OCTOBER 25, 2023

Creating a high-performing data archive

Greg Holt and Philipp Sick

AVEVA

Greg Holt

Staff Software Developer

- AVEVA
- gregory.holt@aveva.com

Philipp Sick

Senior Product Support Engineer

- AVEVA
- philipp.sick@aveva.com

Creating a high-performing data archive

Challenge

- A high-performing data archive is critical to the health of my AVEVA PI System
- Need to be able to read/write data consistently, with minimal downtime or disruptions

Solution

- Appropriate hardware sizing
- Tuning parameter and configuration tweaks, as needed
- System administration tasks
- Take steps to mitigate potential impact of expensive operations

Benefits

- Ability to read and write data as needed, without disruption
- Take full advantage of my AVEVA PI System's capabilities

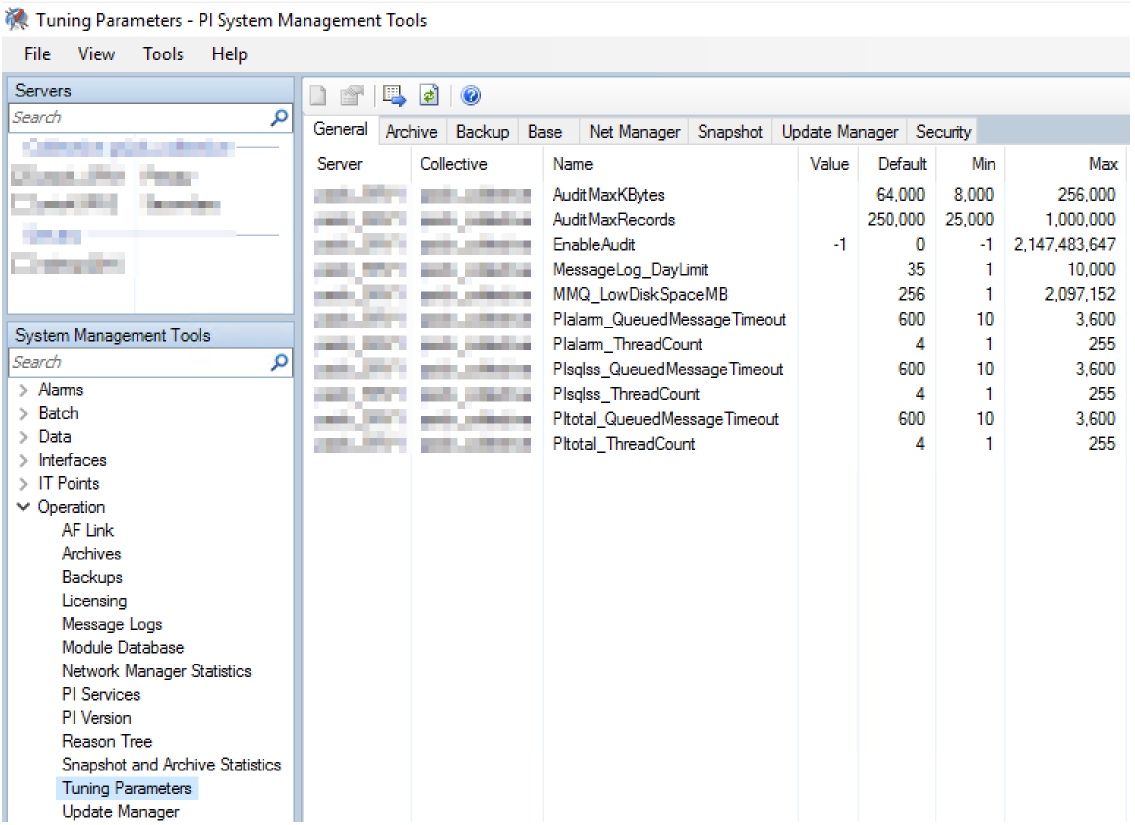
Hardware

System topologies

- Creating a high-performing data archive starts with allocating the appropriate hardware
  - How many cores do I need? How much RAM? How much storage space, and how many drives?
- We have created system topologies to meet your needs (e.g., data ingress/egress rates)
- For new AVEVA PI Systems, your Customer Success Manager and Account Manager will help you with sizing
- For existing systems, reach out to technical support for sizing questions

