K-Scope:

Online Performance Tracking for Dynamic Cloud Applications

Li Zhang, **Xiaoqiao Meng, Shicong Meng, Jian Tan**

System Analysis & Optimization Group
IBM T. J. Watson Research Center

Motivation

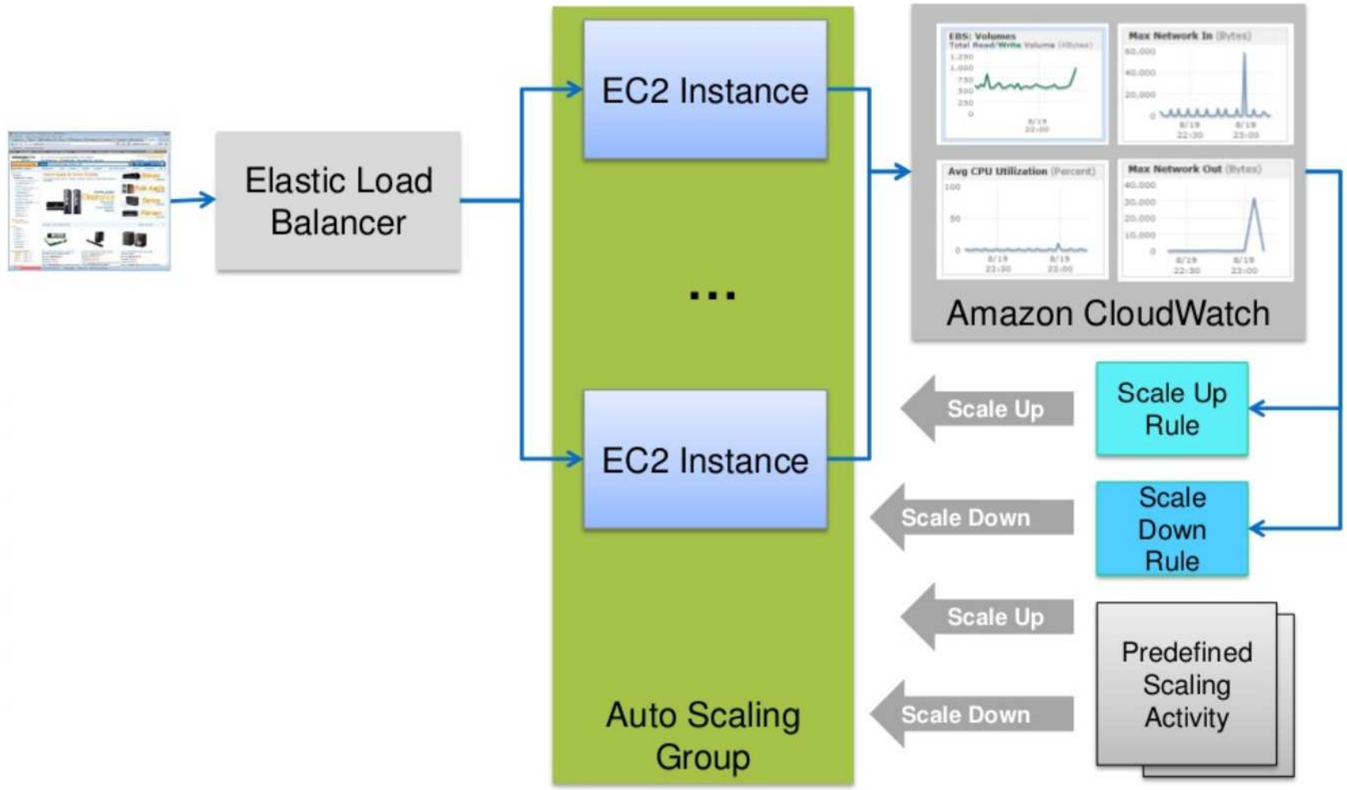
Applications in Dynamic Cloud Environment

- **Continuous delivery**
- **Shared platform services**
- **Auto scaling to satisfy SLA**

