Adaptive Weighing Designs for Keyword Value Computation

John W. Byers
Computer Science Dept.
Boston University &
Adverplex Inc

Michael Mitzenmacher
School of Eng.& Appl. Sci.
Harvard University

Georgios Zervas
Computer Science Dept.
Boston University &
Adverplex Inc

Keyword Value Per Click (VPC) Estimation

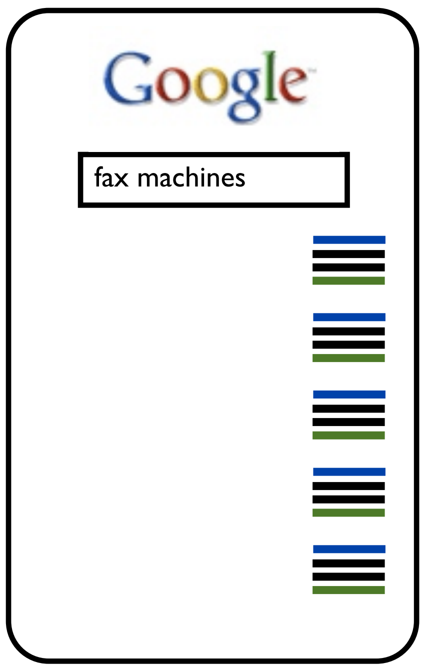
$$\text{Keyword VPC} = \frac{\text{Total keyword revenue}}{\text{Total keyword clicks}}$$

VPC is a foundational quantity in online advertising used to calibrate keyword bids

Highly competitive environment,
a small VPC error can make or break a campaign.

Channelization Setting

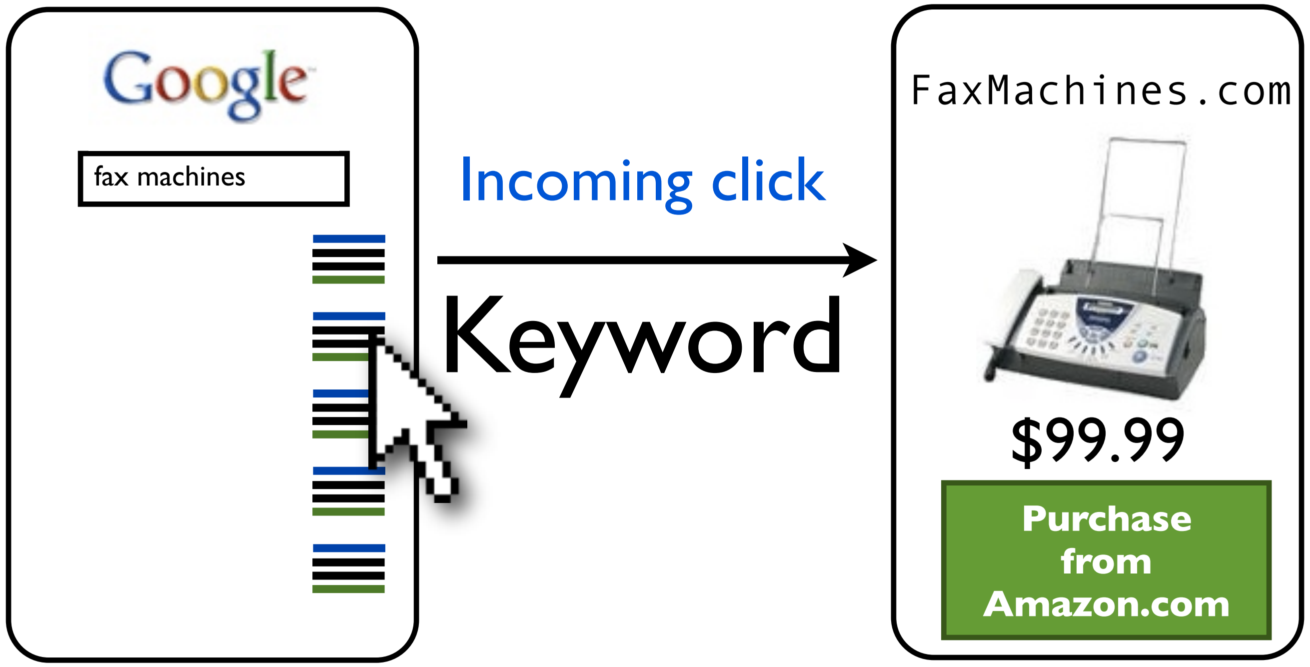
User performs query for
“fax machines”



