

PSS

Physics Services Support

CERN IT  
Department

**COOL**

**Performance Tests and Optimizations**

**The COOL development and testing team**

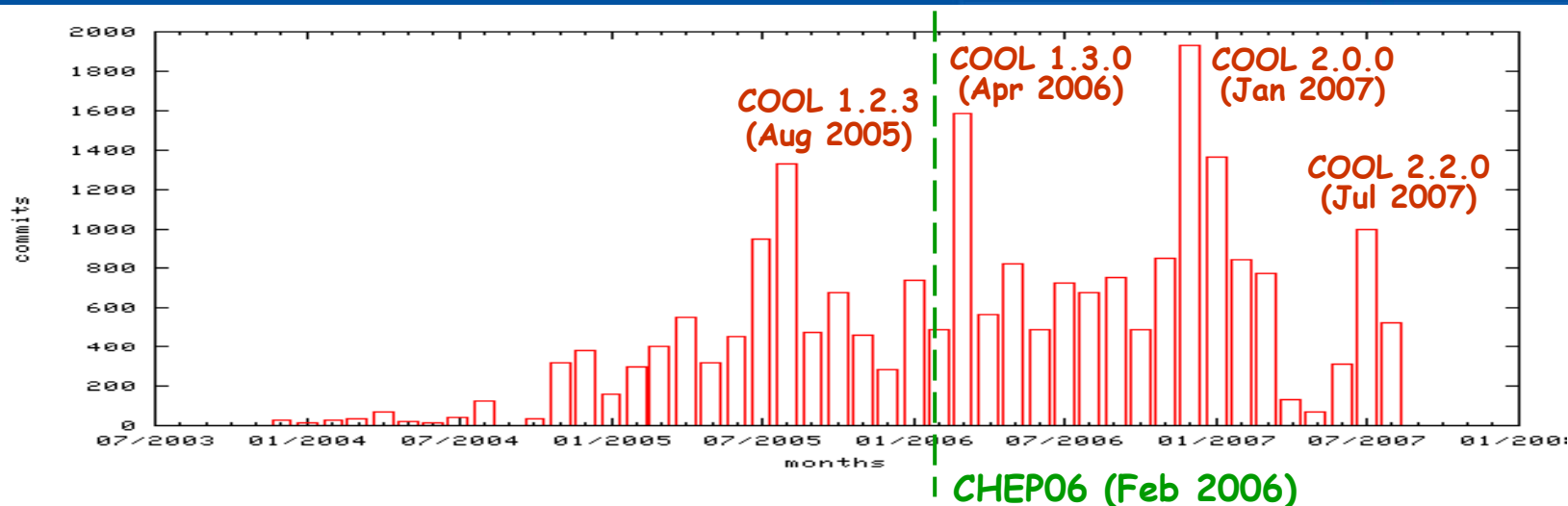
*A. Valassi, D. Front, G. Pucciani, K. Dahl, M. Clemencic, R. Basset, S. A. Schmidt*

***CHEP2007, 3rd September 2007***

***Prepared by Andrea Valassi (CERN IT-PSS-DP)***

***Presented by Marco Clemencic (CERN PH-LBC)***





- **COOL 2.0 development took 9 months**
  - Parallel development on the CVS 1.3.x (bug-fix) and HEAD branches from May 2006 to Jan 2007
  - Main focus: major API and schema changes (group all backward-incompatible changes)
  - Extensive testing (in multi-threaded environment)
- **Further improvements in COOL 2.1 and 2.2**
  - Main focus: performance optimizations

- **Server-side SQL optimizations**
  - Oracle only (may not apply to MySQL/SQLite)
  - Through several means combined
    - Reengineer the SQL query strategy
    - Add missing indexes
  - Several use cases *independently* improved
    - Single-version single-channel IOV retrieval
    - Single-version multi-channel IOV retrieval
    - Multi-version user-tag IOV retrieval
    - Multi-channel bulk insertion
  - *Still the main priority also for future development!*
- **Client-side profile optimization**
  - Reduce COOL overhead over CORAL by avoiding extra in-memory copies of the data

