



au cœur de la société de l'information

Improving web traffic inference using page level embedding information

Olivier Paul, GET/INT

MONAM 07, Toulouse, France





Schedule

- **Motivations**
- **Work performed**
 - Models.
 - Implementation.
- **Conclusion**



Motivations

■ Understand traffic

- Without requiring heavy changes to monitoring devices.
- Without requiring computing intensive processes.
- Without needing access to packets payload.

■ Better

- Get a more synthetic view than

Flow# 1518:

Time: [1182183194.936335:1182183195.224169]

157.159.100.43:37770 -> 157.159.100.56:8080

[**pkts** 56:**bytes** 49341] **session#:** 1

[**fwd** 354:**bkw** 46059] **End Cause:** Fin

■ For web traffic

- Very popular.



Why not

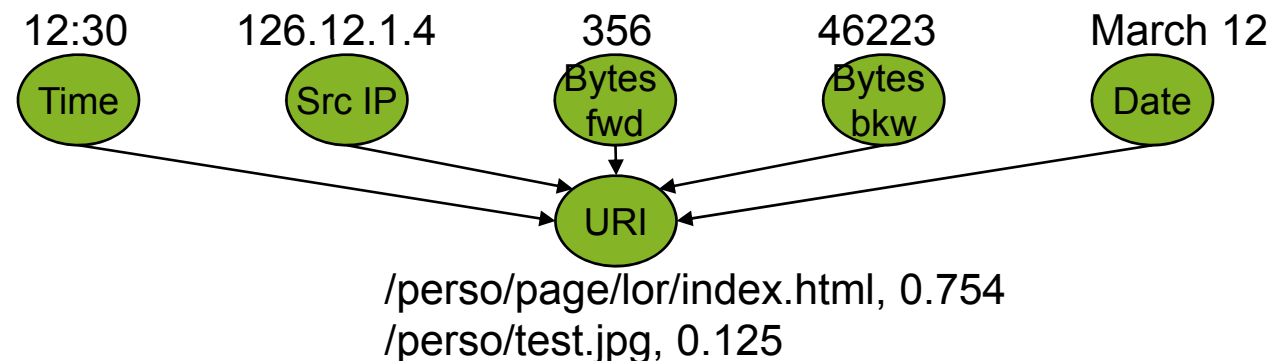
- **Use a proxy**
 - Not always transparent to end user.
 - Requires redirecting traffic.
 - Usually able to handle a few thousands connections per seconds on standard PC equipment.
- **Use web server logs**
 - Requires cooperation with monitored server.
 - Reduced reliability.



RequIn

■ Web traffic inference tool

- Uses flow level measures
- Produces application level information.
 - URI, Method, Response Code.



- Object based: 1 flow record = 1 object request-response.

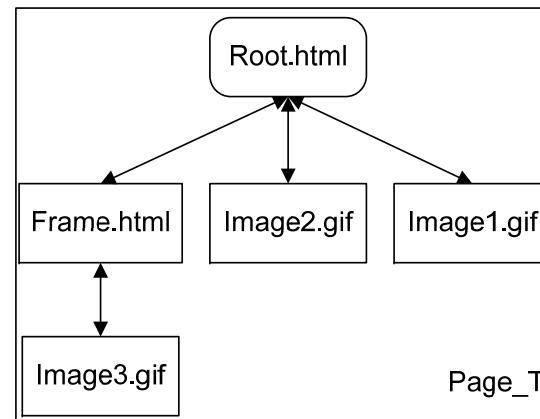
■ Pros: Very fast, Not too inaccurate.

■ Cons: Probabilistic, Inferred information varies.



Page level information

- Information often organized as pages



- **Typical Browsing Behavior**
 - Root.html is transferred.
 - Browser load automatically referred objects.
 - User reading time (>1s).
- **Flow sets separated by thinking time.**