Channelization Setting

User clicks
on paid ad

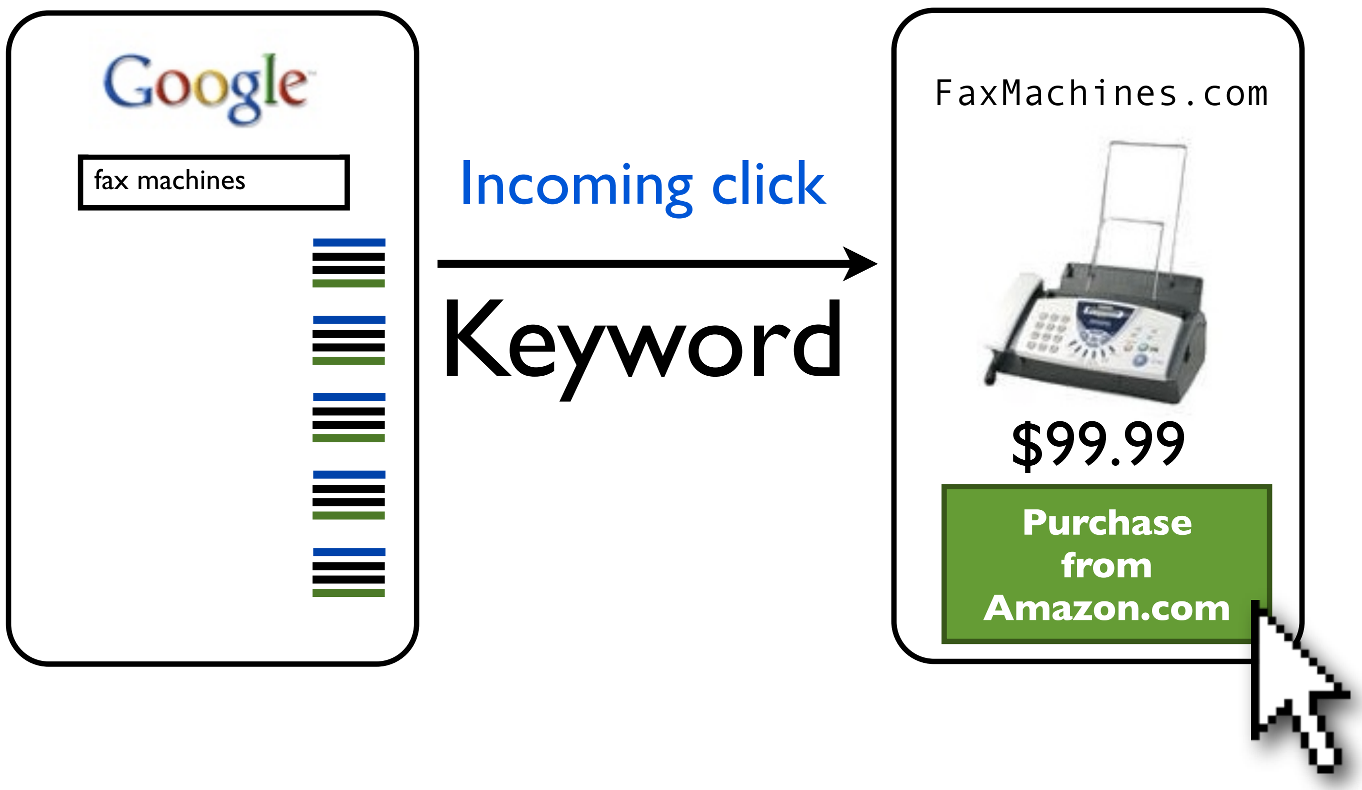
Advertiser charged per click



Channelization Setting

User clicks
on paid ad

Advertiser charged per click



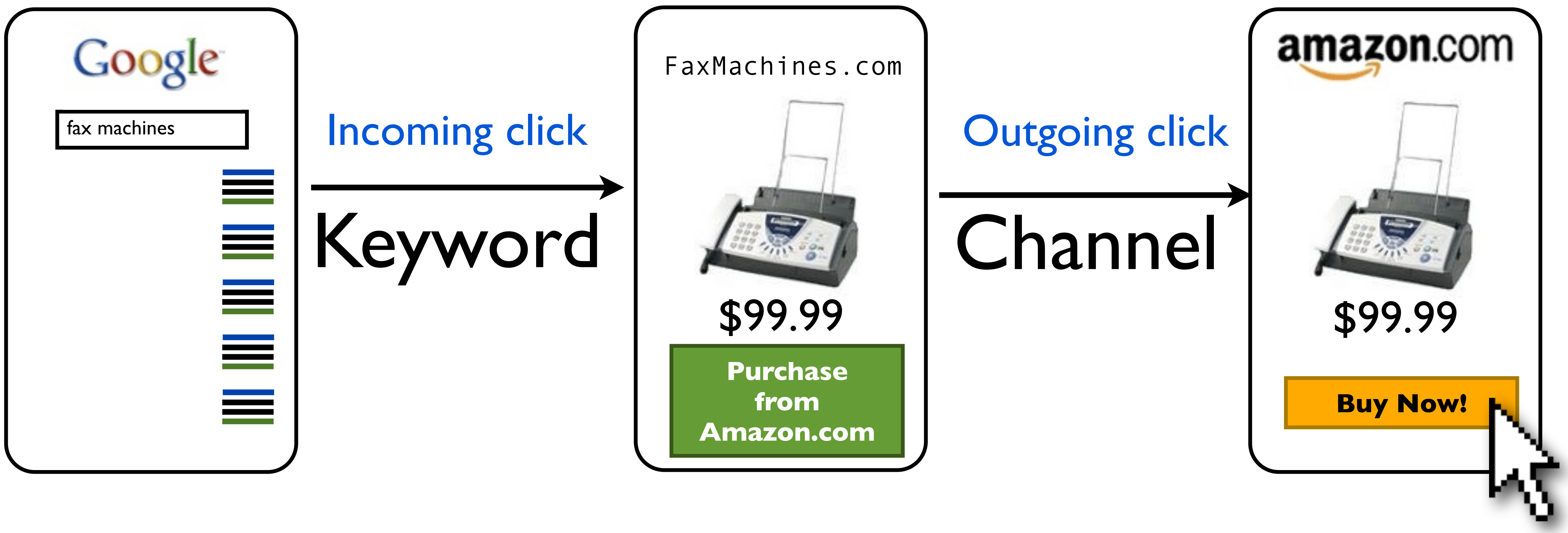
Channelization Setting

User clicks
on paid ad

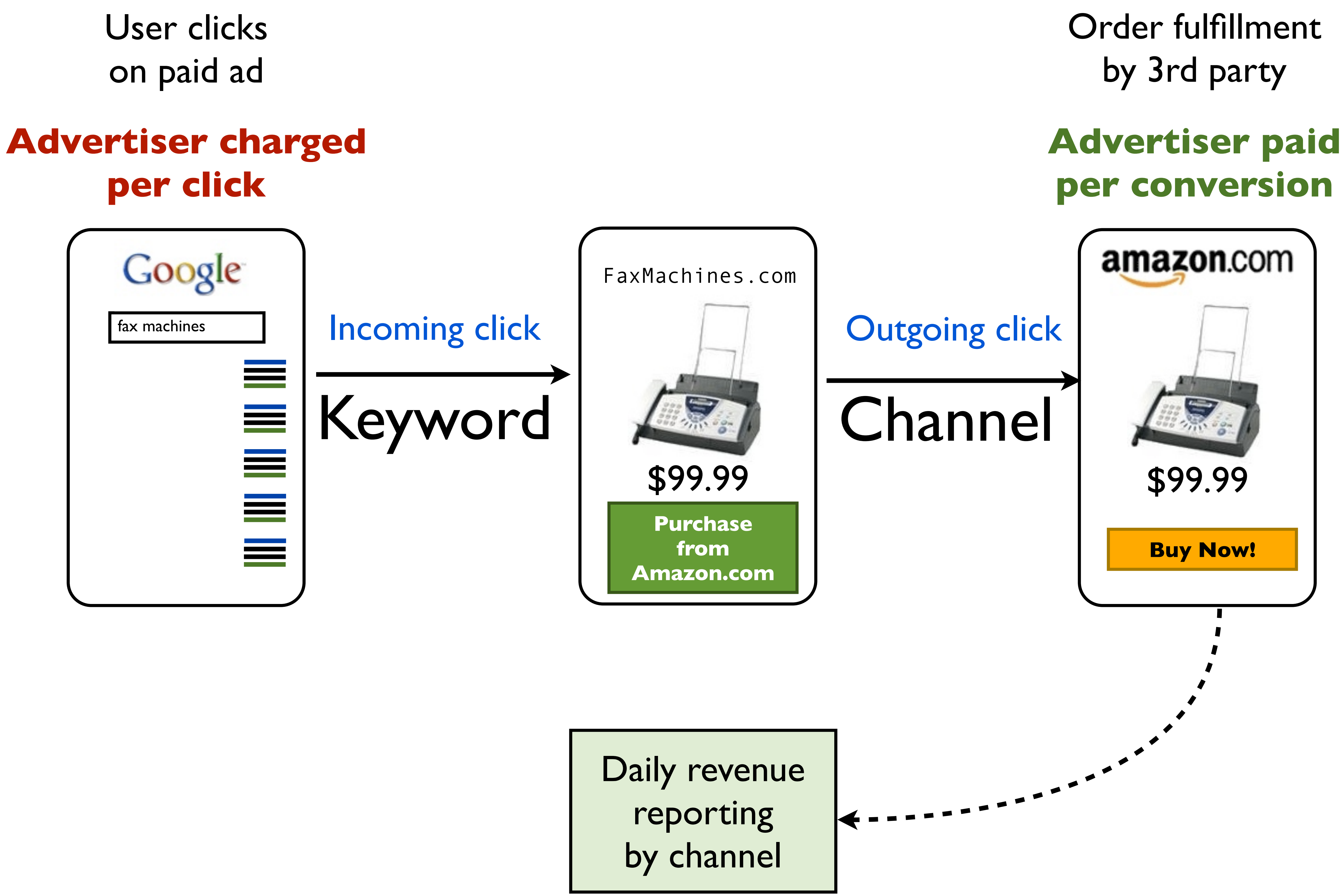
Order fulfillment
by 3rd party