Challenges for Performance Modeling

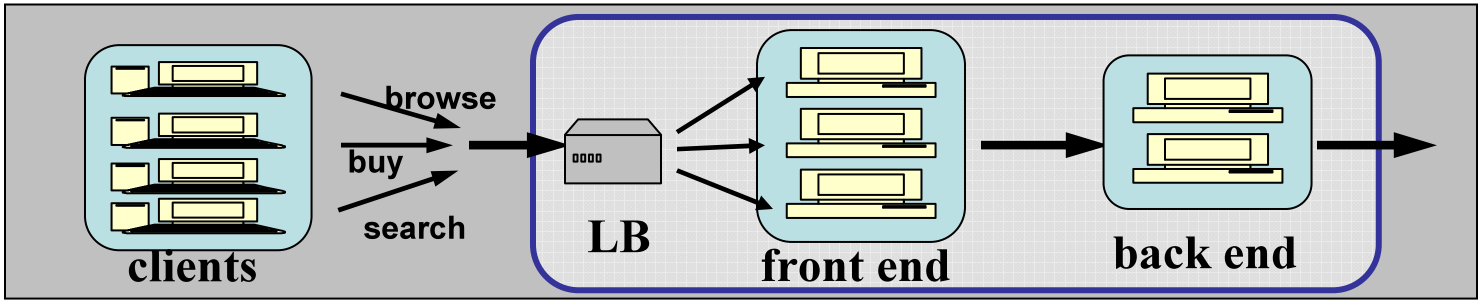
- **Continuous monitoring of appropriate metrics**
- **Changing workload and resource consumption**
  - Intensity, service (processing) time
- **Changing share of resource in Cloud**
  - CPU, IO, Network
- **Multi-tier or more complex deployment**

Auto Scaling Service



- Auto scaling allows cloud applications to scale its resource usage up and down automatically according to load and SLA
- Auto scaling requires a model that can dynamically correlate the application performance with resource assumption

Queueing Network Model



$$\lambda_i = \text{Arrival rate of class } i \text{ jobs.}$$
$$S_{ij} = \text{Average service time of class } i \text{ jobs at tier } j.$$
$$d_i = \text{Additional delay for class } i \text{ jobs in system.}$$
$$u_{0j} = \text{Background utilization for tier } j.$$
$$u_j = \text{Average utilization for tier } j.$$
$$R_i = \text{Average response time for class } i \text{ jobs in system.}$$

Under appropriate assumptions, the system performance & resource utilization can be approximated by the queueing analytic relations:

Observable

$$u_j = u_{0j} + \lambda_1 S_{1j} + \lambda_2 S_{2j} + \lambda_3 S_{3j}, \; j \in \{1,2\} \quad (1)$$
$$R_i = d_i + \frac{S_{i1}}{1-u_1} + \frac{S_{i2}}{1-u_2}, \; i \in \{1,2,3\} \quad (2)$$

Unknown

In vector form: $\mathbf{z} := (u_1, u_2, R_1, R_2, R_3)^T = \mathbf{h}(\mathbf{x}).$

Kalman Filter Dynamics

- **Estimate values of hidden state variables of a dynamic system excited by stochastic disturbances and stochastic measurement noise.**

$$\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t-1)+\mathbf{w}(t) = \mathbf{x}(t-1)+\mathbf{w}(t), \quad (4)$$
$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t-1)+\mathbf{v}(t). \quad (5)$$

$$\mathbf{x} = (u_{01}, u_{02}, d_1, d_2, d_3, S_{11}, S_{21}, S_{31}, S_{12}, S_{22}, S_{32})^T \quad (3)$$

- **Variables:**
  - **x(t): State variable that is not observed**
  - **F(t): State transition model**
  - **w(t): Process noise (zero mean, multivariate Gaussian)**
  - **z(t): Measurement vector**
  - **H(t): Observation model, maps true state into observation space**
  - **v(t): Observation noise (zero mean, multivariate Gaussian)**

Kalman Filter Algorithm

Predict:

$$\hat{\mathbf{x}}(t|t-1) = \mathbf{F}(t)\hat{\mathbf{x}}(t-1|t-1) \quad (6)$$

$$\mathbf{P}(t|t-1) = \mathbf{F}(t)\mathbf{P}(t-1|t-1)\mathbf{F}^T(t) + \mathscr{Q}(t) \quad (7)$$

Update:

$$\mathbf{H}(t) = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right](\hat{\mathbf{x}}(t|t-1)) \quad (8)$$

$$\mathbf{S}(t) = \mathbf{H}(t)\mathbf{P}(t|t-1)\mathbf{H}^T(t) + \mathscr{R}(t) \quad (9)$$

$$\mathbf{K}(t) = \mathbf{P}(t|t-1)\mathbf{H}^T(t)\mathbf{S}^{-1}(t) \quad (10)$$

$$\hat{\mathbf{x}}(t|t) = \hat{\mathbf{x}}(t|t-1) + \mathbf{K}(t)(\mathbf{z}(t) - \mathbf{h}(\hat{\mathbf{x}}(t|t-1))) \quad (11)$$

