A Query Unit for the IPSec Databases

Alberto Ferrante,
Satish Chandra

ALaRI,
University of Lugano
E-mail: {ferrante,
kaverips}@alari.ch

Vincenzo Piuri

DTI,
University of Milano
E-mail: piuri@dti.unimi.it

Outline

IPSec

The Database Query Unit

Multithreaded Unit

Simulations

Conclusions and Future Work

IPSec

- ✔ Is a suite of protocols
  - ✘ adding security at IP (network) level;
- ✔ makes extensive use of cryptographic functions.

AH, ESP

- ✔ IPSec is mainly composed of two protocols:
  - ✘ Authentication Header (AH);
  - ✘ Encapsulating Security Payload (ESP);
- ✔ both protocols can be used in:
  - ✘ transport mode;
  - ✘ tunnel mode.

Databases

- ✔ IPSec uses two databases:
  - ✘ the Security Policy Database (SPD);
  - ✘ the Security Association Database (SAD):
    - ✓ the records are the Security Associations (SAs).

Security Associations

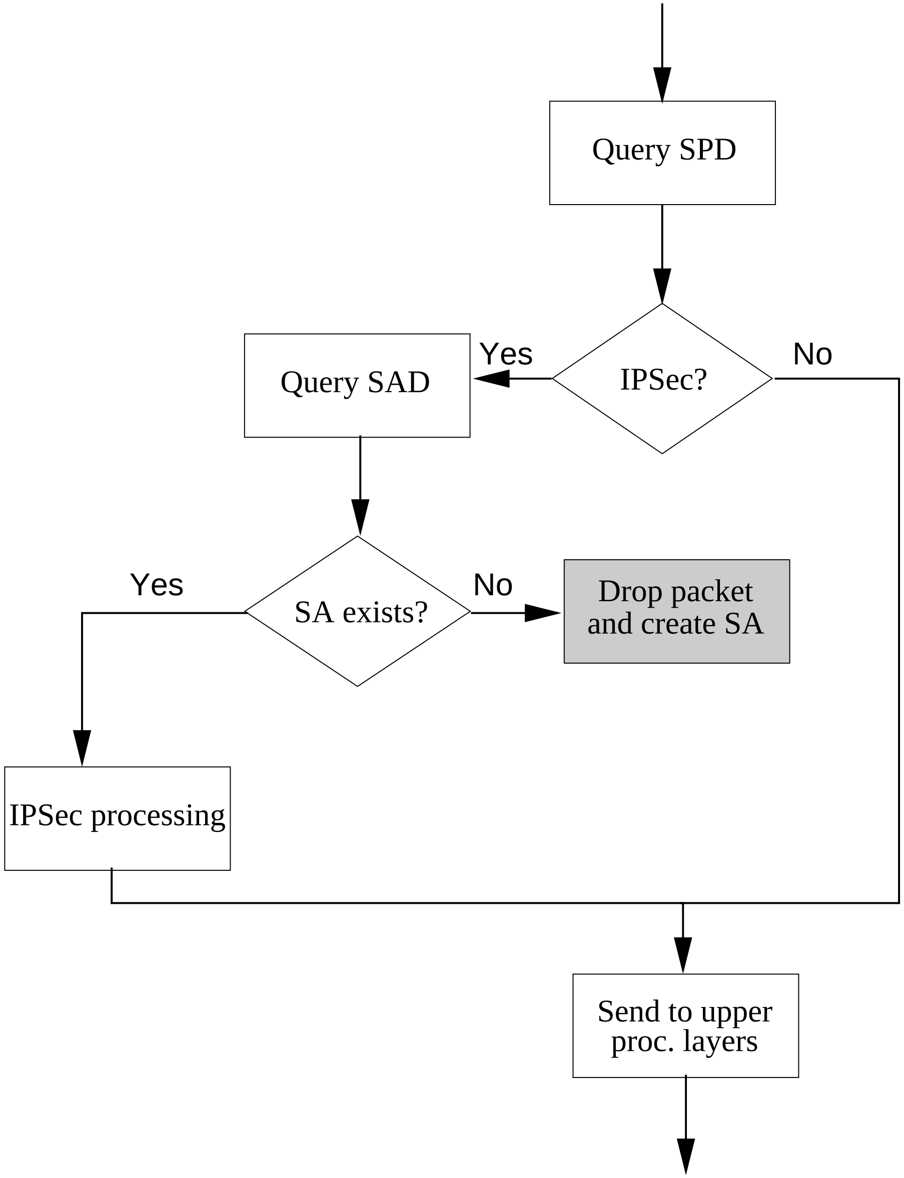
✔ Each SA contains:

- ✘ protocol/algorithms settings;
- ✘ keys for cryptographic algorithms;

✔ SAs are mono-directional:

- ✘ two SAs need to be created
  for normal bidirectional communications.