Advertiser charged per click

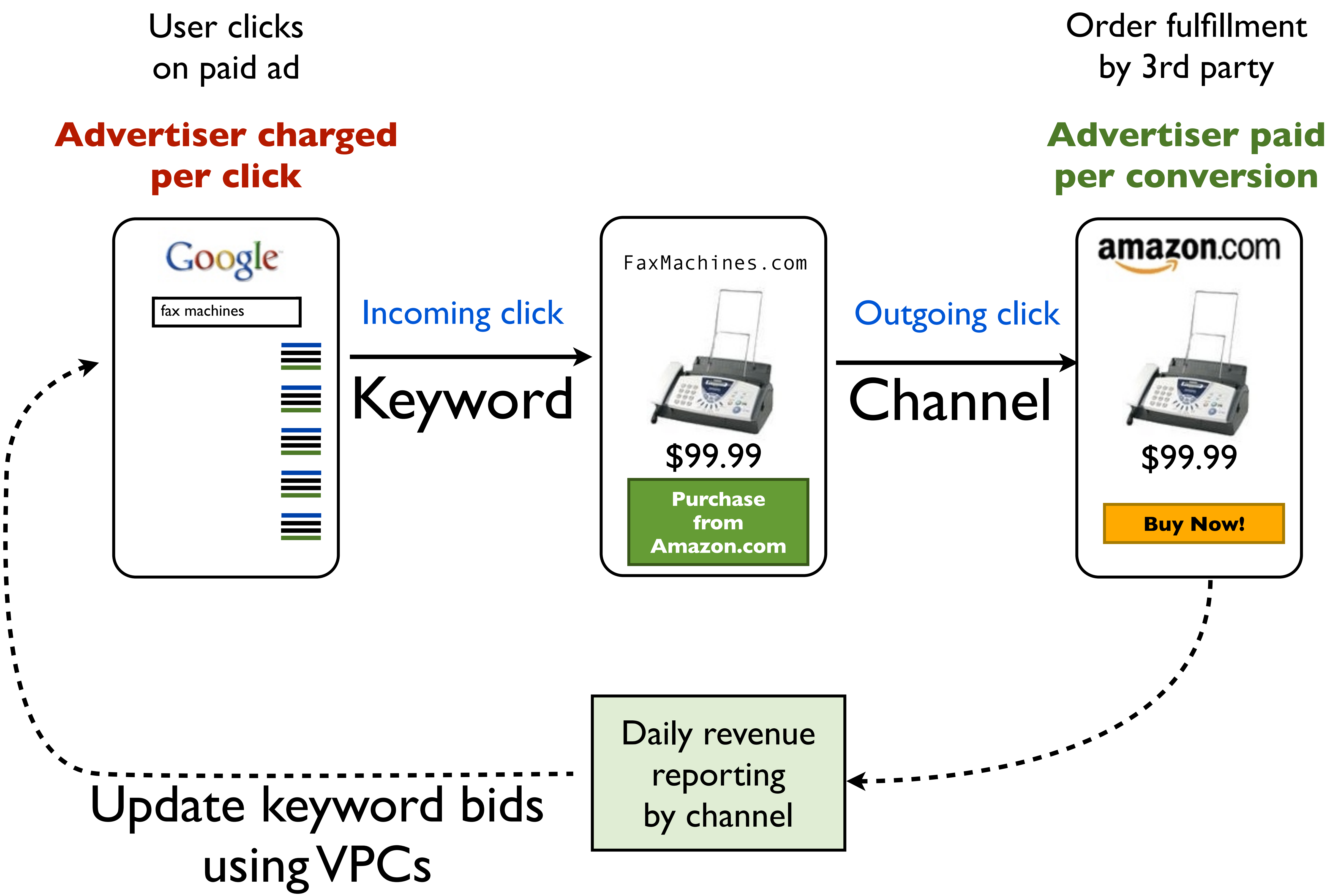
Advertiser paid per conversion



Channelization Setting



Channelization Setting



Channelization Setting

Abundance of keywords **vs** Scarcity of channels

Revenue from many keywords has to be bundled within the same channel

Channels

Each outgoing click is associated with a channel selected by the advertiser.