Tuning parameters and hardware

- Recall: Tuning parameters are configuration settings for the data archive



AVEVA

Tuning parameters and hardware

- **piarchss_ThreadCount**
  - Depending on the number of processors on the machine, this value may be increased so more RPC requests can be handled simultaneously. If all the threads are busy, RPCs are queued up and processed in chronological order
  - Two recommendations
    - Large queries, no real-time users – 2X number of core
    - Standard use-cases – Number of cores – 1
  - Ensure your applications and clients can make use of the additional threads

- **Archive_AutoArchiveFileSize**
  - Specifies the size of the new primary archive when an automatic archive create shift occurs
  - Will be relevant to Windows File System Cache discussion (Stay tuned!)

Other hardware considerations

Hardware of physical storage

- Lower latency storage has a greater impact on performance queries than having more IOPS
- Avoid complicated storage configurations with physical storage daisy changed in iSCI

Hardware considerations of client machines

- Consider capabilities of client machines

Network considerations

- Latency between client and server machines

AVEVA

How to efficiently use your data archive

High availability

- Data archive high availability solution: PI Collective
- Secondary data archive is essentially a copy of the primary data archive (we replicate points and point configuration, security settings, e.g., mappings and trusts, etc.)
- Main use case: minimize downtime
  - If the primary data archive has a downtime event, the PI Analysis service can connect to a secondary
- Performance benefits?

AVEVA

High availability

- Client connection balancing introduced with AF Client 2018
  - Client connections will now be automatically distributed to all available collective members, instead of just the primary
  - After an HA failover, failback switches the client connection back to the original member
- Can point expensive clients/queries at secondary, while other users connect to primary
- HA may reduce the load on each server, but how else can we improve read performance?

Windows file system cache

- An OS feature that improves performance for file access
- When a file is read from physical disk, the OS will cache 256 KB sections of the file in memory
- The next time the file is read or written to, the operation will be much faster than the initial access as it will not have to read from disk
    - Recommendation is to have enough RAM to be able to fit three archives into memory; archive file size should be no larger than a third of the system’s RAM

Archive read cache

- In addition to the Windows File System Cache, PI Archive Subsystem leverages its own read cache
- Best use case for the archive read cache is for summary type calculations: you have a source point and are constantly doing different calculations or doing running totals, so you keep retrieving the same data over and over for that point
- Relevant tuning parameters:
  - Archive_CacheRecordsPerPoint

AVEVA

Archive reprocessing

- Archive reprocessing is a user-initiated function of archive editing that reorganizes and compacts archive files
- Reprocessing may improve performance of deep queries (long-time-range query for a small number of points)
    - Need to set Archive_ReprocessThreadCount tuning parameter to 1 for PI Data Archive 2018 SP2 and later

Archive reprocessing

- Other benefits:
  - Defragment archive file
  - Recover space from deleted points
  - Coerce archive data to current PointType
- Caveats:
  - Does not improve performance for wide queries (large number of points over small time range)

