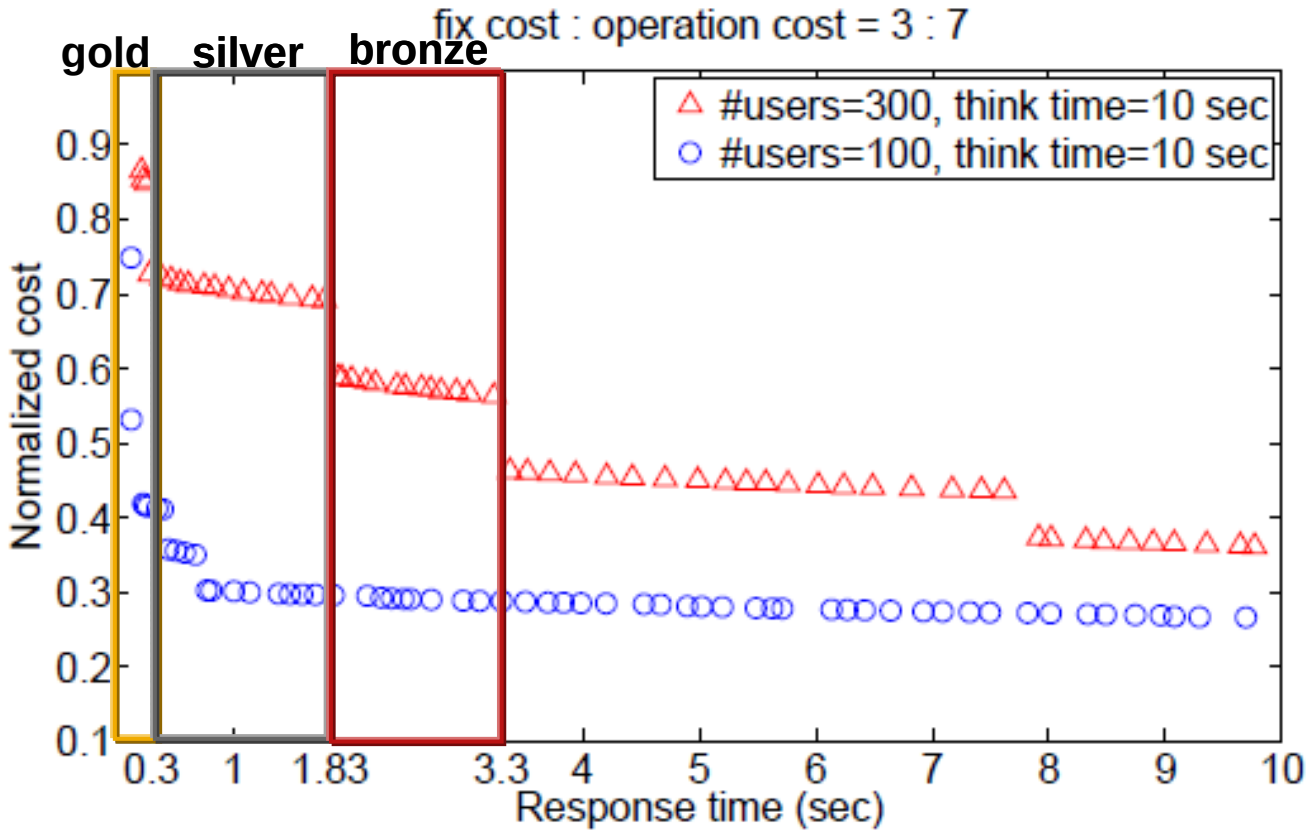
source: wikipedia

SLA Planning Framework



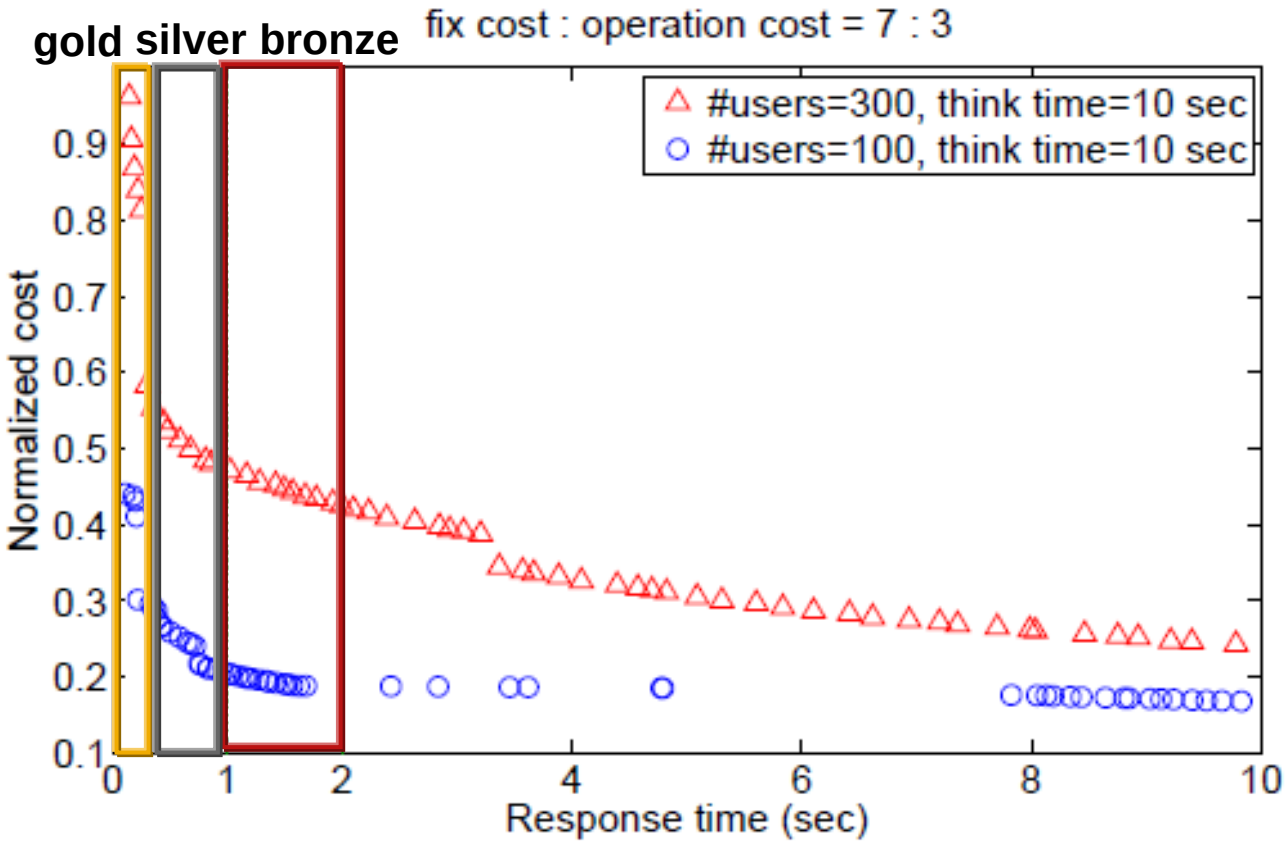
Pareto Front: Results

- Consolidated data center scenario

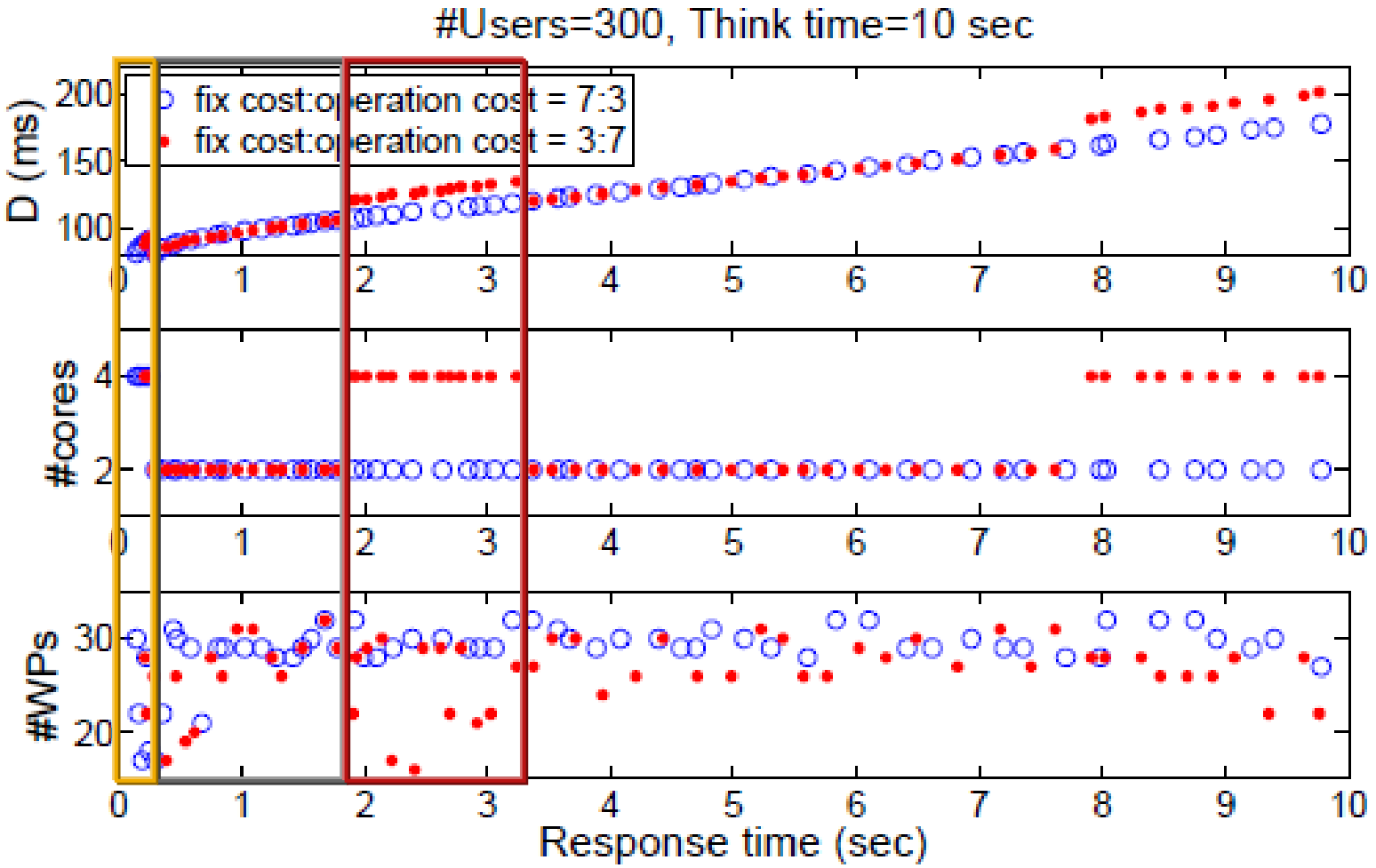


Pareto Front: Results

- Traditional data center scenario



Solution Space: Results



Conclusion

- Quantitative techniques for (semi-)automatic sizing
  - Queueing-theoretic performance model
  - TCO models
  - Multi-objective optimization for decision making

- Future work
  - Generalization of methodology to arbitrary application
  - Development of analytical approximations for FCR models
  - Validation of power models for ERP applications

Thanks!

g.casale@imperial.ac.uk