A channel is a place to aggregate keyword revenue.

Why is this important?

3rd-party revenue reporting is broken down by channel

Problem Statement

- **Goal:** Obtain accurate VPC estimates per keyword
- We have daily channel measurements
  - **Known:** clicks per keyword, revenue per channel
  - **Unknown:** keyword VPCs
  - |keywords| >> |channels|
- "Abusing" channels
  - keywords can be remapped to channels each day

Disaggregating aggregate data

- Regression methods are well-established
- Our contributions

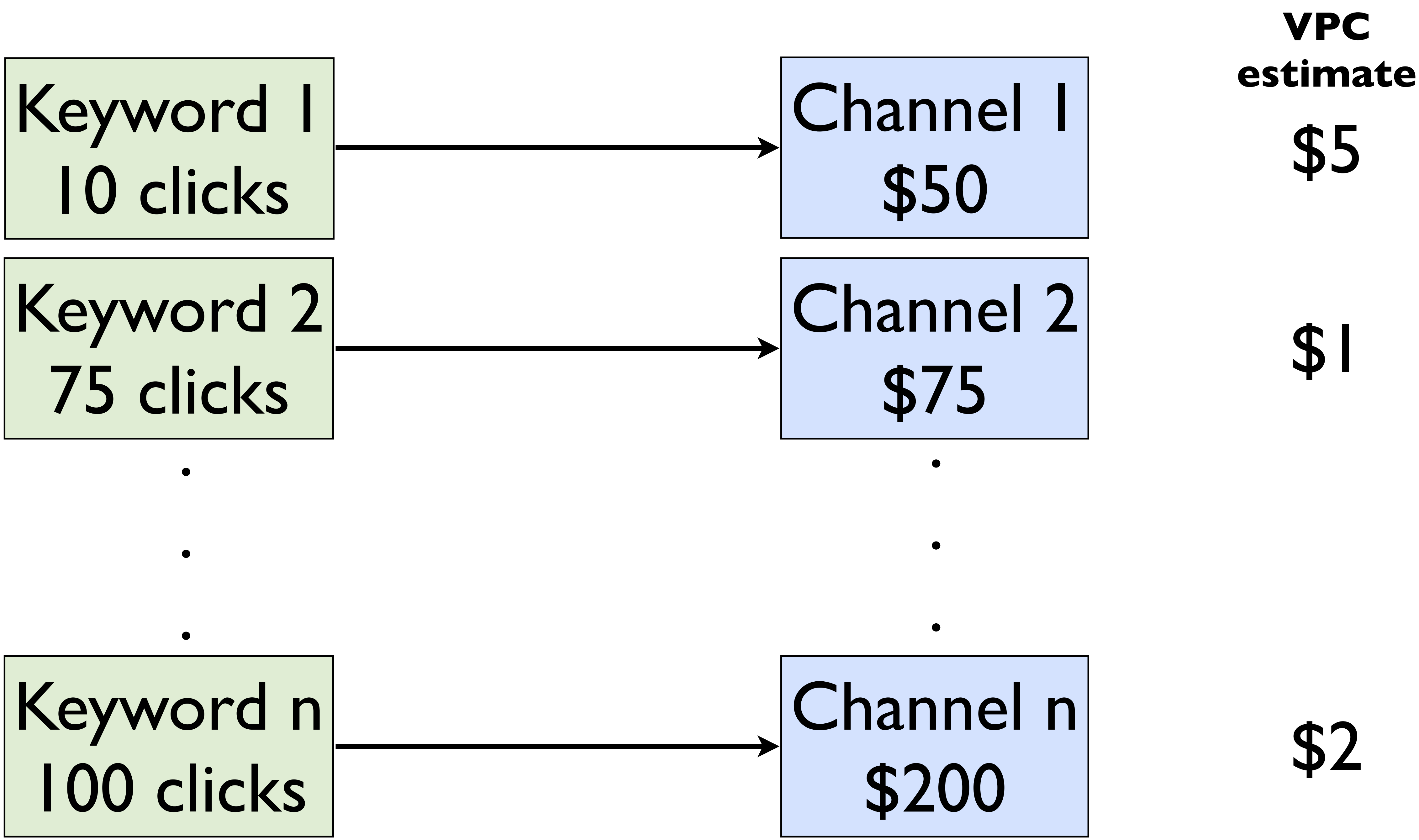
Unique setting: High skew & scale, variances per keyword instead of per channel measurement

Adaptivity:

