**QoS-Aware Cluster Management in Heterogeneous Datacenters**
Christina Delimitrou and Christos Kozyrakis
Stanford University

### Application Scheduling

**Paragon: QoS-aware DC Scheduling [ASPLOS13]**

- Problem: Scheduling in large cloud providers (e.g., Amazon EC2, Windows Azure, Google Compute Engine, vSphere)

- Challenges:
  1. **Unknown applications** → no a priori assumptions
  2. Workload interference → performance loss when high workloads are shared
  3. Server *heterogeneity* → loss when running on wrong servers
  4. Cannot afford detailed profiling → high overheads

- **Insight:** Leverage the system's knowledge on previously-scheduled applications → fast and accurate app classification

**Paragon: Heterogeneity and Interference-aware DC Scheduler**

- Similar to an online recommendation system (e.g., Netflix)
- **QoS-aware:** minimize interference from co-scheduled apps
- **Scalable & lightweight:** scales to 10,000s apps & servers
- **App agnostic:** no assumptions on app behavior

1. **Gain**
   - 178 apps 40 servers
   - 5,000 apps 1,000 EC2 servers

2. 47% higher utilization (without QoS violations)
3. More balanced utilization across servers
4. Shorter scenario execution time \(+\) per-app QoS guarantees

### Cluster Management

**Quasar: QoS-aware Cluster Management**

- Cluster manager: Orchestrates DC operation
  - Where are applications scheduled? → **Paragon**
  - How many resources are allocated? → Mesos, Cloudscale, ...
  - When are apps scheduled? → priorities, admission control, ...

- **Naive approach:** Stitch together a resource allocation and a scheduling system → cluster manager

- **Problem:** Resource quantity & resource quality are dependent
  - allocation and scheduling should happen jointly to guarantee QoS & increase utilization

**Quasar:** Cluster management system that performs coordinated resource allocation and scheduling

- Considers both resource quantity (amount of resources) & quality (type of resources)
- Shifts from reservation-centric to performance-centric approach
- Leverages robust classification techniques to quickly classify an app for resource quantity & quality
- Organizes classification in classification layers to avoid exponential state space explosion
- Monitors & adapts allocation at runtime
- Applicable in: distributed frameworks, latency-critical online services, DBaaS, conventional single-node apps.

**Gain**

**EC2 scenario:** 2000 apps 200 servers
**Memcached + best-effort apps**

### Admission Control

**ARQ: Application-aware Admission Control**

- When is admission control needed?
  1. Systems can become oversubscribed → determine which applications run when resources are scarce
  2. Some apps may have relaxed QoS constraints → promote apps with strict QoS guarantees

**ARQ: Resource Quality-Aware Admission Control Protocol**

- Multi-class queueing network → easy-to-satisfy apps not blocked behind demanding workloads
- Guarantee QoS → diverge apps to other queues to preserve performance
- Guarantee stability under different app arrival distributions
- Exits the oversubscribed phase faster

**Statistical analysis of per-server pool freed times → decide switching time between queues**

8500 apps, 1000 EC2 servers
99% of workloads → less than 10% degradation

### Future Work

- Couple Quasar with isolation & partitioning schemes → performance isolation → higher utilization
- Enable OS policies geared towards latency-critical apps (usec granularity) → tail latency QoS for interactive online services
- Implications of resource-efficient cluster management in cloud pricing
- Implications in fairness, priorities, ...
- Classification-aided app development → resource-efficient software