- **Example: a simple single-version IOV table**
  - System-managed common “**metadata**”
    - Data items: many tables, each with many “channels”
    - *Interval of validity - IOV: since, until*
    - Versioning information with handling of interval overlaps
  - User-defined schema for “**data payload**”
    - Support for simple C++ types as CORAL “AttributeList”
- **Most server-side optimizations are about improving the SQL queries on this table**

| <i>objectID</i> | <i>channelID</i> | <i>since</i> | <i>until</i> | <i>pressure</i> | <i>temperature</i> |
|-----------------|------------------|--------------|--------------|-----------------|--------------------|
|                 |                  |              |              |                 |                    |
|                 |                  |              |              |                 |                    |

**Metadata**

System-controlled

*(versioning metadata not shown)***Data payload**

User-defined schema

*(different tables for different schemas)*

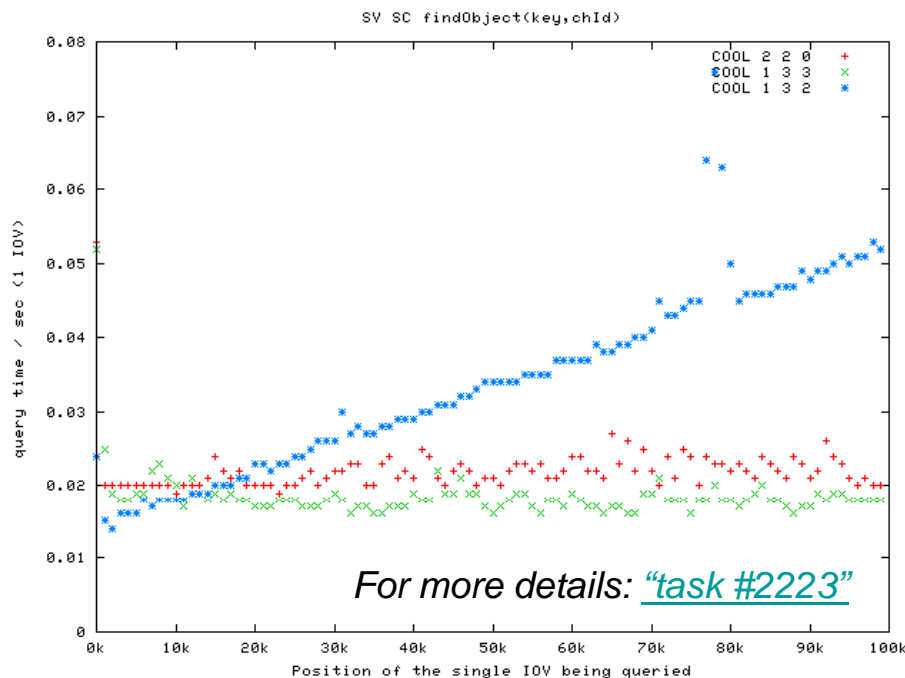
- **Example: get the single IOV at t=20 for channel 5**
  - SV (single version): there is only one version at any time t
  - SC (single channel): just select *ChannelId=5*
- **Problem until COOL 1.3.2 included**
  - Retrieval time is longer for IOVs at the end of the IOV table
  - Poor SQL "( *Since* <= 20 AND 20 < *Until* )" - needs two columns
    - Index lookup on first column, full scan for second column

- **Fixed in COOL 1.3.3**

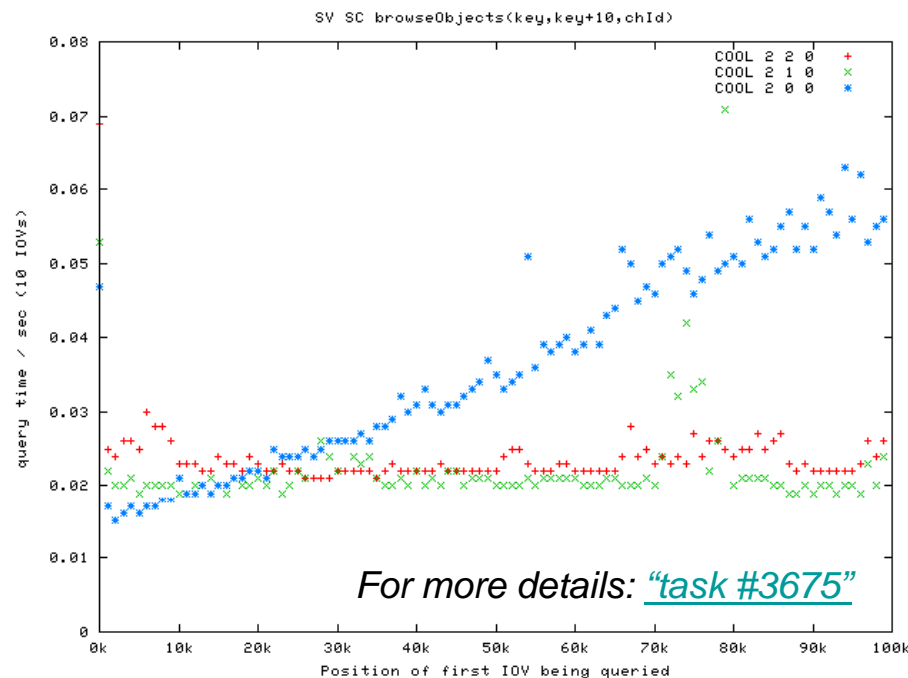
- Better use of *Since* column
- Two separate SQL queries
  1. *MAX(Since) WHERE Since < 20*
  2. *Since = maxSince* (from query1)