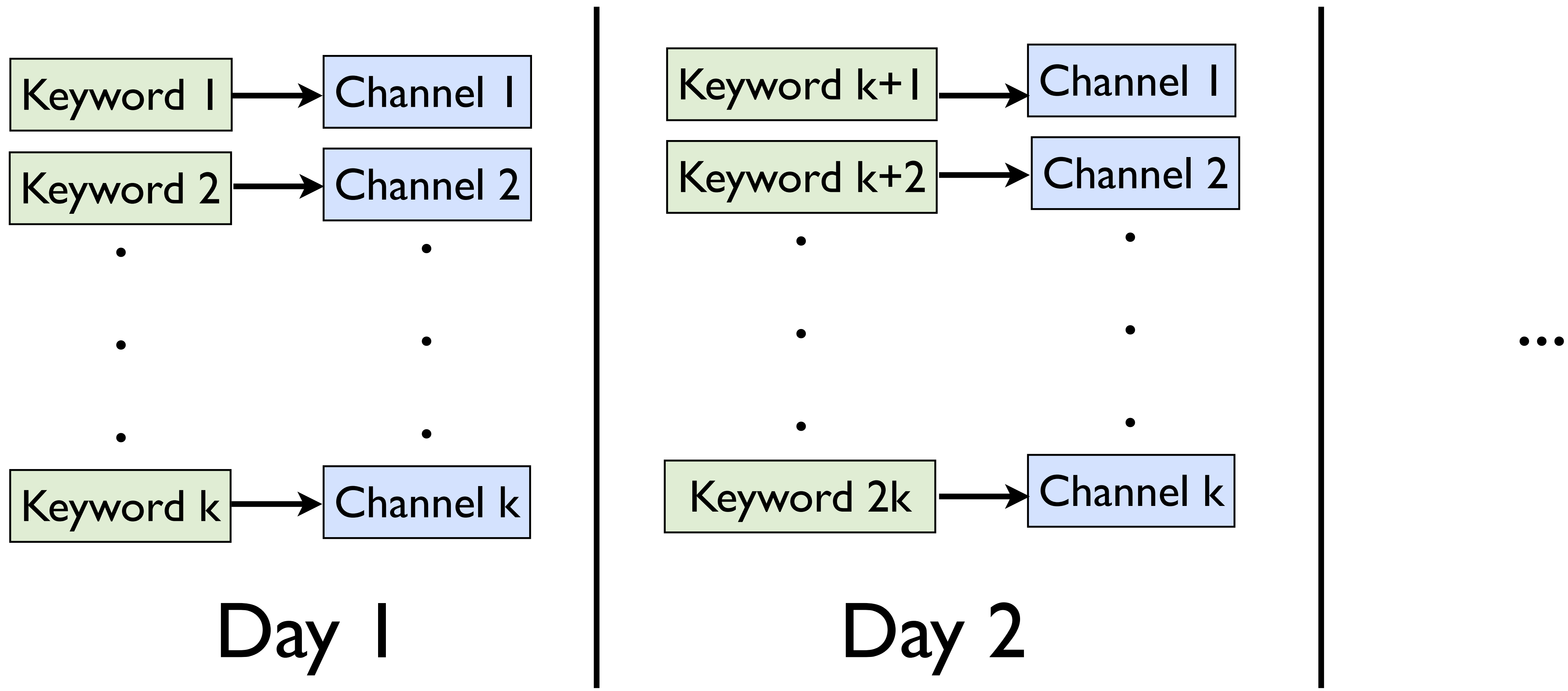
- We arrange measurements over time to speed up convergence
- Take cues from design of experiments [Hotelling 44]
    - Error associated with scale (equipment)

VPC Estimation

Straightforward if |keywords| ⩽ |channels|



What if |keywords| >> |channels|?
Idea 1: Round-Robin



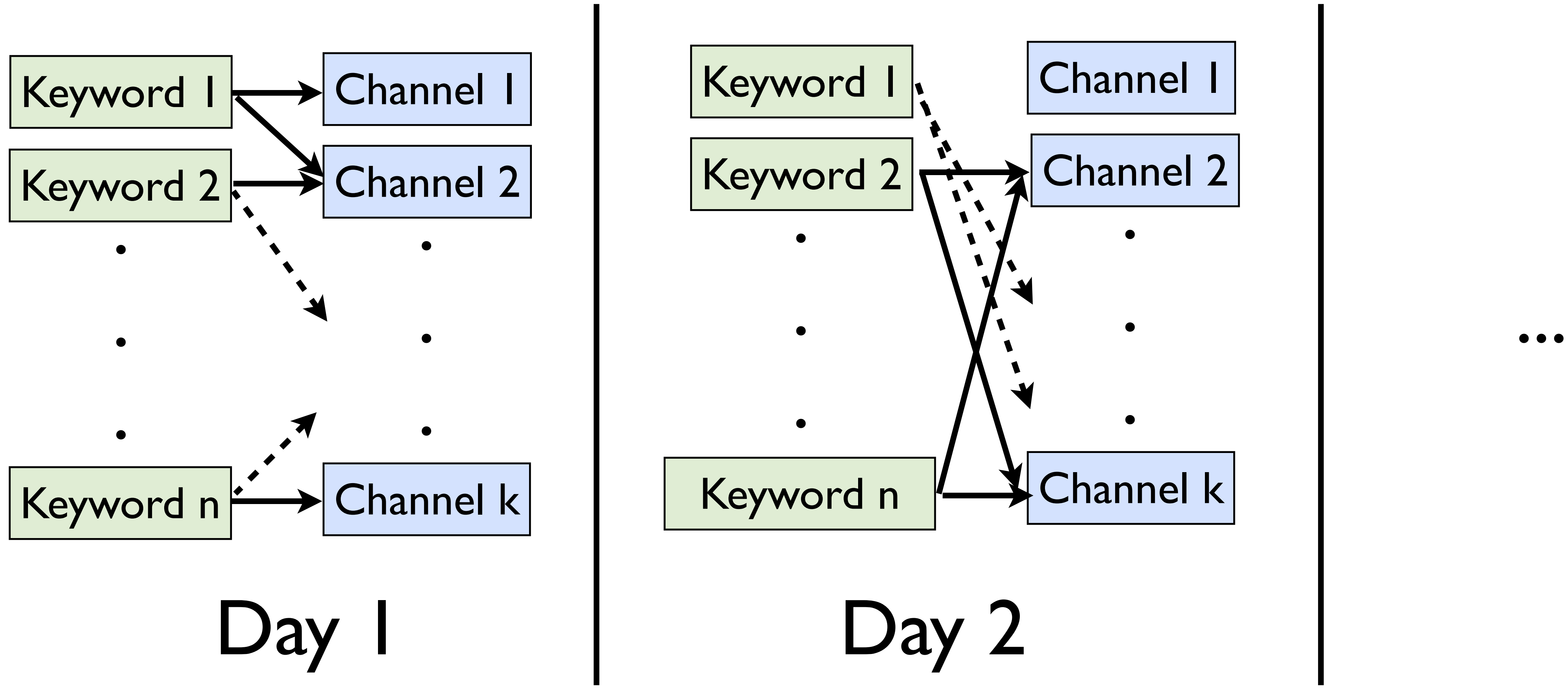
Solution

$$vpc_i = \frac{\text{Total revenue}}{\text{Total clicks}}$$

Very inefficient!