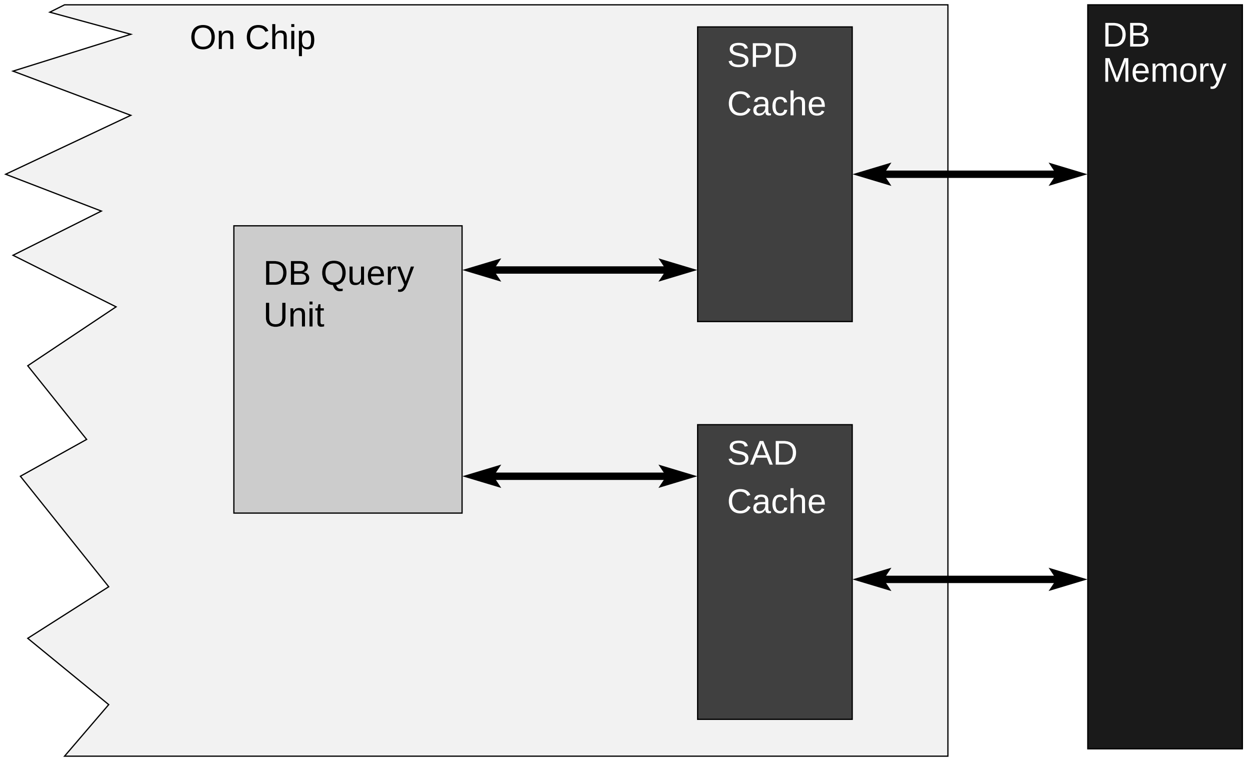
Main IPSec Processing Steps



Database Query

- ✔ More than 3 million queries/s in a 1Gbit/s system (worst case);
- ✔ may be a bottleneck;
- ✔ may become a weak point (DoS).

High-level Architecture



Caches are Content Addressable Memories.

IPSec
The Database Query Unit
High-level Architecture
How it Works
Record Size
Memory Query Techniques
Cache Replacement Policies
Multithreaded Unit
Simulations
Conclusions and Future Work

How it Works

- ✔ SPD query:
  - ✘ cache query;
  - ✘ main DB query if not in cache;
- ✔ SAD query:
  - ✘ cache query;
  - ✘ main DB query if not in cache:
    - ✓ SPD-provided pointer.