AVEVA

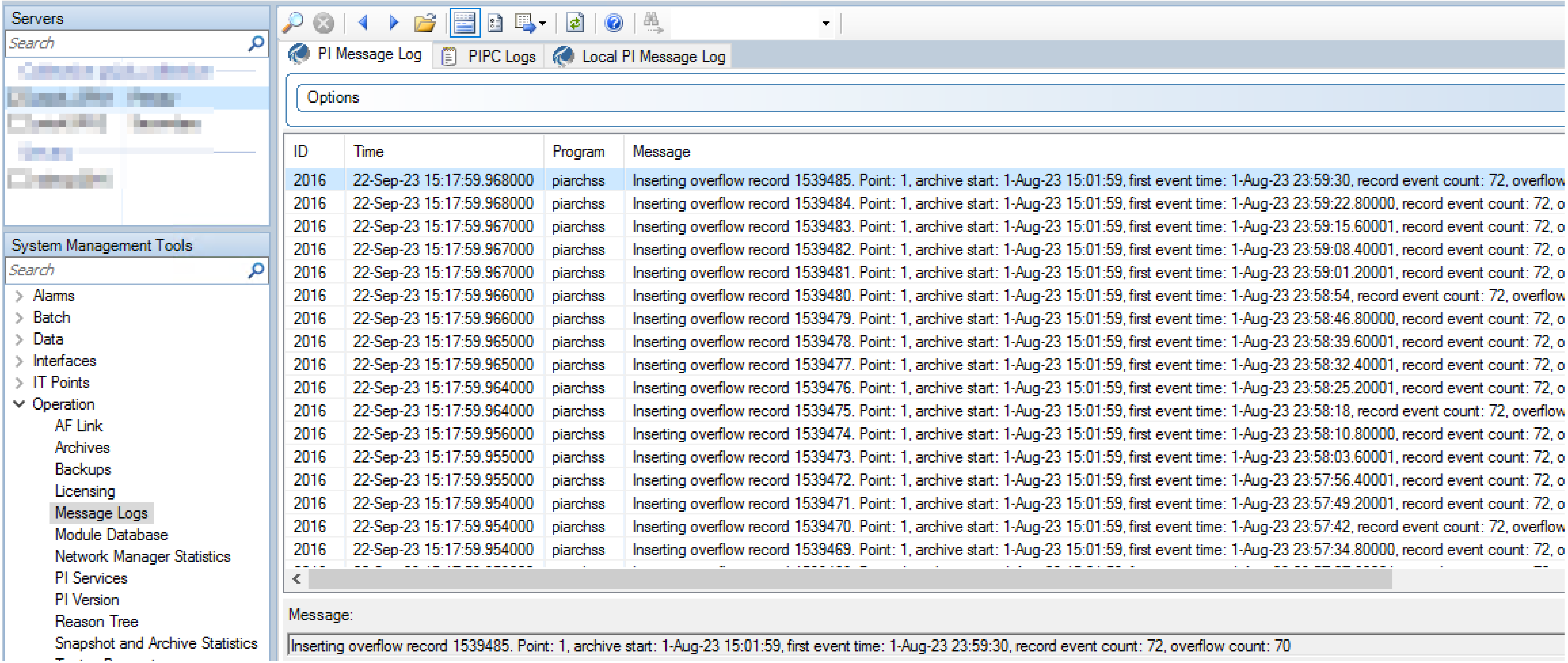
Common performance issues

Backfilling

- One of the most common causes of data archive performance issues is writing out-of-order data
  - Causes: analysis recalculation, interface run in history recovery, e.g., PI to PI, RDBMS, UFL
- Symptoms:
  - High CPU or memory usage
  - Data archive operations are slow or time out
- Tell-tale sign: PI Message Log shows messages with ID (2016), “Inserting overflow record”