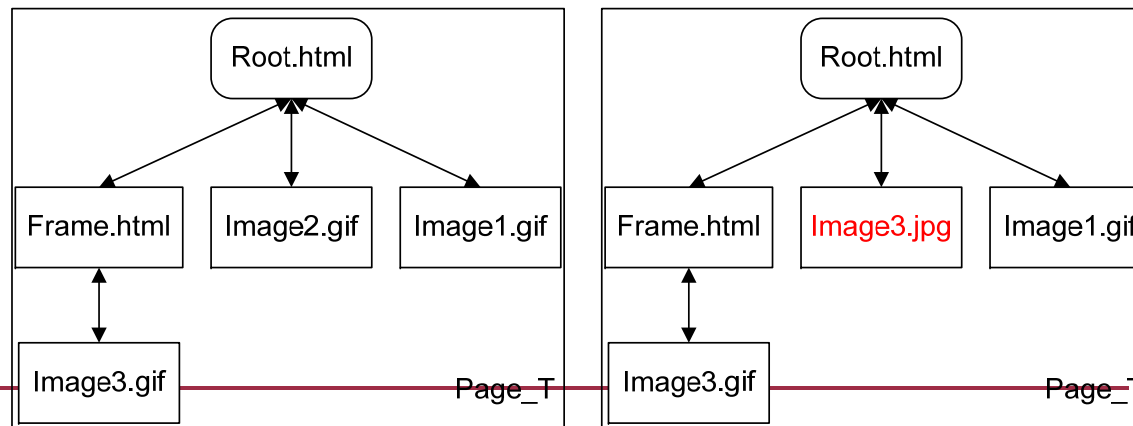


Expected Benefits Examples

■ Expanding inference results.

Result code	Method	URI	Object Size
200	GET	Yes	Yes
	Others	No	Yes
Others	PUT/POST	No	Yes
	Others	No	No

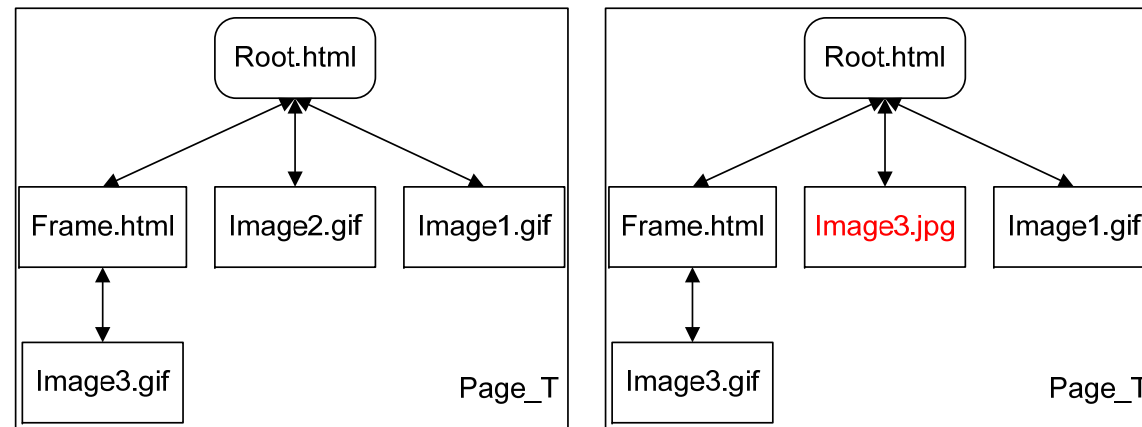
■ Correcting false predictions.





Expected Benefits Examples

- Expanding inference results.
- Correcting false predictions.

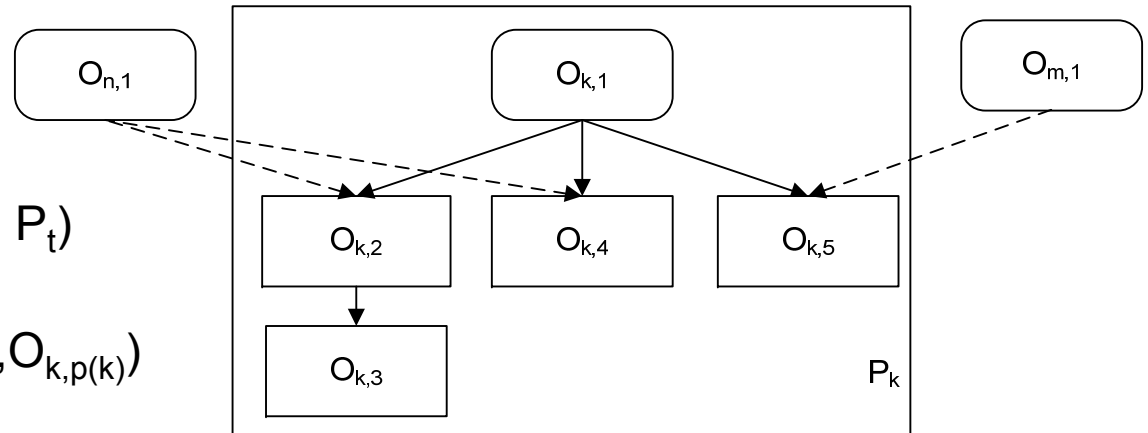




Formalization

■ Notation:

- Site: (P_1, \dots, P_t)
- Page: P_k
 $(O_{k,1}, O_{k,2}, \dots, O_{k,p(k)})$



■ We get before using object level inference:

- A set of flows: $F=(F_1, \dots, F_f)$.

