Resource Demand Modeling for Multi-Tier Services

Jerry Rolia[1], Amir Kalbasi, **Diwakar Krishnamurthy**[3], Stephen Dawson[2]

[1]: Automated Infrastructure Lab, HP Labs, Bristol, UK, e-mail: jerry.rolia@hp.com

[2]: SAP Research, CEC Belfast, UK, e-mail: firstname.lastname@sap.com

[3]: University of Calgary,Calgary, AB, Canada, e-mail: dkrishna@ucalgary.ca

LABS hp

SAP RESEARCH

UNIVERSITY OF CALGARY

Outline

- Motivation
- Related work
- Problem statement and methods
  - LSQ and LAD Regression
  - Demand Estimation with Confidence (DEC)
- Experimental case study
- Summary and conclusions

Motivation

- We consider Software as a Service (SaaS) environments
- SaaS permits scope for massive customizations
  - Different users can use different mixes of system functions
- Need to characterize performance of a customized workload
- We focus on resource demands of customized workloads
  - Inputs for analytic models used for sizing/resource management
- **Need techniques to accurately predict demands**
  - Many possible customizations - direct measurements infeasible
- Contribution – Demand Estimation with Confidence (DEC)

Related work

- Linear regression
  - Utilization and demands are linearly related (U=XD)
  - Measure utilization and execution counts of system functions
  - Get per-function demands – predict for arbitrary function mixes
  - Variants - Least Squares (LSQ), Least Absolute Deviations (LAD)
- Queuing Network Model (QNM) based approaches
  - Assume a QNM and measured response times available for a mix
  - Estimate demands such that QNM R matches measured R
  - DEC intended when QNM and measured R not available
- How does DEC compare with LSQ and LAD?

Problem statement

- Consider
  - system with *M* functions and *R* resources
  - finite number of benchmarks $B_1 \ldots B_B$
    - Benchmark – Semantically correct sequence of requests
    - Examples – TPC-W sessions, SAP SD benchmark
  - Specified custom *mix* $F = F_1 \ldots\ldots F_M$
    - F_i *is execution* count for i^{th} function

- Estimate for the specified custom workload mix
  - demands $D_1 \ldots D_R$ on *R* resources
  - confidence intervals for $D_1 \ldots D_R$

LABS hp
SAP RESEARCH
UNIVERSITY OF CALGARY

LSQ method

- Execute benchmarks $B_1 \ldots B_B$
- When benchmarks are executing, for each resource
  - Measure busy time Y_i
  - Measure observed function counts $F_{1,I} \ldots F_{M,I}$ for sampling period i
- Apply LSQ

$$Y_i = D_1 F_{1,i} + D_2 F_{2,i} \cdots + D_M F_{M,i} + E_i, i = 1,2, \cdots N$$

$$D_i \geq 0, i = 1..N$$

$$O(D_1, \cdots D_M) = \sum_i \left(Y_i - D_1 F_{1,i} - D_2 F_{2,i} \cdots - D_M F_{M,i} \right)^2$$

$$\hat{Y} = D_1 F_1 + D_2 F_2 \cdots + D_M F_M$$

Inputs

Solve for
per component demands

Estimate the overall demand for
desired workload mix at resource

LAD method

- LAD minimizes absolute error instead of sum square of errors
- More robust towards demand outliers

$$Y_i = D_1 F_{1,i} + D_2 F_{2,i} \cdots + D_M F_{M,i} + E_i, i = 1,2,\cdots N$$

$$D_i \geq 0, i = 1..N$$

$$O(D_1, \cdots D_M) = \sum_i \left| Y_i - D_1 F_{1,i} - D_2 F_{2,i} \cdots - D_M F_{M,i} \right|$$

$$\hat{Y} = D_1 F_1 + D_2 F_2 \cdots + D_M F_M$$

LAD minimizes absolute error

Notes on LSQ and LAD

- Both techniques rely on a series of assumptions
  - Linear relationship between utilization and function counts
  - Function demands are deterministic
  - Errors - normally distributed (LSQ);Laplacian distributed (LAD)
- Both techniques impacted by violation of assumptions
  - Poor demand estimates
  - Poor confidence interval estimates
- Both techniques can be impacted by *multicollinearity*
  - Execution counts of 2 or more functions are correlated
  - Observed in production systems (Pacifici et al, PEVA)
  - Can’t distinguish per-function demands under correlations