Record Size

SPD:

- ✔ two parts:
  - ✘ repeatedly used information (IP, SA pointers, . . .);
  - ✘ rarely used information (proposals);
- ✔ repeatedly used information (232 bits) are cached.

SAD:

- ✔ all fields are repeatedly used (792 bits).

Memory Query Techniques

- ✔ Linear LookUp Technique (LLUT);
  - ✘ memory queried in a linear fashion;
- ✔ Partitioned LookUp Technique (PLUT).
  - ✘ memory divided into pages;
  - ✘ IP address is used
    to associate a record to a page;
  - ✘ linear search inside the pages;
  - ✘ "fragmentation" problem.

Cache Replacement Policies

- ✔ First In First Out;
- ✔ Least Recently Used.

Parallelizing Queries

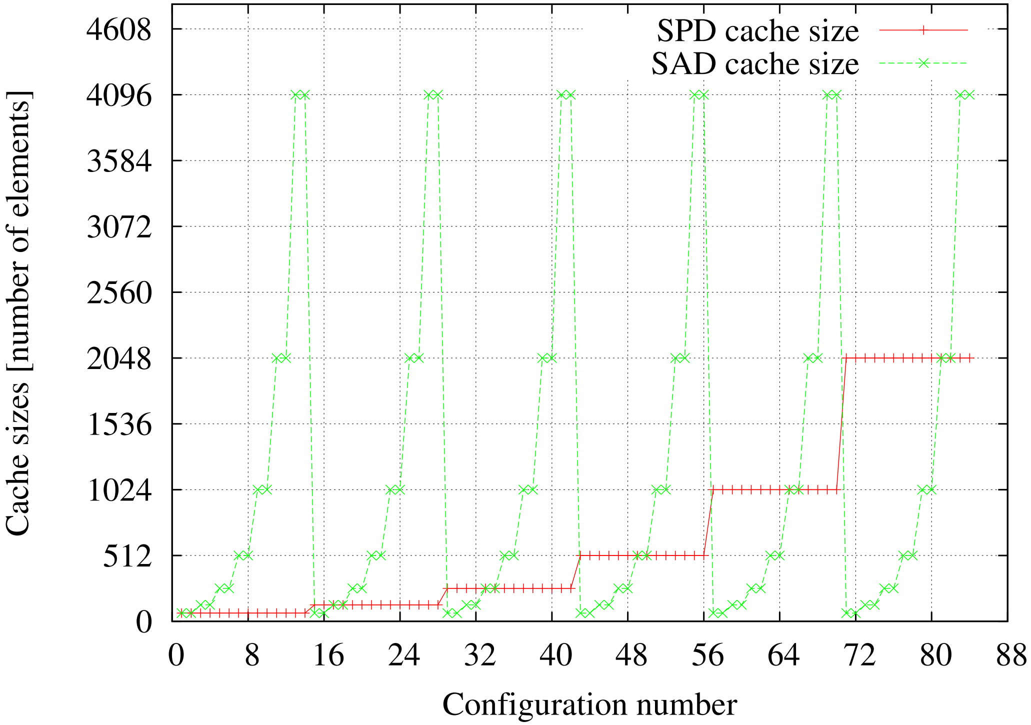
- ✔ Queries in memory take a long time;
- ✔ other queries in cache can be done during this time;
- ✔ parallel queries related to the same SA are not allowed.