■ We get for each flow after using object level inference:

- A Method: C_i , a Response Code R_i .
- A set of objects with their respective likelihoods:

$$S_i = ((O'_{i,1}; L_{i,1}), (O'_{i,2}; L_{i,2}) \dots (O'_{i,n(i)}; L_{i,n(i)})).$$

■ We want to match these S_i sets with a Page P_j

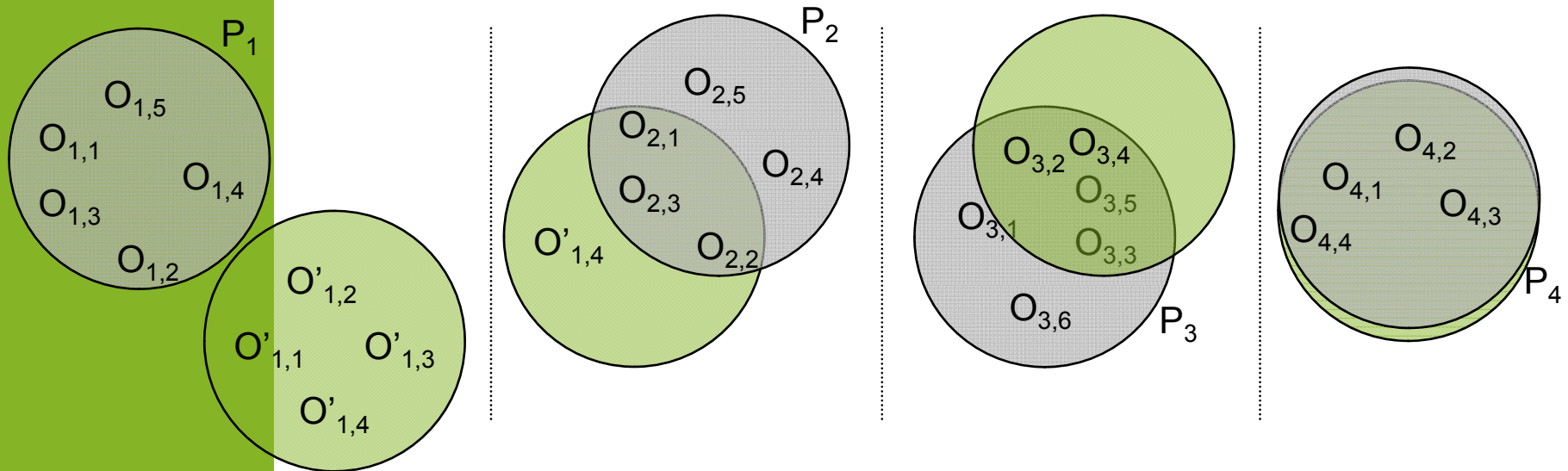


Naïve proposal

■ Matching procedure.

(1) $\forall i, 1 \leq i \leq f; \exists j, 1 \leq j \leq n(i); \exists k, 1 \leq k \leq t; \exists l, 1 \leq l \leq p(k) / O'_{i,j} = O_{k,l}$

(2) $\sum_{i=1}^f L_{i,j}$ is maximized.

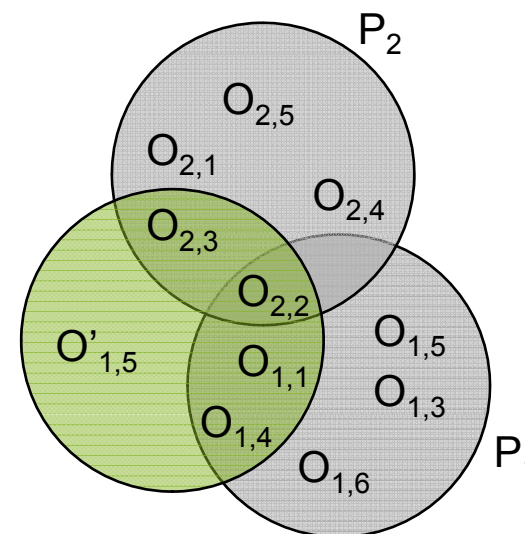




Naïve proposal

■ Problems.