AVEVA

Backfilling



ID	Time	Program	Message
2016	22-Sep-23 15:17:59.968000	piarchss	Inserting overflow record 1539485. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:59:30, record event count: 72, overflow
2016	22-Sep-23 15:17:59.968000	piarchss	Inserting overflow record 1539484. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:59:22.80000, record event count: 72, o
2016	22-Sep-23 15:17:59.967000	piarchss	Inserting overflow record 1539483. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:59:15.60001, record event count: 72, o
2016	22-Sep-23 15:17:59.967000	piarchss	Inserting overflow record 1539482. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:59:08.40001, record event count: 72, o
2016	22-Sep-23 15:17:59.967000	piarchss	Inserting overflow record 1539481. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:59:01.20001, record event count: 72, o
2016	22-Sep-23 15:17:59.966000	piarchss	Inserting overflow record 1539480. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:54, record event count: 72, overflow
2016	22-Sep-23 15:17:59.966000	piarchss	Inserting overflow record 1539479. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:46.80000, record event count: 72, o
2016	22-Sep-23 15:17:59.965000	piarchss	Inserting overflow record 1539478. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:39.60001, record event count: 72, o
2016	22-Sep-23 15:17:59.965000	piarchss	Inserting overflow record 1539477. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:32.40001, record event count: 72, o
2016	22-Sep-23 15:17:59.964000	piarchss	Inserting overflow record 1539476. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:25.20001, record event count: 72, o
2016	22-Sep-23 15:17:59.964000	piarchss	Inserting overflow record 1539475. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:18, record event count: 72, overflow
2016	22-Sep-23 15:17:59.956000	piarchss	Inserting overflow record 1539474. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:10.80000, record event count: 72, o
2016	22-Sep-23 15:17:59.955000	piarchss	Inserting overflow record 1539473. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:58:03.60001, record event count: 72, o
2016	22-Sep-23 15:17:59.955000	piarchss	Inserting overflow record 1539472. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:57:56.40001, record event count: 72, o
2016	22-Sep-23 15:17:59.954000	piarchss	Inserting overflow record 1539471. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:57:49.20001, record event count: 72, o
2016	22-Sep-23 15:17:59.954000	piarchss	Inserting overflow record 1539470. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:57:42, record event count: 72, overflow
2016	22-Sep-23 15:17:59.954000	piarchss	Inserting overflow record 1539469. Point: 1, archive start: 1-Aug-23 15:01:59, first event time: 1-Aug-23 23:57:34.80000, record event count: 72, o

Mitigating the cost of backfilling

- Reduce OOO operations
- Reduce expensive read operations
- Perform global lock operations on off-hours: Tag creation, edits, deletes, etc.
- Reduce use of:
  - String tag usage
  - Blob tag usage
  - Float64 events
  - Sub-second timestamps
  - Annotations
- Reduce archive file duration
- Restructure OOO writes to be in-order with respect to each archive file
- Be mindful of auditing

AVEVA

Heavy hitters

- Heavy hitter: A client application/user that is performing expensive queries, affecting the overall performance of the data archive
- Symptoms:
  - High CPU or memory usage
  - Data archive operations are slow or time out
- The query may involve:
  - Long time range
  - Many tags involved
  - Dense data
  - Data type, e.g., string

AVEVA

Heavy hitter troubleshooting techniques

1. Check PI Message Log for relevant messages
2. Take thread dumps to identify long duration RPCs
3. Take Connection ID gathered from steps 1 or 2 to find the offending client connection

Heavy hitter troubleshooting techniques

Relevant messages in the PI Message Logs

- Reminder: Read PI Message Log via PI System Management Tools > Operation > Message Log, the PI SDK Utility, or the command line tool pigetmsg
- Relevant messages include:

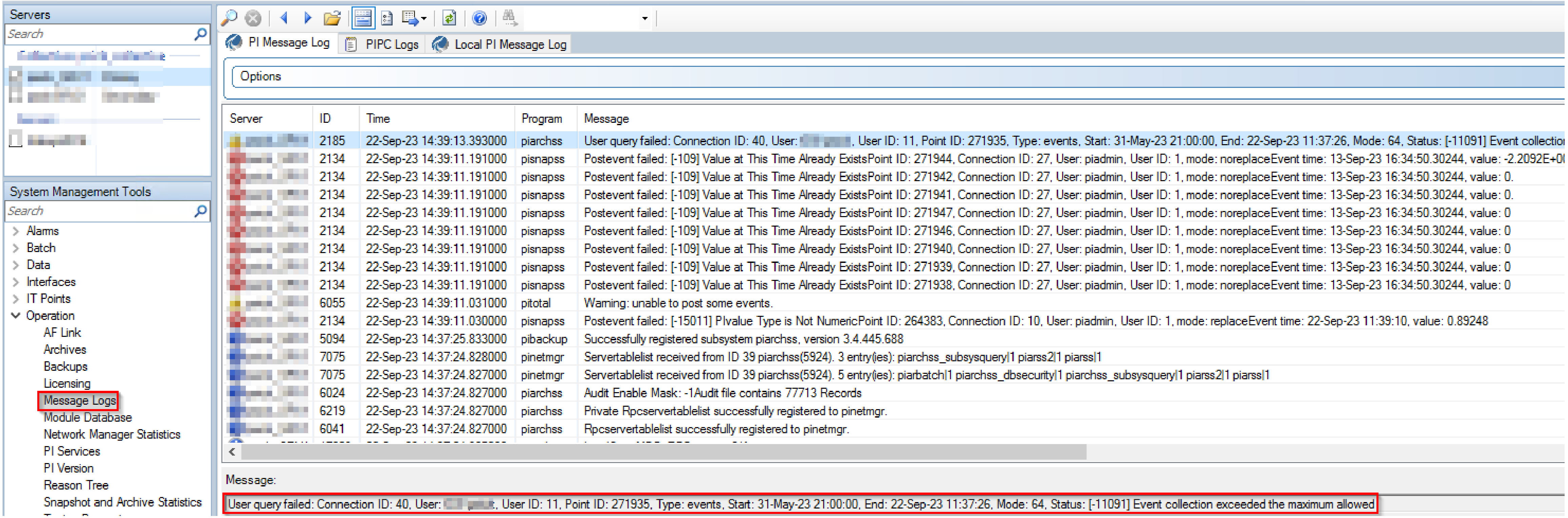
```
[-11091] Event collection exceeded the maximum allowed

[-11140] Archive query exceeded maximum execution time (see Archive_MaxQueryExecutionSec)

[-10767] Client exceeded maximum concurrent queries in RPC thread pool
```