Simulation Description

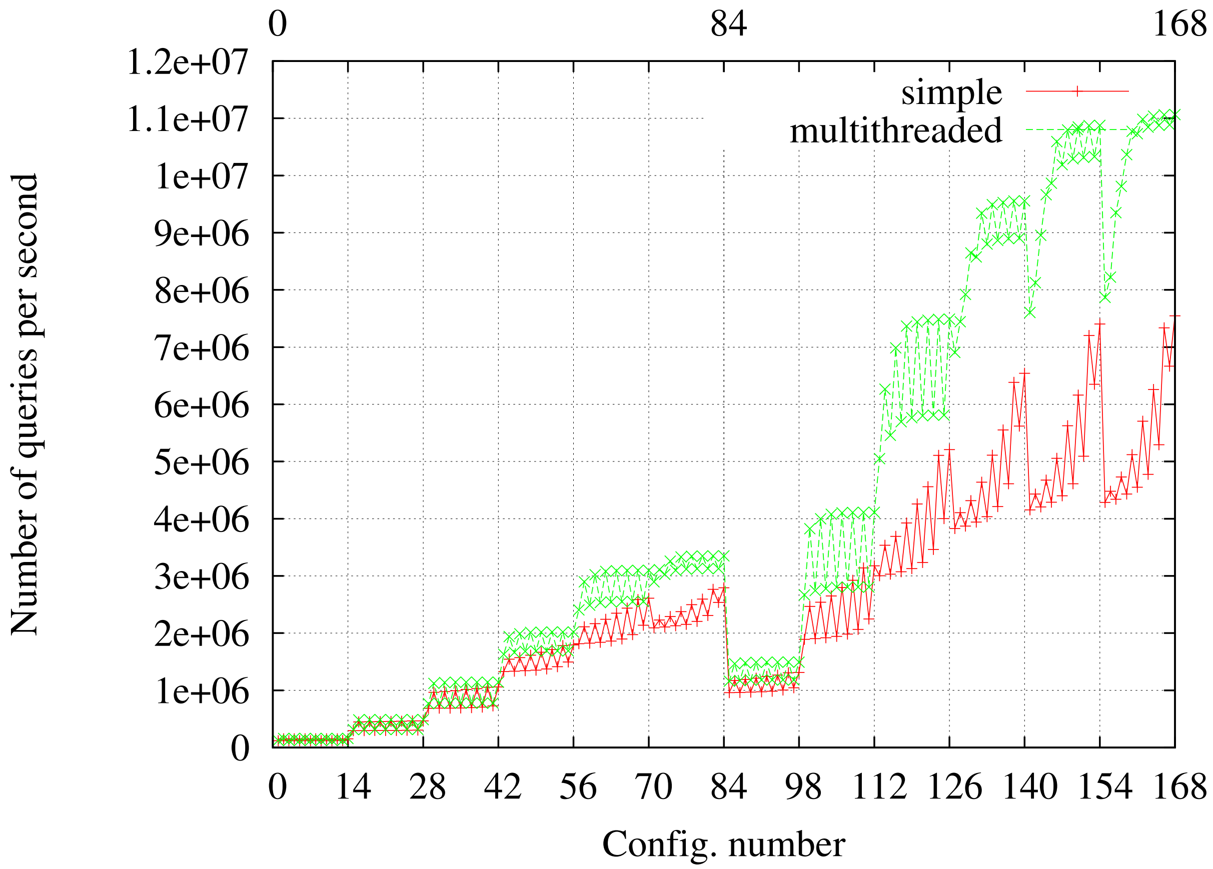
- ✔ SystemC functional model;
- ✔ simulates behavior and delays of the blocks;
- ✔ input: ITA tracefiles.

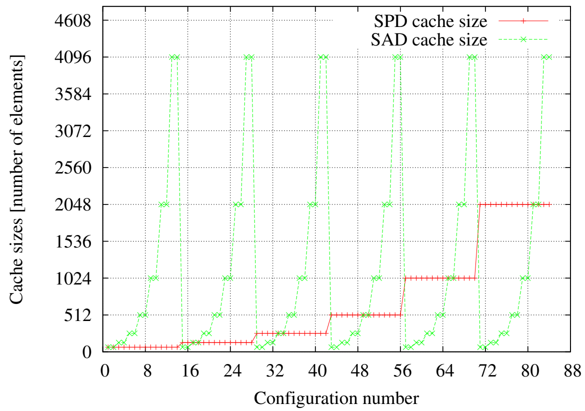
Design Space

- ✔ 168 different configurations;
- ✔ LLUT for the first 84 configurations, PLUT for the others;
- ✔ FIFO for odd configurations, LRU for even ones.