- $O'_{i,j}$ might not match any requested object $O_{k,l}$.
- a requested object $O_{k,l}$ might not match any inferred object $O'_{i,j}$.
- Computationally intensive.
 - Average number of objects per page.
 - Average number of objects per flow.
 - Number of flow per request.
 - Number of pages.





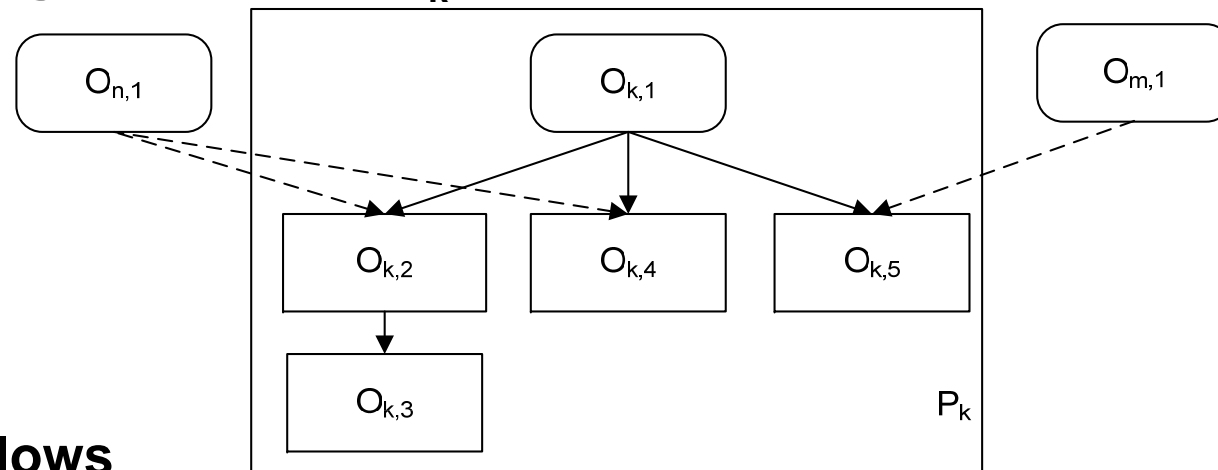
Scoring based approach

- **Scoring potential root object $O'_{i,j}$**
 - Score depends on likelihood $L_{i,j}$
 - Score depends on likelihood $L_{k,l}$ of inferred referred objects $O'_{k,l}$.
 - Score depends on (Number of flows = Number of objects referred by $O'_{i,j} - 1$).
 - Score depends on $O'_{i,j}$ being a root.
- **Object with highest score is considered as the page root.**
- **Computational cost:**
 - Number of flows.
 - Average number of objects per flow.
 - Average number of referrer per object.



Example (1)

■ Page Request to P_k



■ Flows



■ Inferred objects





Example (2)

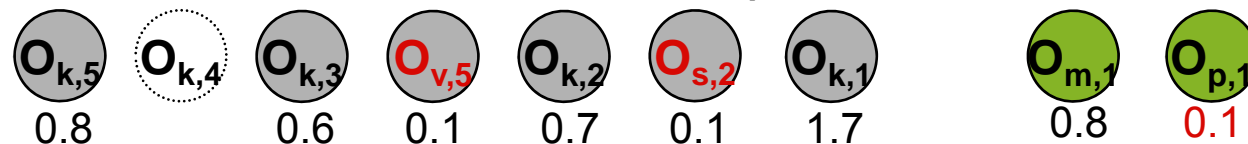
- Selecting $O_{k,5}$ referred by $O_{m,1}$ and $O_{k,1}$