Heavy hitter troubleshooting techniques

Relevant messages in the PI Message Logs



Heavy hitter troubleshooting techniques

Relevant messages in the PI Message Logs

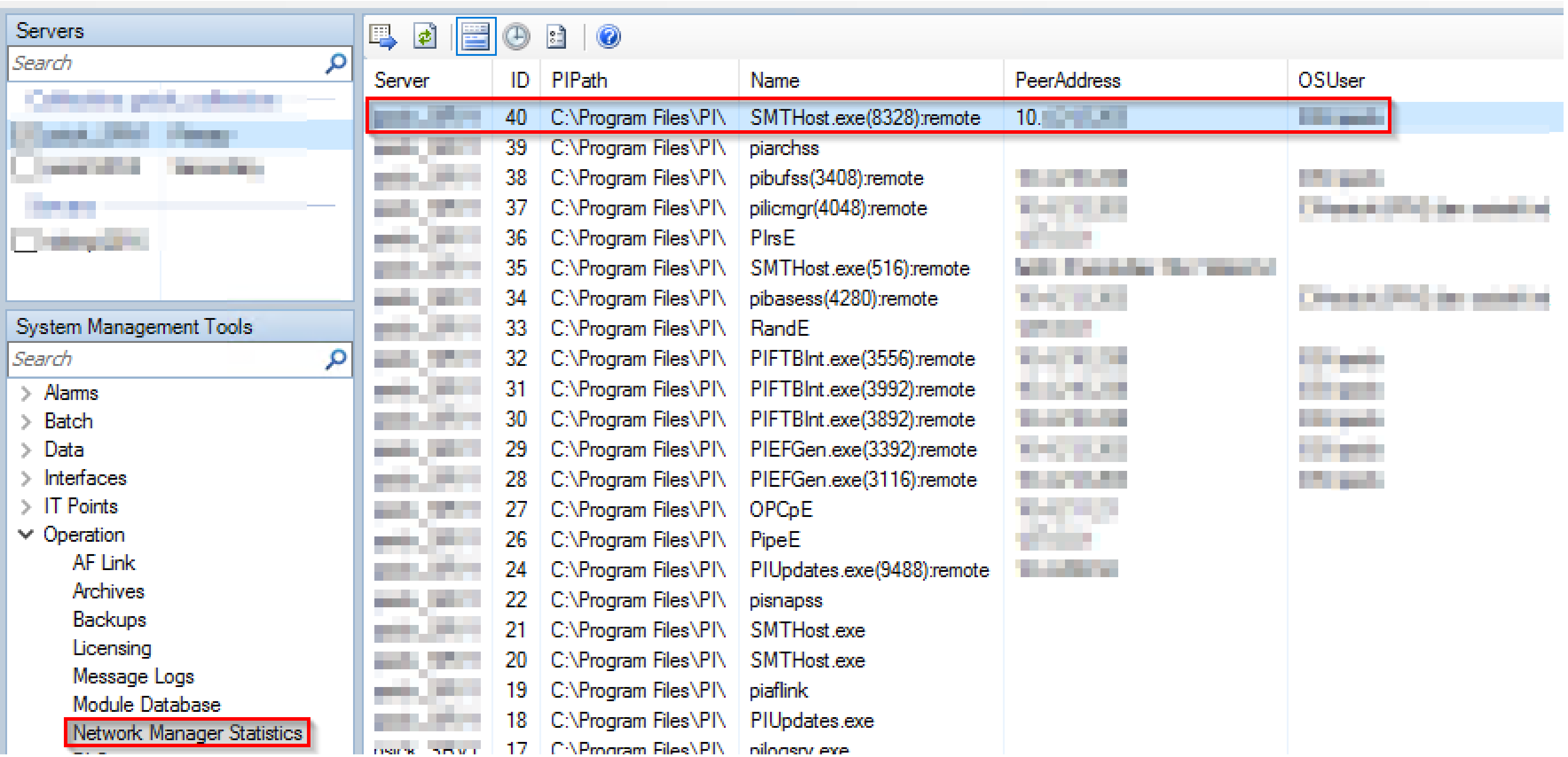
- Message will include Connection ID, which can be used to identify client
  - PI System Management Tools > Operation > Network Manager Statistics

`User query failed:` **`Connection ID: 40`**`, User: <user>, User ID: 11, Point ID: 271935, Type: events, Start: 31-May-23 21:00:00, End: 22-Sep-23 11:37:26, Mode: 64, Status: [-11091] Event collection exceeded the maximum allowed`

AVEVA

Heavy hitter troubleshooting techniques

Relevant messages in the PI Message Logs



AVEVA

Heavy hitter troubleshooting techniques

Take thread dumps to identify long duration RPCs

- Very simplified definition of a thread for our purposes: a worker that executes the tasks given to a program by other threads or other processes

- Example:
  - A ProcessBook display needs to load a trend with data, so it'll send the request to the PI Archive Subsystem
  - PI Archive Subsystem will assign the specific task - GetArcEvents - to a thread

Heavy hitter troubleshooting techniques

Take thread dumps to identify long duration RPCs

- We can take thread dumps of a subsystem, which will tell us (amongst other things):
  - The active RPC threads, their duration, and the Connection ID of the client
- Command Line: navigate to the PI\adm directory
  - `cd /d %piserver%adm`
- Thread dump command:
  - `piartool –thread piarchss –info`