Sparse clicks and conversions

What if |keywords| > |channels|?
Idea 2: Regular-p



Assign each keyword's clicks and revenue to p channels each day

(chosen uniformly at random, note that some channels might be left empty)

Now each keyword is measured daily

Regular-*p* solution

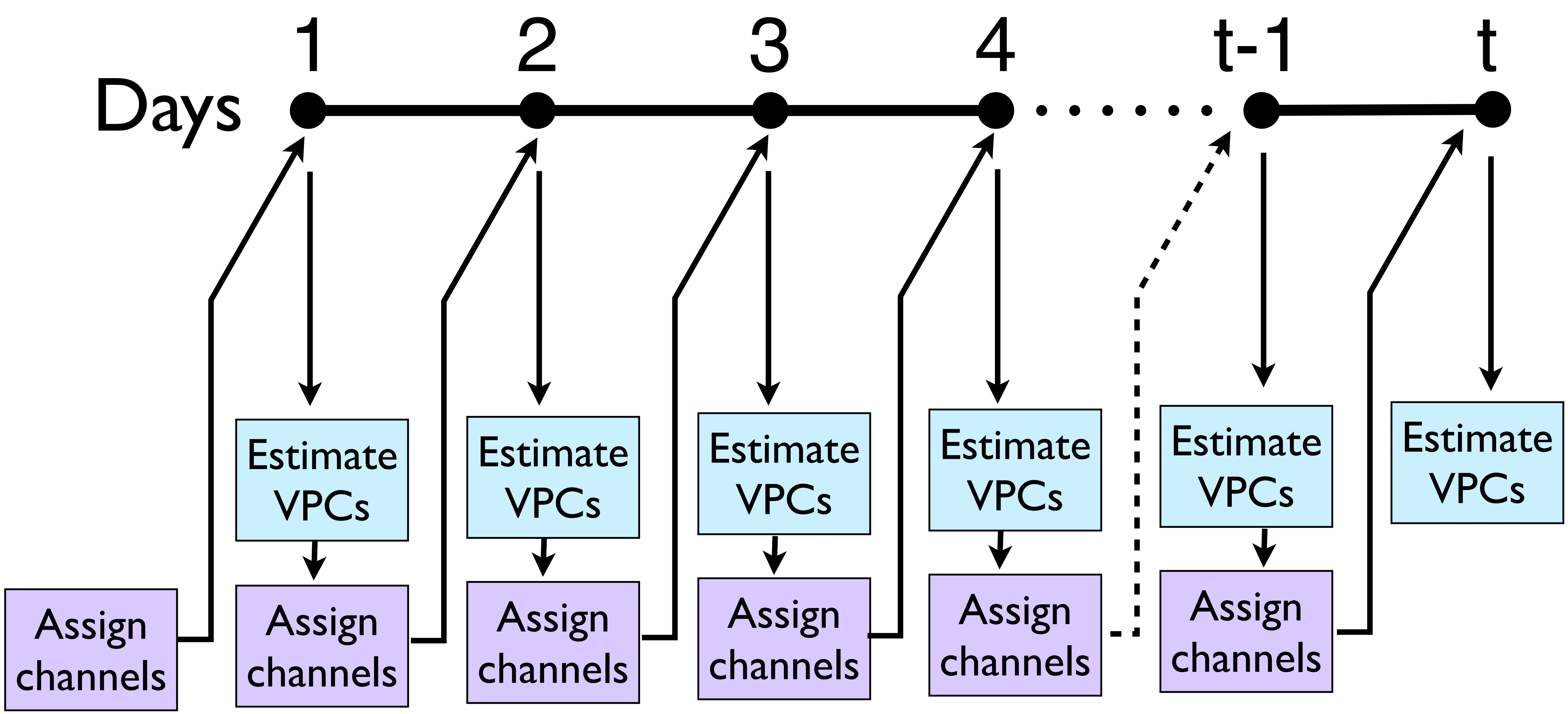
- After enough measurements we are left with the following **overdetermined** system of equations

$$r_{jt} = \sum_{i \in c_{jt}} \text{clicks}_{ijt} * \text{vpc}_i$$

$$\text{where } c_{jt} = \{\text{keywords in channel j on day t}\}$$

- Solve using linear regression to get daily keyword VPC estimates: $\hat{vpc}_{it}$

Possible improvement: adaptivity (previous solutions oblivious)



Use VPCs up to day *t* to inform channel selection for day *t+1*

Adaptivity with Linear Regression

- **Algorithm:** Adaptive-OLS
- Try to distribute expected revenue uniformly among channels to avoid overfitting.
- Use least-full bin-packing to pack:

$$E(revenue) = v\hat{p}c_i * avg(clicks_i)$$

- We also tried packing VPCs, clicks, etc, but expected revenue performed best.

Methods so far...

