

Filling the Gap: A Tool to Automate Parameter Estimation for Software Performance Models

Weikun Wang, Juan F. Pérez, Giuliano Casale
Department of Computing
Imperial College London
weikun.wang11@imperial.ac.uk

September 1, 2015

Motivation

- ▶ DevOps - a recent trend in Software engineering
- ▶ Bridges the gap between software development and operations
- ▶ Use performance models for QoS analysis
- ▶ Accurate parametrization is challenging

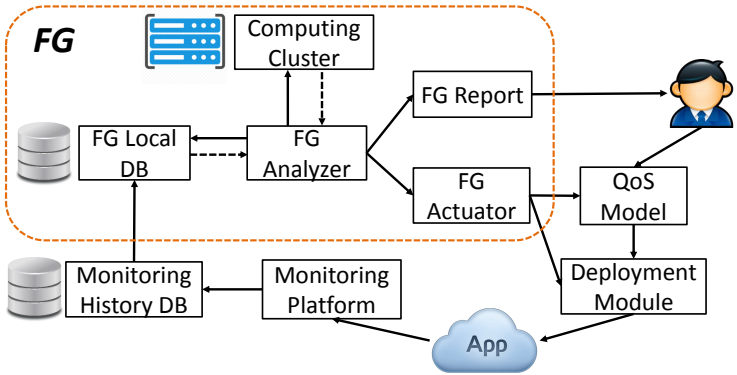
FG tool

- ▶ Continuous performance model parametrization
- ▶ Advanced estimation algorithms
- ▶ Statistical inference from monitoring data
- ▶ Application QoS report generation

FG Components

- ▶ FG Local DB: monitoring data storage
- ▶ FG Analyzer: statistical analysis
- ▶ FG Actuator: performance model update
- ▶ FG Reporter: application performance report

FG Architecture



Resource Demand

Definition: the cumulative execution time a request seizes from a server, excluding contention

- ▶ An important parameter of queueing models
- ▶ Difficult to obtain directly
- ▶ Extensive monitoring poses overhead

Supported Demand Estimation Algorithms

- ▶ Complete Information (CI)
- ▶ Gibbs sampling with Queue Lengths (GQL)
- ▶ MINPS/FMLPS
- ▶ Extended Regression-Based approach (ERPS)
- ▶ FCFS
- ▶ Utilization-Based Regression/Optimization (UBR/UBO)

CI and GQL

CI

- ▶ Uses full trace: ts. of arrivals and departures
- ▶ Poses additional overhead for intensive monitoring

GQL

- ▶ Requires queue length samples, i.e. number of requests at the server
- ▶ Estimates demand with Bayes' theorem
- ▶ Uses Gibbs sampling to obtain demand

MINPS/FMLPS

MINPS/FMLPS

- ▶ MINPS: a maximum likelihood (ML) method
- ▶ FMPLS: ML method with fluid approximation
- ▶ Both requires response times and queue lengths (arrival)

ERPS and FCFS

ERPS

- ▶ Requires response time and queue length (arrival)
- ▶ Linear regression to obtain demand

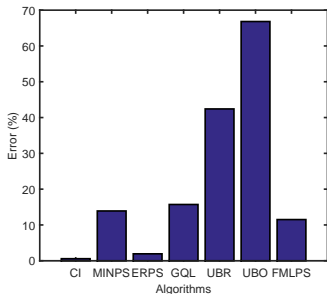
FCFS

- ▶ Estimation for FCFS servers
- ▶ Requires response time and queue length (arrival)
- ▶ Linear regression to obtain demand

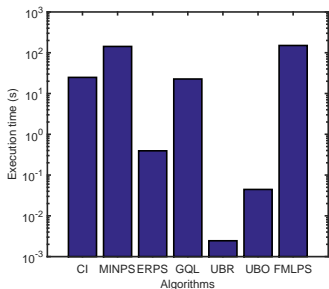
Summary: required monitoring data

Data Required	Algorithm
Full trace	CI
Utilization	UBR
Throughput	UBO
Queue length	GQL
Response Times	MINPS
Queue length (arrival)	ERPS FCFS

Comparison between Demand Estimation Algorithms



(a) Error (%)



(b) Execution time (s)

Most accurate: CI Fastest: UBR

Discussions

- ▶ i) The feedback for different demand estimation algorithms: e.g.
 - ▶ How much monitoring information can be timely brought to the developers?
 - ▶ Which metric is the easiest or most readily available?
 - ▶ Which metric poses the least overhead?
- ▶ ii) How to correlate the resource IDs as well as the request types for inconsistent design time model/deployment model/monitoring data?