Heavy hitter troubleshooting techniques

Take thread dumps to identify long duration RPCs

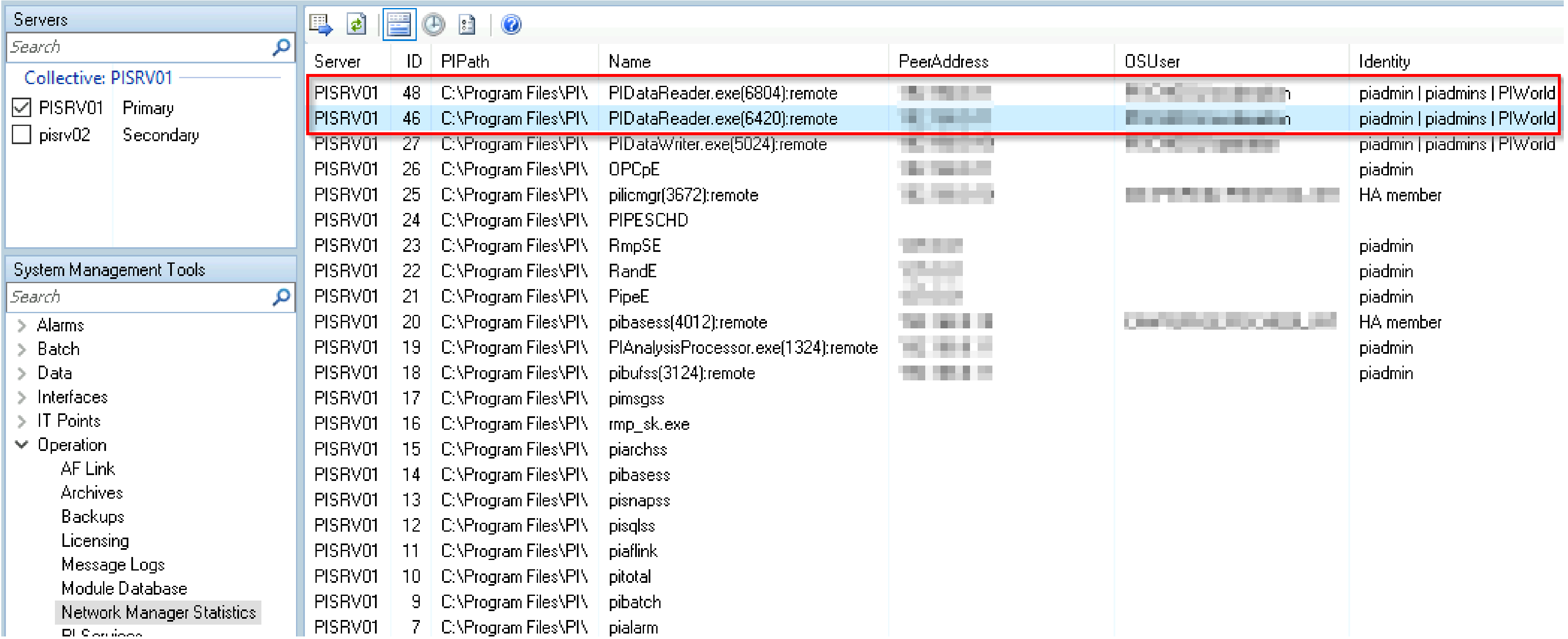
Output (simplified):

```
RPC|MaxNumberOfThreads:8|CurrentThreads:8
TID, TimeInQueue, StartTime, Duration, TaskName, ConnectionID, RPC Extras
2544, 599983, 19-Sep-23 21:18:27.44385, 143318, piarss|1|getarcevents, 46, PointID: 1.
1432, 599454, 19-Sep-23 21:18:36.06366, 134698, piarss|1|getarcevents, 46, PointID: 1.
5612, 599475, 19-Sep-23 21:19:47.08025, 63682, piarss|1|getarcevents, 48, PointID: 1.
5336, 599384, 19-Sep-23 21:19:56.12967, 54632, piarss|1|getarcevents, 48, PointID: 1.
3520, 599459, 19-Sep-23 21:19:57.23048, 53532, piarss|1|getarcevents, 48, PointID: 1.
2572, 599597, 19-Sep-23 21:20:28.11984, 22642, piarss|1|getarcevents, 46, PointID: 1.
4052, 599150, 19-Sep-23 21:20:39.08138, 11681, piarss|1|getarcevents, 46, PointID: 1.
5528, 599955, 19-Sep-23 21:20:50.76207, 0, piarss|1|getarcevents, 48, PointID: 1.
2132, 99, 19-Sep-23 21:20:50.76237, 0, piarchss_subsysquery|1|ThreadControl, 81, Action: Live.
```