DEC

- Predicts demands for *joint* use of functions
- Consider benchmarks $B_1 \ldots B_B$ – each with its own mix
- Measure mean resource demands of each benchmark

$$D^B = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1B} \\ D_{21} & D_{22} & \cdots & D_{2B} \\ \vdots & \vdots & \cdots & \vdots \\ D_{R1} & D_{R2} & \cdots & D_{RB} \end{bmatrix}$$

- Express desired mix as linear combination *L* of a subset of benchmarks
  - Subset of *B'* benchmarks executed as per *L* yields same mix as desired mix
- Estimate demand as linear combination *L* of demands of *B'*

$$D^{B'} = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1B'} \\ D_{21} & D_{22} & \cdots & D_{2B'} \\ \vdots & \vdots & \cdots & \vdots \\ D_{R1} & D_{R2} & \cdots & D_{RB'} \end{bmatrix}$$

$$\text{where } B' \le B$$

Estimated demand

$$D^S \approx D^{B'} L$$

DEC – (cont'd)

- Example – (system with 3 functions and 1 resource)

Desired mix = [4 1 7]

$B'\ \{B_3=[2\ 0\ 3]\ B_7=[0\ 1\ 0]\ B_8=[\ 0\ 0\ 1]\}$

$D^{B'} = [2\ 5\ 1]$

$L = [2\ 1\ 1]^T\ (2^*\ B_3 + 1^*\ B_7 + 1^*\ B_8)$

$D^S = D^{B'}\ L = 10$

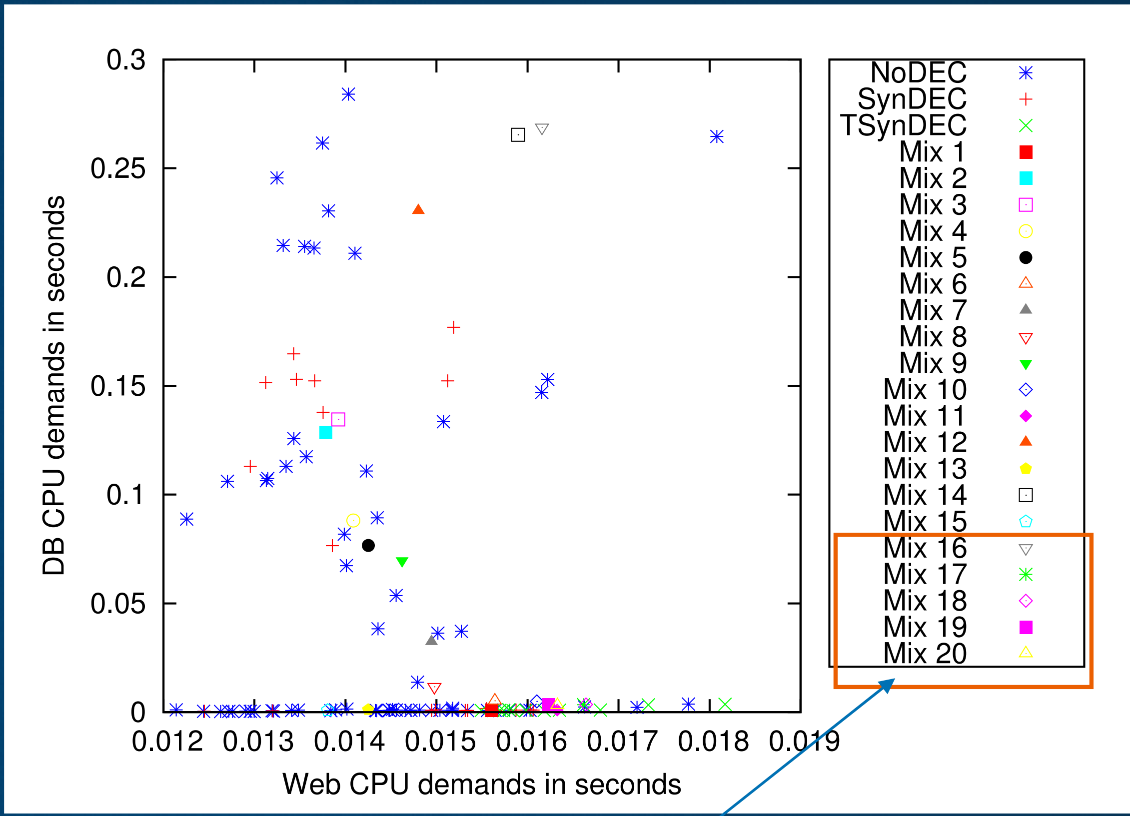
We use an iterative approach that employs linear programming to determine *B'* and *L*