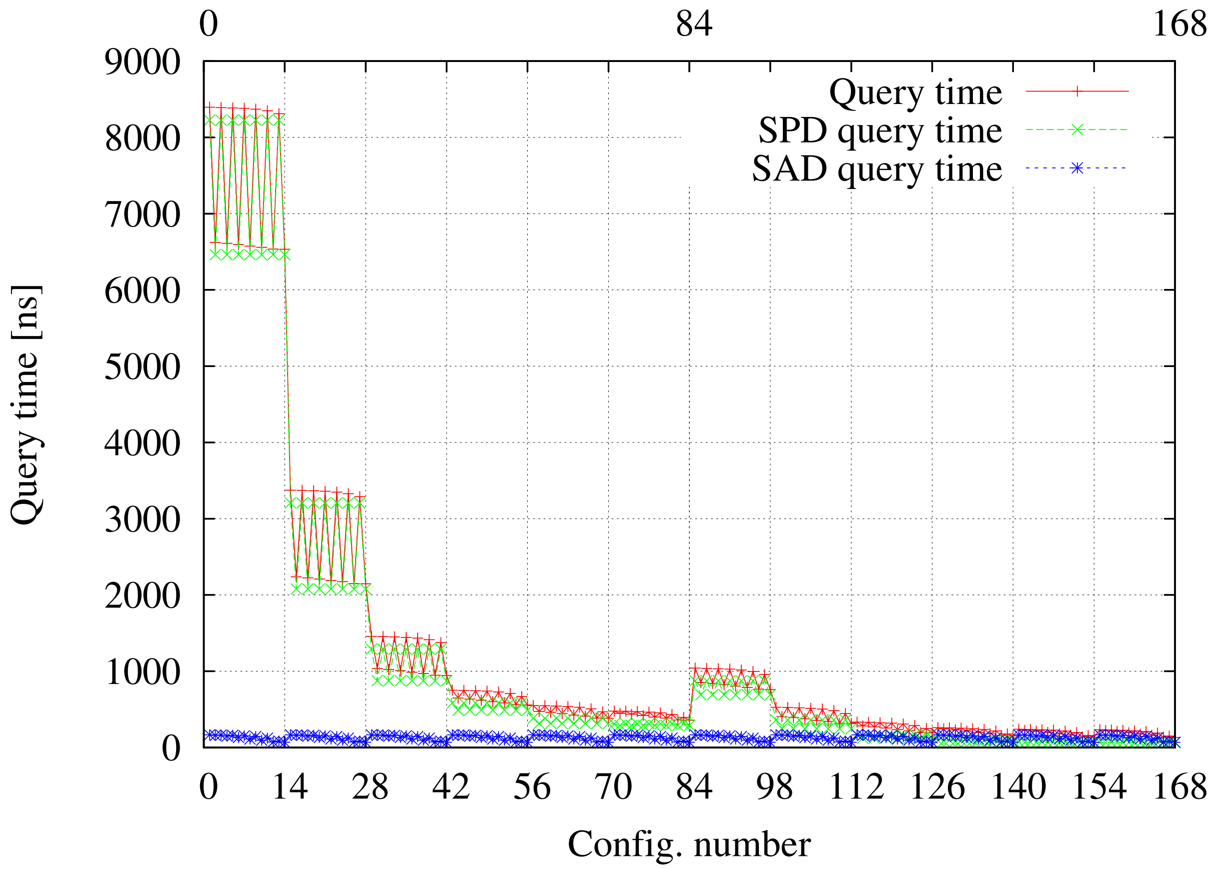


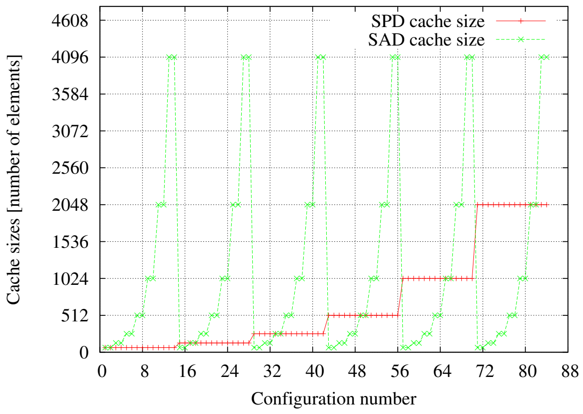
Queries/second



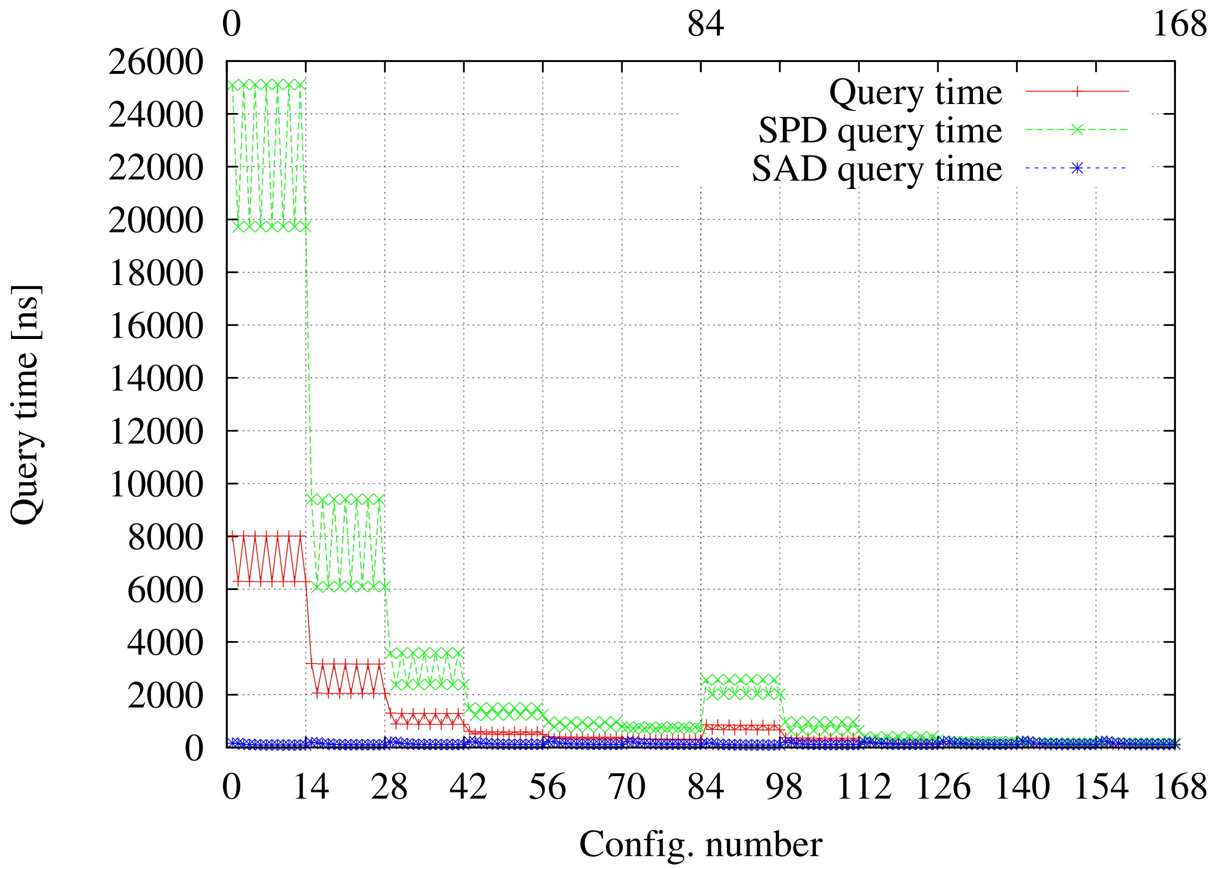


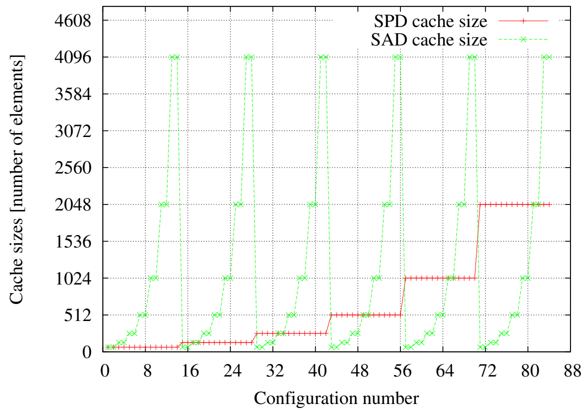
Sequential System: SAD and SPD Query Times





Multithreaded System: SAD and SPD Query Times





Conclusions and Future Work

We designed a DB query unit:

- ✔ able to exceed 11 million queries per second;
- ✔ efficient.

Future Work:

- ✔ more accurate simulations;
- ✔ out of order queries.