EECS 750: Advanced Operating Systems

2/14 /2014

Heechul Yun

Administrative

- **Project proposal**
  - Deadline is **extended to Feb 22**
  - Discuss with me about your project idea before the deadline
- Next summary assignment due: 11:59 p.m., Sunday
  - Email subject line: [EECS750] Summary: Paper name
- Class presentation
  - Email subject line: [EECS750] Presentation: Paper name
  - Don't need to write a summary for the paper you present
    - If you present one paper this week, you only need to write 1 additional summary. If you present two, 0 summary is required.

Recap: Paragon

- Recommendation systems
  - *Learn each user's taste* in something (e.g., movies, books,…) and *recommend a few items* out of many
  - Netflix (movie), Amazon (books, music, products)

Recommendation System

- Key idea
  - If we know each **movie's features**
    - E.g.) Titanic: action=0.2, romance=0.5, …
  - And each **user's preference**
    - E.g.) Alice: action=5, romance=3
  - A movie's **rating** for an user can be **estimated**
    - Rating(Titanic, Alice) = 0.2*5 + 0.5*3 = 2.5

Recommendation System

Movie's feature

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9
User's Preference romance	5	1	2	2		
action	3	5	2	2		

Table source: Adopted from Machine Learning at Coursera by Andrew Ng of Stanford

Collaborative Filtering

- Automatically assign values for movie features and user preferences
  - Given movie features and movie ratings
    estimate *user preferences*
  - Given user preferences and movie ratings
    estimate *movie features*

Classification for Heterogeneity

The Netflix Challenge	Platform Classification
Recommend **movies** to **users**	Recommend **platforms** to **apps**
Utility matrix rows → **users**	Utility matrix rows → **apps**
Utility matrix columns → **movies**	Utility matrix columns → **platforms**
Utility matrix elements → **movie ratings**	Utility matrix elements → **app scores**

- Offline mode
  - Profile a few apps (20-30) across the different configurations
  - Assign performance scores per run (IPS, QPS, other system metric)
- Online mode
  - For each new app, run briefly on two platforms (1min)
  - Assign performance scores
  - Derive missing entries & identify similarities between apps

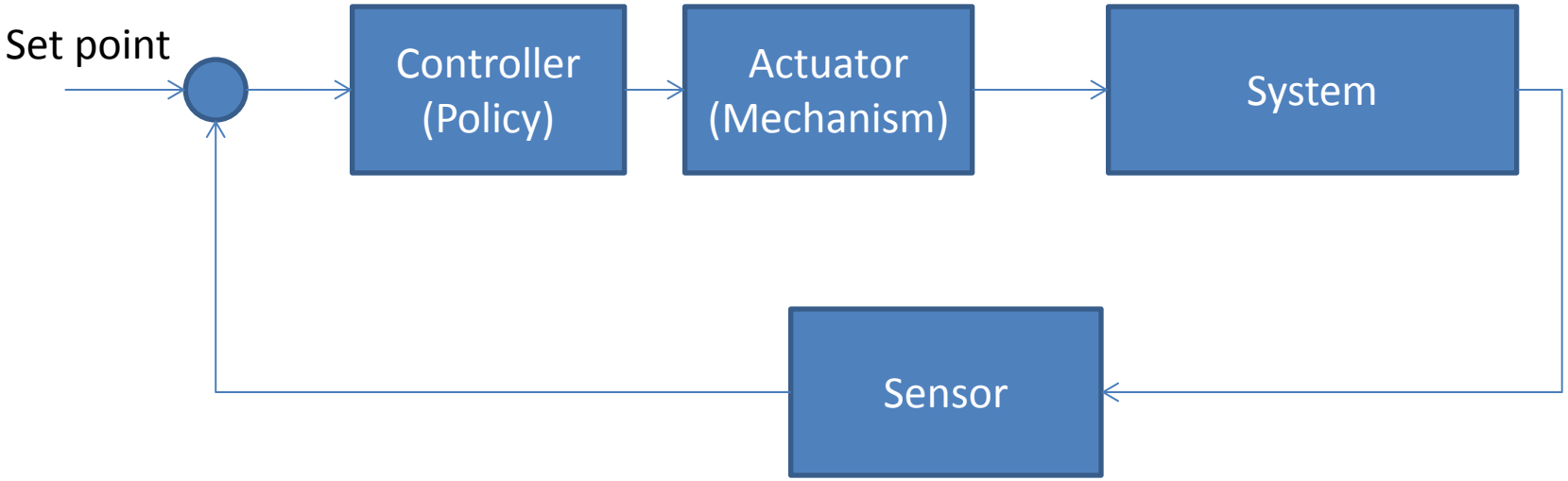
13

Slide source: http://csl.stanford.edu/~christos/publications/2013.paragon.asplos.slides.pdf

Today

- Topic: Cloud level scheduling 3
- Some background
- Q-clouds: managing performance interference effects for qos-aware clouds, EUROSYS'10

Feedback Control Loop



Comparison

- CPI^2
  - Runtime monitoring and troubleshooting
- Paragon
  - Preventing (minimizing) troubles via scheduling
- **Q-clouds**
  - Runtime monitoring and **controlling**

Today's Paper

- Q-clouds: managing performance interference effects for qos-aware clouds, EUROSYS'10