DEC VS Regression

- Advantages
  - Insensitive to multicollinearity - doesn't rely on per-function demands
  - More robust confidence interval calculations
    - Mean demand of benchmarks are normally distributed under central limit theorem (assuming large number of runs)
    - It follows linear combination of mean demands is also normally distributed
  - Can prepare a validation performance test from the combination *L*
    - Execute chosen benchmarks as per L – validate demands or performance objectives of the customized workload
- Limitations
  - May not be always possible to realize exact match of mix
  - Non-unique - multiple combinations possible for a given mix

Case study

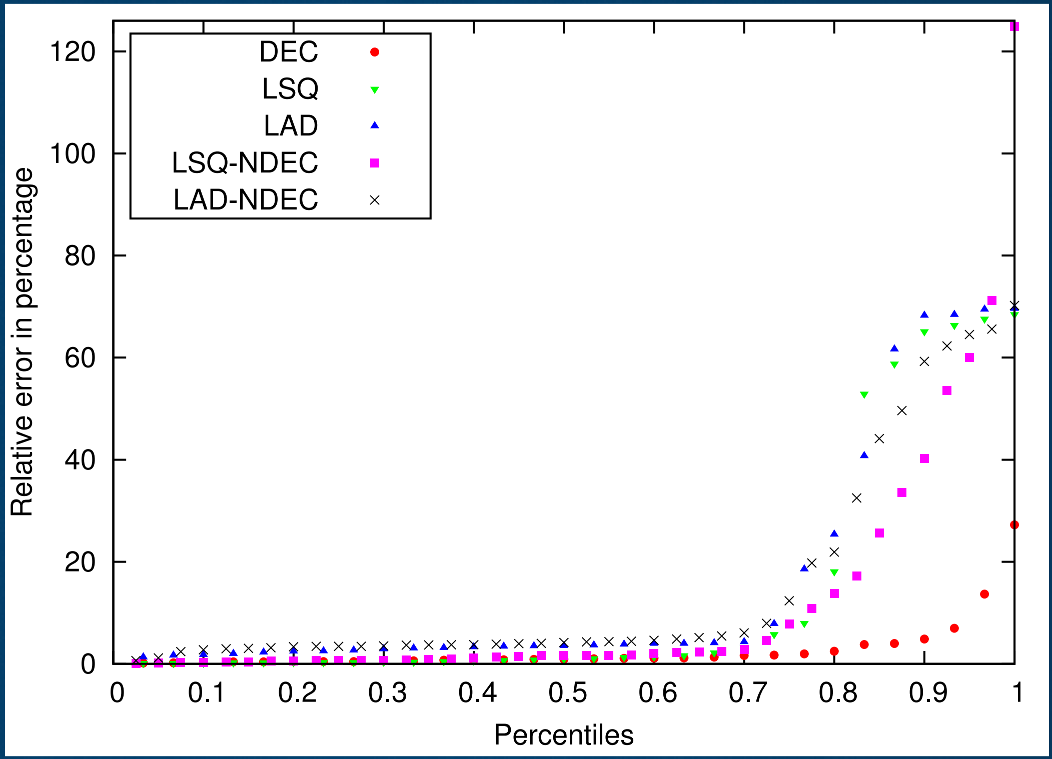
- 3-tier TPC-W system
- 100 benchmarks
- 120 “customized” mixes
- 1000x variation in D_{db}
- Compare DEC, LSQ, LAD for 120 mixes