	Weighted Averages (1kw/ch)	**Linear Regression (many kw/ch)**
Oblivious	Round-Robin	Regular-p
Adaptive	Adaptive-1	Adaptive-OLS

Experimental Setup

- VPC (μ_i) and click (ν_i) means follow a skewed distribution (Zipf 1.8)
- Variances proportional to means
- Normally distributed daily VPCs and clicks. On day *t* keyword *i* has:

$$vpc_{it} \sim N(\mu_i, \sigma_i^2),\ clicks_{it} \sim N(\nu_i, \tau_i^2)$$

Comparing algorithms: Weighted RMSE

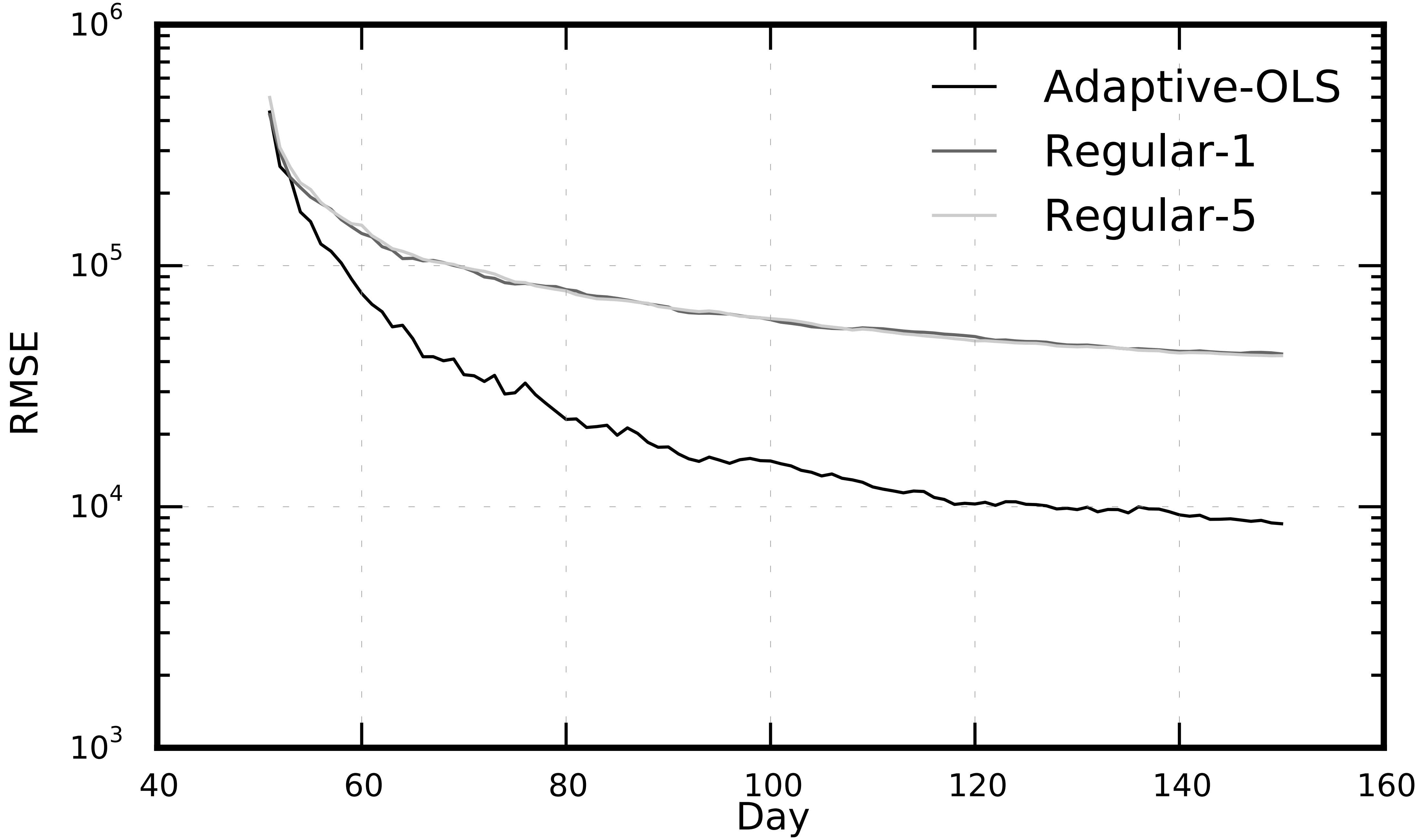
- Define the error on day *t* as

$$e_t = \sqrt{\frac{1}{n}\sum_i v_i^2 (\mu_i - v\hat{p}c_{it})^2}$$

- Error is units of dollars ($), suitable for optimizing bids

Adaptive vs Oblivious: Linear Regression

500 keywords and 10 channels



$$\sigma^2 = 2\mu,\ \tau^2 = \frac{1}{4}\nu^2$$

Heteroskedasticity

- Least squares assumption: equal variances in measurement errors.

  Violated: Errors per keyword, not channel

  Implication: Inefficient estimates

- **Solution:** Feasible Generalized Least Squares

  - Run OLS once
  - Divide each equation by $\sqrt{\text{residual}}$
  - Rerun OLS to get VPC estimates

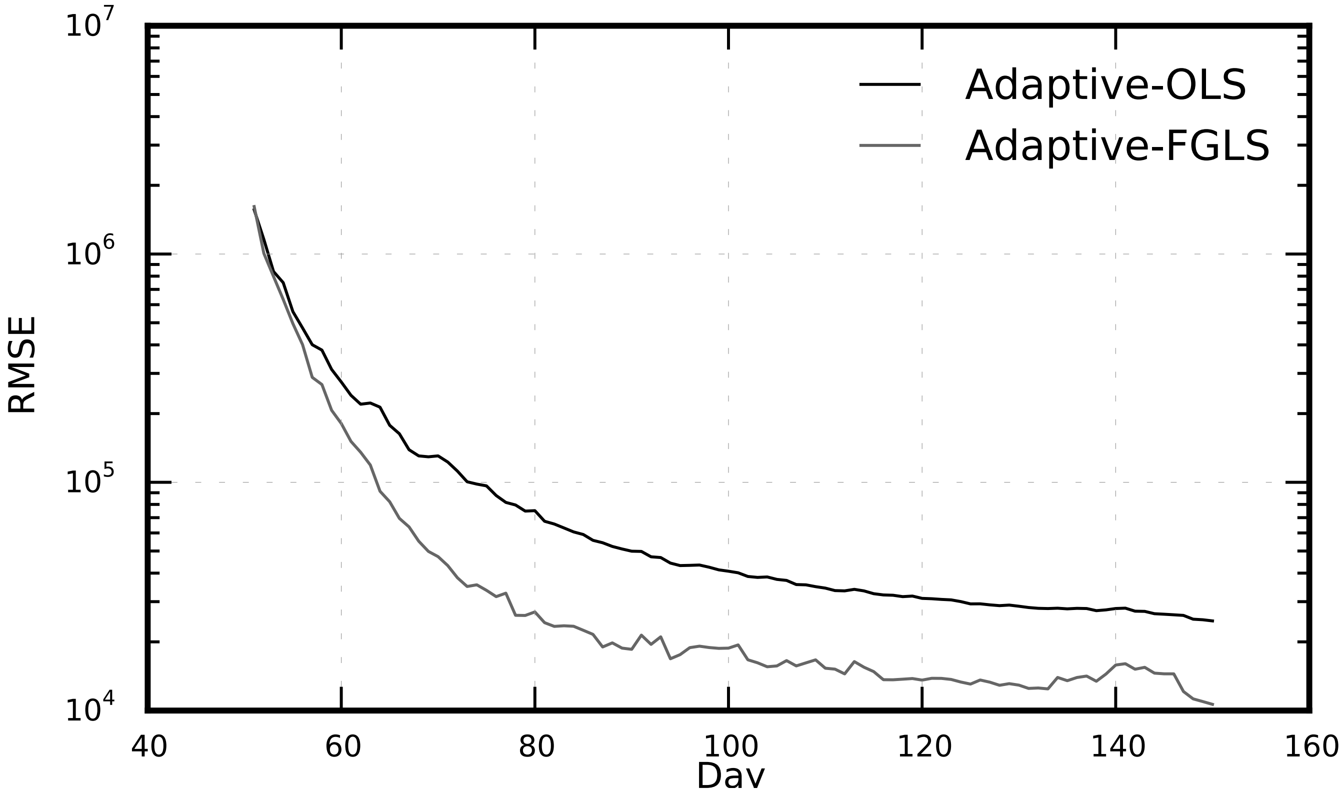
Effectively discounts equations with high measurement error