- **Design difficulty for IOVs**

- IOV2 [*s2*,*u2*] follows IOV1 [*s1*,*u1*]
  - We know that  $s2 \geq u1$  (no overlap)
  - The database does not know it
- There is no "time range" data type



- **Example: get all IOVs in  $t=[20,30]$  for channel 5**
  - SV (single version): there is only one version at any time  $t$
  - SC (single channel): just select *ChannelId=5*
- **Problem until COOL 2.0.0 included**
  - Retrieval time is longer for IOVs at the end of the IOV table
  - $(\text{Since} \leq 20 \text{ AND } 20 < \text{Until}) \text{ OR } (20 < \text{Since} \text{ AND } \text{Since} \leq 30)$   
IOV valid at  $t=20$  (inefficient lookup – two columns)



- **Fixed in COOL 2.1.0**

- Optimize lookup of first IOV
  - As in fix for SV SC single-IOV find
- Two separate SQL queries
  1.  $\text{MAX}(\text{Since}) \text{ WHERE } \text{Since} < 20$
  2.  $\text{Since} = \text{maxSince}$  (from query1)

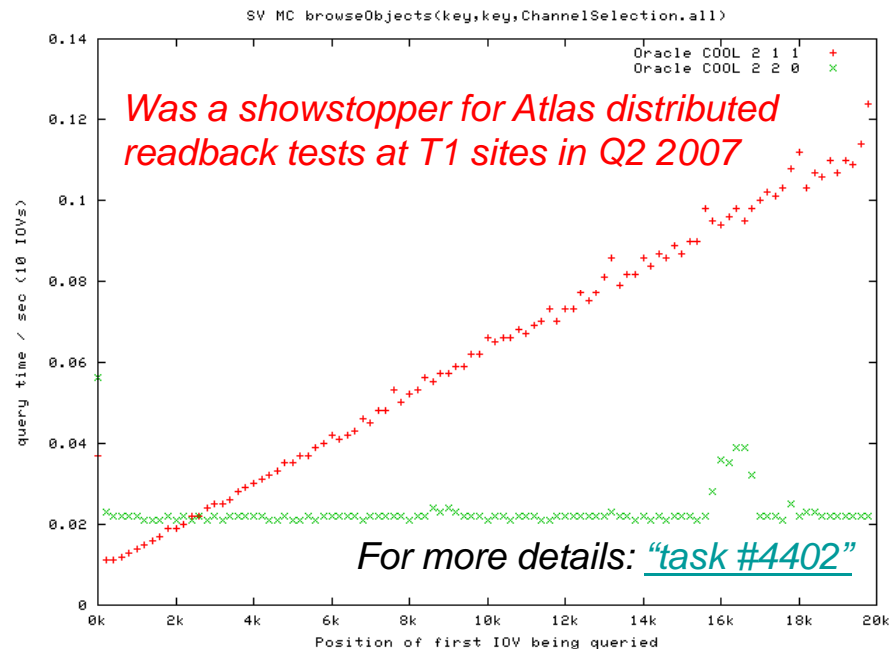
- **New strategy in COOL 2.2.0**

- Merge two queries in a single SQL query (use subqueries)
- Needed for SV MC case

- **Example: get all IOVs in  $t=[20,30]$  in channels 1-99**
  - SV (single version): there is only one version at any time  $t$
  - MC (multi channel): ( $1 \leq ChannelId \text{ AND } ChannelId \leq 99$ )
    - Special case: all channels (no selection on *ChannelId*)
- **Problem until COOL 2.1.1 included**
  - Retrieval time is longer for IOVs at the end of the IOV table
  - ( $Since \leq 20 \text{ AND } 20 < Until$ ) OR ( $20 < Since \text{ AND } Since \leq 30$ )
  - Same problem as for single-channel case in COOL 2.0.0