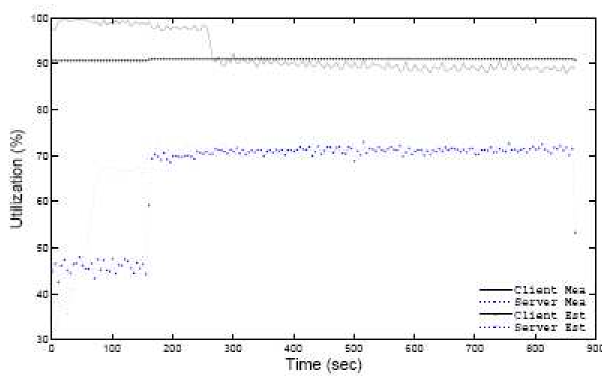
$$\mathbf{P}(t|t) = (\mathbf{I} - \mathbf{K}(t)\mathbf{H}(t))\mathbf{P}(t|t-1) \quad (12)$$

- **Apply Predict & Update iteratively over time**

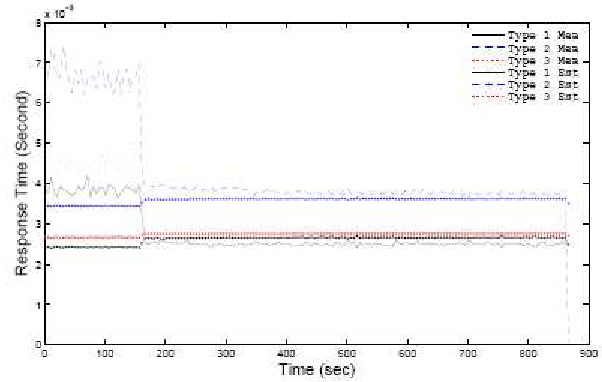
  Adapt to changing service times x(t) & observations z(t)

SOABench Experiment

Model Fitting

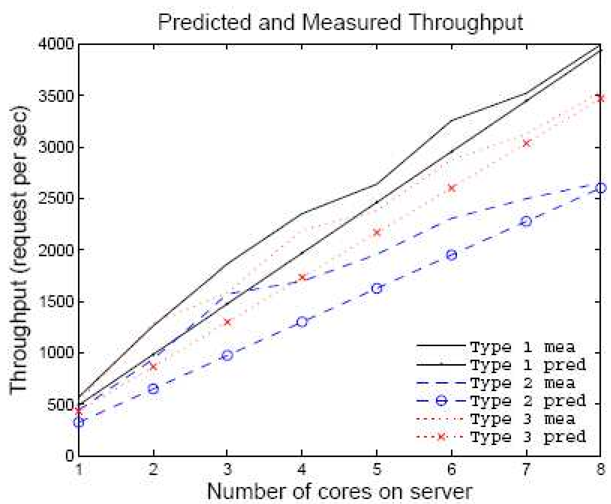


(a) CPU utilizations

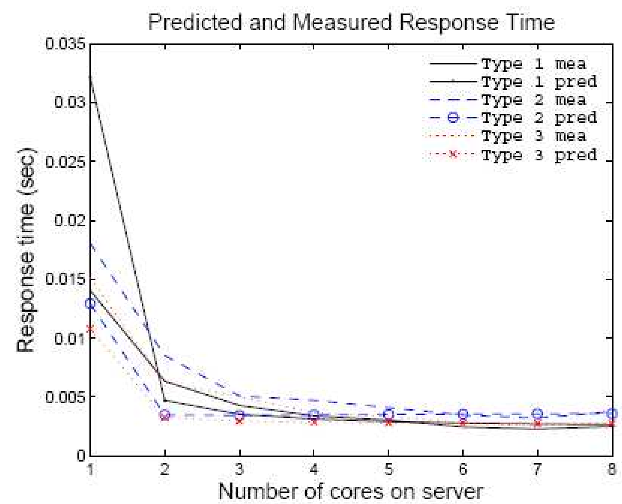


(b) Response time

Prediction



(a) Predicted throughput when reducing CPU cores on server



(b) Predicting response time when reducing CPU cores on server

Conclusion

Approach

- Queueing network based model to quantify performance
- Model based capacity planning, problem identification …

Key problem

- Inference formulation to find best fit parameters
- Kalman filter for online parameter inference

Extensive Experiments

- Validate the quality of the solution
- Apply to real scenarios