Algorithms overview

	Weighted Averages (1 kw/ch)	**Linear Regression (many kw/ch)**
Oblivious	Round-Robin	Regular-*p* *Regular-p-FGLS*
Adaptive	Adaptive-1	Adaptive-OLS *Adaptive-FGLS*

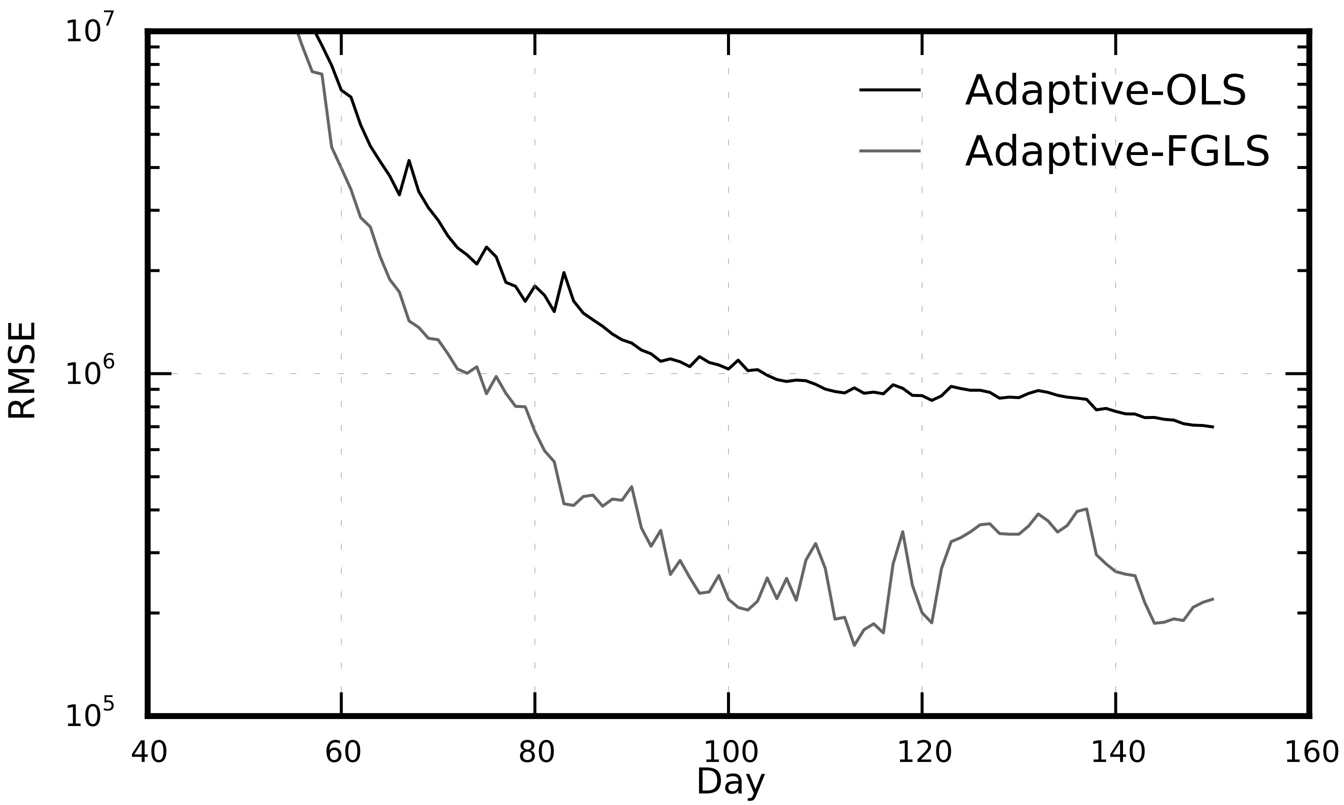
OLS vs FGLS

500 keywords and 10 channels



Low VPC and click variance:

$$\sigma^2 = 2\mu,\ \tau^2 = \frac{1}{4}\nu$$



High VPC and click variance:

$$\sigma^2 = 4\mu,\ \tau^2 = \frac{1}{2}\nu$$

Conclusion

- VPC computation is a foundational problem in online advertising
- We have demonstrated how to solve this problem in a setting with a limited number of channels, high noise and dimensionality

Applying similar methods in a real world setting with

- 100,000's of keywords
- 100's of channels

Yielded a doubling of ROI