- **Fixed in COOL 2.2.0**

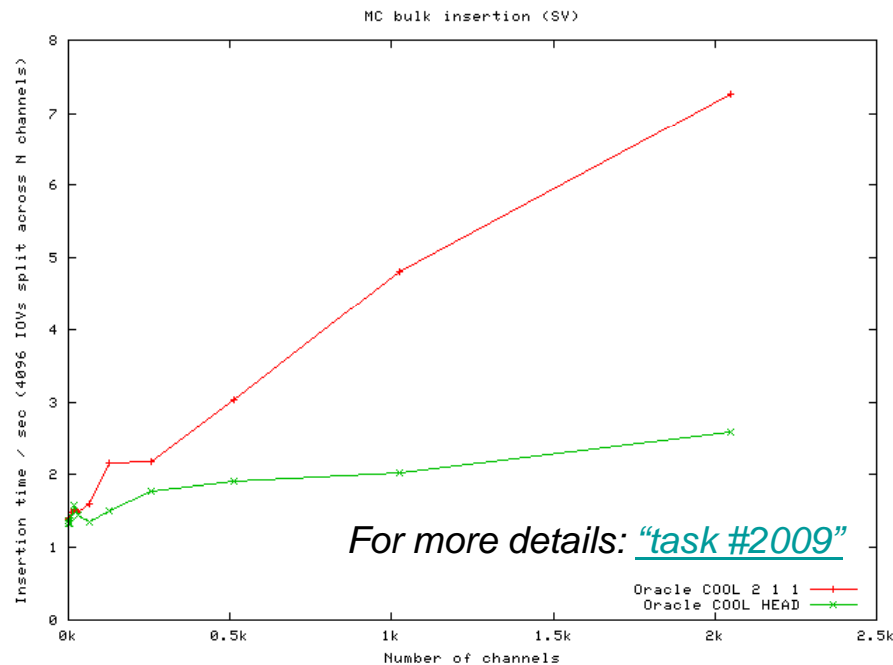
- Optimize lookup of first IOV for each channel as in fix for SC case
  - With *Max(Since)* subquery
- Loop over selected channels via a join on the IOV and channel tables
  - Execution plan (table order in join) depends on first value used (“*bind variable peeking*”): fix it using hints, `/*+ LEADING(c i) USE_NL(c i) */`



- **User requirement (limited client memory)**
  - Before fetching all data for a given IOV selection, users can ask how many IOVs would be returned
- **Problem in COOL 2.1.1**
  - Separate SQL statements for the ‘select rows’ and the ‘select count(\*)’ associated to it
- **Improvement in COOL 2.2.0**
  - New internal infrastructure makes it easy to simply ‘select count(\*) from ( select rows... )’
  - Not yet applied consistently to all use cases



- **Example: insert data for t=[20,30] in channels 1-99**
  - SV (single version): there is only one version at any time t
    - At insertion time, this implies “extra checks” (selects, updates...)
  - MC (multi channel): insert data for many channels in one go
  - *Bulk insertion: group all channels in a single SQL statement*
- **Problem until COOL 2.1.1 included**
  - IOV table insertion is a single SQL statement for all channels
  - But the “extra checks” use # SQL proportional to # channels