Controlled mixes to study multicollinearity

Results – cases with exact match

Prediction errors for DB CPU demand (cases with exact match)

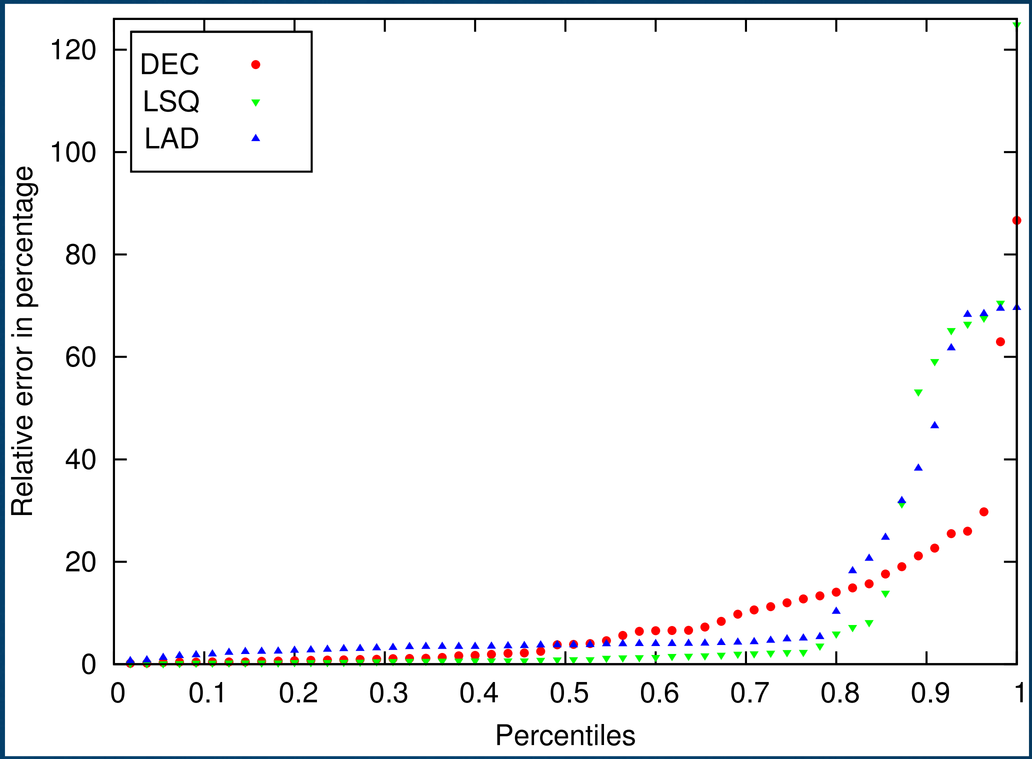


DEC outperforms LSQ and LAD

DEC achieved exact match of mix for 55 cases

Results – all cases

Prediction errors for DB CPU demand (cases with non-exact matches included)