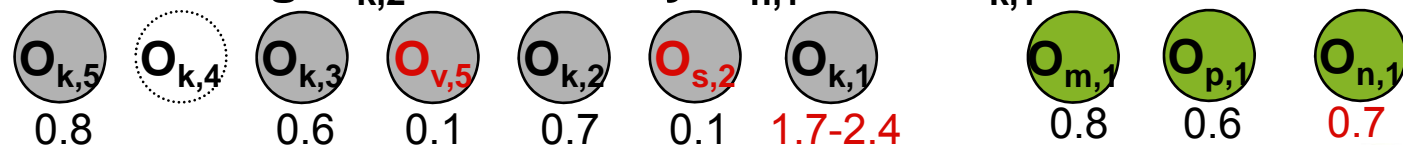
- Selecting $O_{k,3}$ referred by $O_{k,2}$



- Selecting $O_{v,5}$ referred by $O_{p,1}$



- Selecting $O_{k,2}$ referred by $O_{n,1}$ and $O_{k,1}$



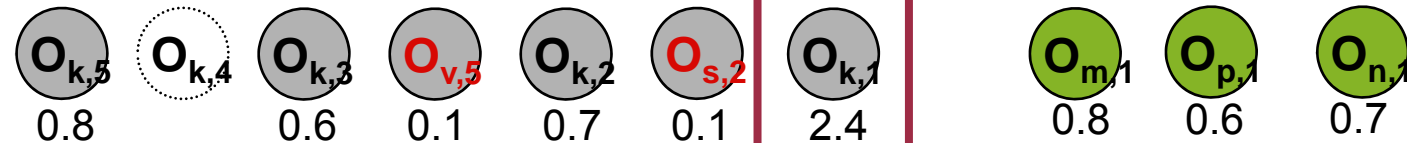


Example (3)

■ Selecting $O_{s,2}$



■ Selecting $O_{k,1}$



Selected root

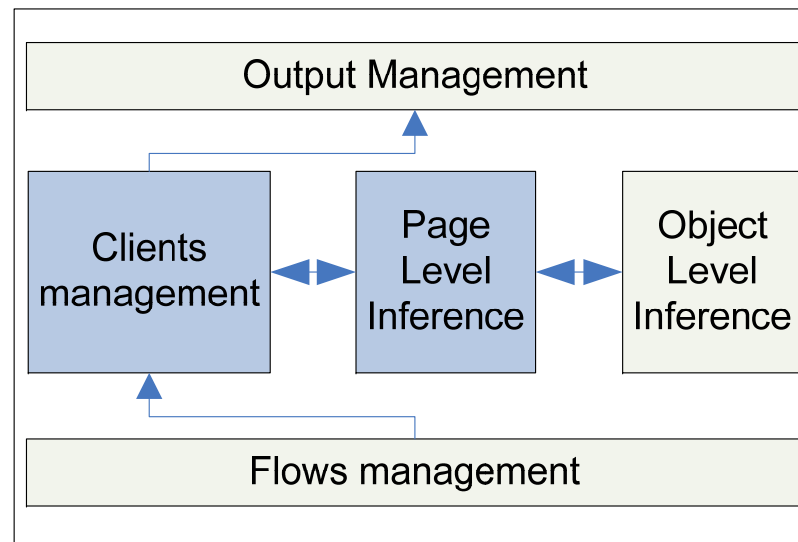
■ Mapping flows to objects





Implementation

■ Architecture





Results

■ Comparison with object based approach.

- Server with 17k objects, 3.5k pages.

Method, Set	Correctly Inferred	Incorrectly Inferred	Time per request
Object Based,Set1	64%	36%	1.9us
Page Based,Set1	82%	18%	9.7us
Object Based,Set2	57%	43%	2.0us
Page Based,Set2	81%	19%	21us



Existing work

■ Sun & al, IEEE S&P 2002.

- Flows Sizes → Page vs Flow Descriptors → Objects → Page
- Each page request is associated with a set of flows sizes.
- Comparison between an actual request U and signature V using Jaccard coefficient:

$$J(U, V) = \frac{|U \cap V|}{|U \cup V|}$$

Method, Set	Correctly Inferred	Incorrectly Inferred	Time per request
Jaccard Based, Set1	49%-71%	51%-29%	/
Page Based, Set1	82%	18%	9.7us
Jaccard Based, Set2	45%-63%	55%-37%	/
Page Based, Set2	81%	19%	21us



Conclusion

- **New technique to obtain application level information**
 - Without using packet content.
 - Without requiring heavy changes in networking devices.
- **Might work with encrypted traffic.**
- **Computing intensive (~50-100k requests/s) but less intensive than proxying.**
- **Potential problems:**
 - Highly dynamic web sites (blogs/forums).



Thanks !