- **Fixed in COOL 2.2.0**

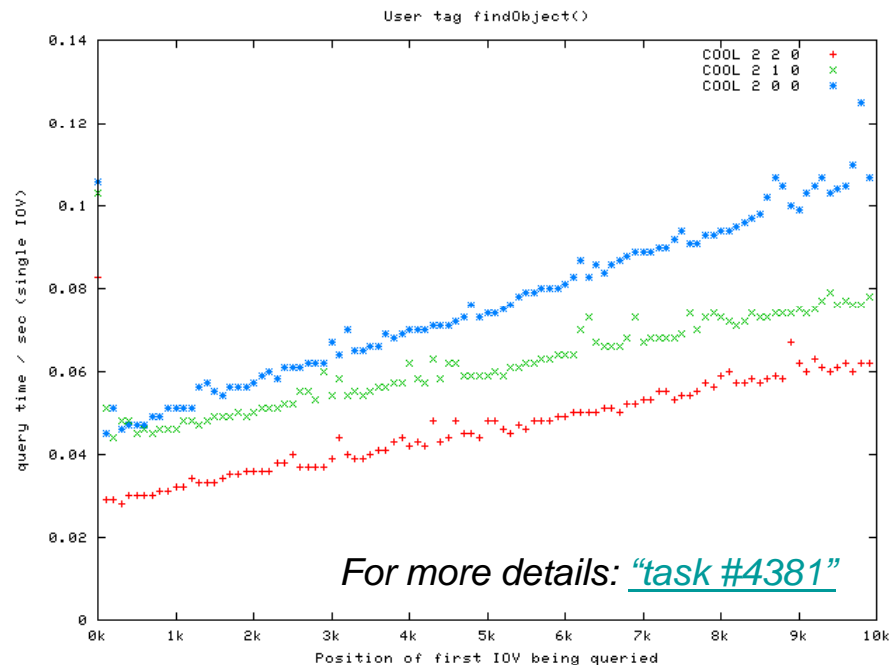
- Optimize lookup of first IOV *for each channel* as in fix for SC case
  - With *Max(Since)* subquery
- Loop over selected channels via a join on the channels table
  - Execution plan depends on table order in the join: fix it using hints, `/*+ LEADING(c i) USE_NL(c i)*/`

- **If retrieving  $n$  IOVs takes  $x$  seconds, what % of this time are spent in OCI, CORAL and COOL?**
  - Goal is to make COOL a thin client (minimal overhead)
- **Problem until COOL 2.1.1 included**
  - Old API returns boost *shared pointers* to the data: by construction, this triggers copies of the CORAL data
  - Example: 5.9s to fetch 200k IOVs (SV MC browse)
    - Oracle OCI and CORAL ~1.8s, COOL overhead ~4.1s
    - The 1.8s is low thanks to the server-side optimizations we described before (5.9s is COOL 2.2.0 using the 2.1.1 API)
- **Fixed in COOL 2.2.0 (API extensions)**
  - New API returns *const references* to the data: implemented by returning wrappers of the CORAL data (no extra copies)
  - Example: 2.3s to fetch 200k IOVs (same SV MC browse)
    - Oracle OCI and CORAL ~1.8s, COOL overhead ~0.5s
    - *Total time to fetch IOVs reduced by almost a factor 3*
    - *COOL overhead reduced from 200% to <30% of OCI/CORAL*

- **Cache the node table for R/O connections**
  - Fixed in COOL 2.1.1
- **Prevent the opening of multiple cursors on the database server at the same time**
  - Fixed in COOL 2.1.1
- **Use the channels table to list channels**
  - Fixed in COOL 2.2.0 for the SV case
  - Still pending for the MV channels table

- **SQL optimization of other use cases**
  - MV cases (involve other tables, e.g. IOV2TAG)
    - Three different cases: HEAD, tag, “user tag”
  - Open ( [1,∞], [2,∞]...) vs. closed IOVS ( [1,2], [2,3]...)
  - In general: use the same C++ and SQL code to handle different use cases (one fix to fix them all)
  - *COOL flexibility (many use cases) has a price*
- **Large scale (distributed) stress tests**
  - Many such tests are underway (COOL team and experiment users) but were not described here
- **Backends other than Oracle?**
  - Many of the optimizations described in this presentation do not apply to MySQL and SQLite

- **Example: get the single IOV at t=20 for channel 5, amongst IOVs inserted with user tag “My tag”**
  - MV (multi version): within the set of IOVs with a given tag, there is only one version at any time t
    - “User tag” is one of three different types of MV tag queries
- **Problem currently exists in COOL 2.2.0**
  - Retrieval time is longer for IOVs at the end of the IOV table



- **Improvement in COOL 2.1.0**
  - A 5-dimensional index on the most relevant columns was added
  - At the time it seemed that the issue had been solved (flat query time), but actually it has not
    - Most likely, optimization will require that the SQL query be changed
- **One of the many open issues!**

- **Major performance optimizations have been achieved since CHEP'06**
  - Server-side optimizations for many use cases
  - Client-side profile improvements as well
- **Performance optimization is not over**
  - Most multi-version use cases are still missing
  - *Highest priority for next phases of development*
- **Collaboration with DBAs is essential**
  - *Many thanks to the CERN IT-PSS Physics Database Service team and the ATLAS DBAs!*