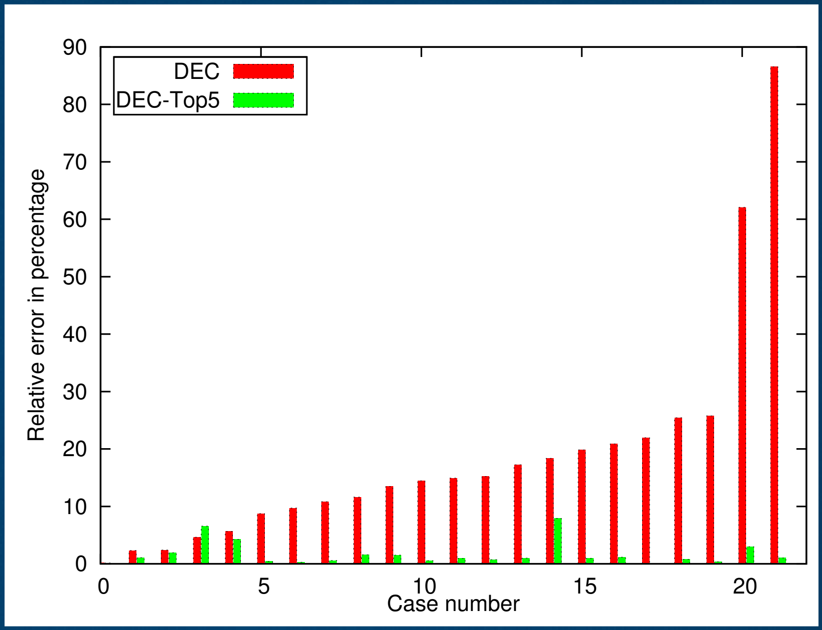


DEC predictions become less reliable

However, errors still comparable with those of LSQ and LAD

Results – exploiting flexibility of DEC to reduce errors

Prediction errors for DB CPU demand (non-exact cases)



DEC can be improved for non-exact cases by relaxing some constraints

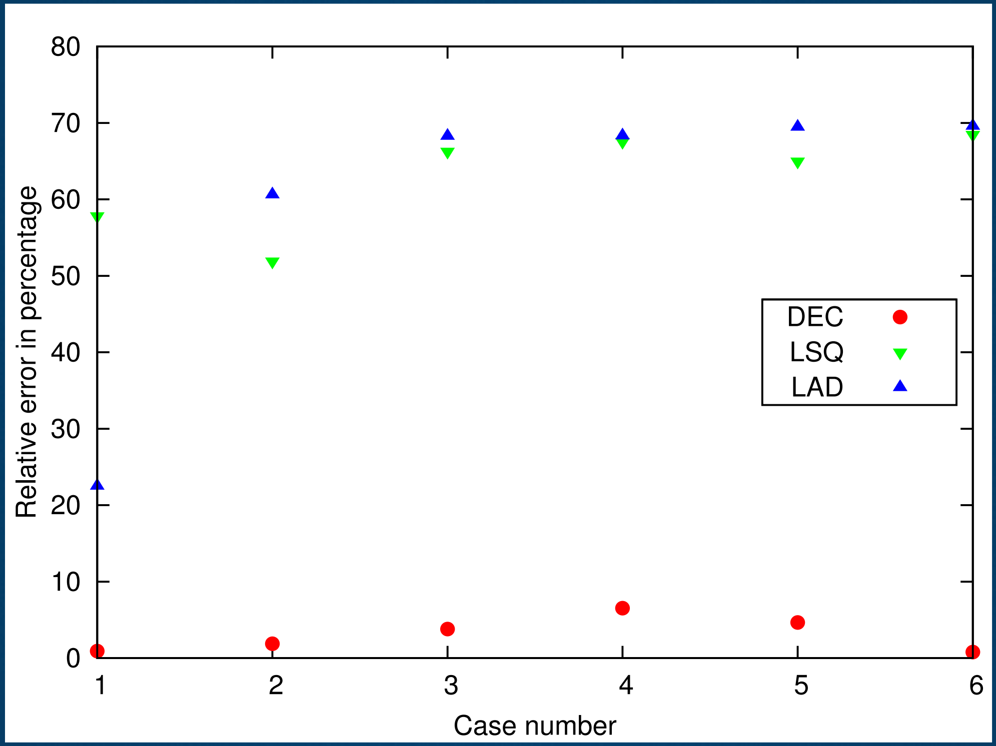
Modified LP formulation to match top 5 resource intensive functions exactly

“Best effort” match for other functions

DEC errors dropped significantly

Results - multicollinearity

Prediction errors for DB CPU demand for cases impacted by multicollinearity

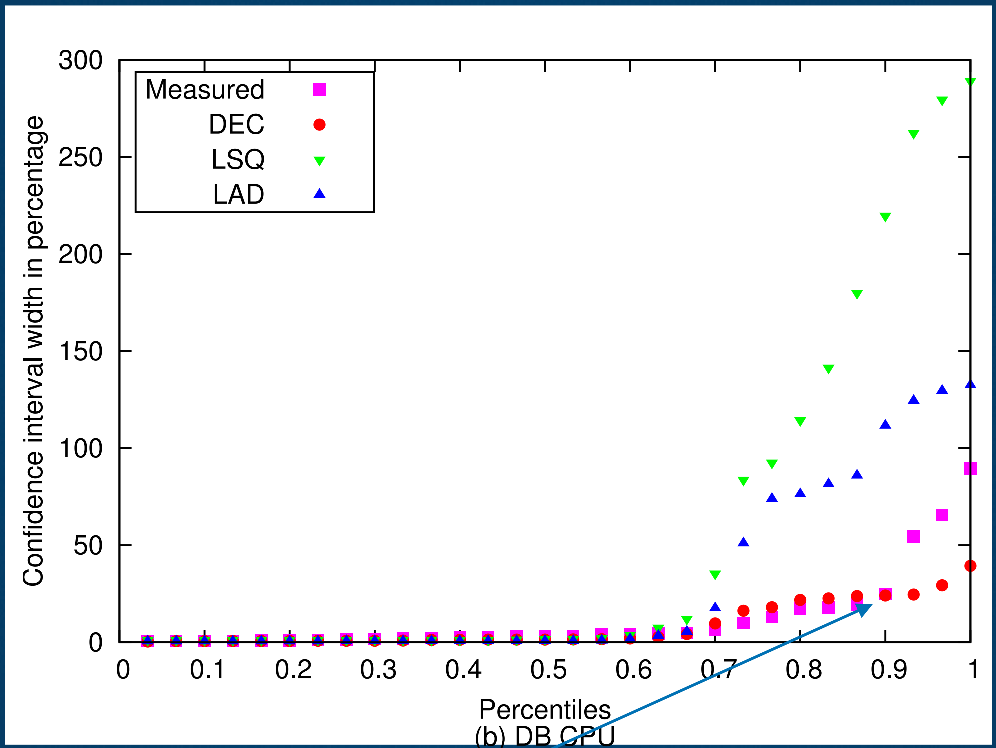


LSQ and LAD exhibit very high errors

DEC has significantly lower errors – it is not impacted by multicollinearity

Results – confidence intervals

Confidence interval width of mean demand predictions



(b) DB CPU

DEC's CI predictions closely track CIs for measured demands

Summary and conclusions

- DEC provides an alternative to regression-based demand estimation
  - Accuracy compares favorably to regression
  - Supports more robust confidence interval calculations
  - Insensitive to multicollinearity
  - Provides a performance-test based validation for predictions

- Next steps
  - Validate on other systems
  - Study impact of service demand variability in a controlled manner
  - Automate handling of cases with non-exact matches
  - Consider systems whose demands for a given mix shift with time

LABS hp

SAP RESEARCH

UNIVERSITY OF CALGARY