- Duration is in milliseconds: > 1000 ms (1 second) is generally long
- Use Connection ID of long duration RPC to identify heavy hitter
  - PI System Management Tools > Operation > Network Manager Statistics

Heavy hitter troubleshooting techniques

Take thread dumps to identify long duration RPCs



Reduce the risk of a heavy hitter

Set tuning parameters:

- ArcMaxCollect
  - Can limit the number of compressed events that can be retrieved by a single query for a given client
  - Default value: 1.5 million
- Archive_MaxQueryExecutionSec
  - Limits how long archive queries can run
  - Default value: 260 seconds

AVEVA

PI Point creation/deletion

- Creating (or deleting) a PI Point is a relatively expensive process—Why?
- Three files that store point information are involved:
  - PI Point Table
  - PI Snapshot Table
  - Primary archive file
- All three must be updated when a point is created or deleted!
  - Entails a global lock

PI Point creation/deletion

- Be mindful of bulk point creation:
  - PI Builder
  - AF Analyses
- Best to schedule during off-hours
- Can keep eye on PI Message Log for point creation messages:

```
I 01-Sep-23 21:50:38 pibasess:Point Table
(6079)

 >> Point [Name: <Point name>, ID: <PointID>] - Created by user piadmin (userid: 1, cnxnid: <Connection
ID>)
```


AVEVA

Expression-based queries

- Q: What are expression-based queries?
- A: Vision/DataLink /ProcessBook calculations that are executed on the data archive
  - Starting with AVEVA™ PI Vision™ 2020, we can perform simple mathematical expressions on PI Points (or AF Attributes) on demand
  - Includes arithmetic calculations and summary calculations (minimum, maximum, and average) on a per display basis
- When several of these queries are being run at the same time, and CPU usage is high, Archive Subsystem RPC threads may become bottlenecked
- If point creations/deletions are being done at the same time, PI Base Subsystem can be affected as well

Expression-based queries

- AVEVA PI Vision calculation usage report: PI Vision Administration website > Reports > Calculation usage information

Name	Description	Expression	Context	Interval	Display ID	Display Name	Use Count	Error Message
LOAD_PREVIOUSMONTH	Average Monthly Load	TagAvg('LOAD.MWB', '* - 1 mo', '*')	pireg	1s	227	Pump Display	1	
Sinsuoid Doubled	Sinsuoid Doubled	'Sinusoid'*2	pireg	1s	227	Pump Display	1	

- Avoid costly calculations, e.g., a tag average over a long duration, that executes at a high frequency
- For more complex calculations, configure PI Analyses instead

Stale tags

- Stale tag: point that has not updated in some time
- Can result in lower performance as we need to navigate through many archives, like when:
  - Requesting interpolated events for stale tags
  - Deleting the snapshot value for a tag with no historical data (i.e., only “Pt Created”)

AVEVA™ PI Server 2024

AVEVA PI Server 2024

AVEVA PI Server 2024 will address several performance bottlenecks

- Driven by customer feedback

Showcasing a few examples of changes forthcoming:

- Improving out-of-order data scenarios
- Improving archive read cache
- Optimizing attempts to write events to read-only archives

AVEVA PI Server 2024

Improving out-of-order data scenarios

- Out-of-order data can occur for a variety of reasons
- Out-of-order data can significantly impact PI data archive performance

Reduces amount of time required to insert larger bursts of out-of-order events due to

- Recalculations performed by PI Analytics
- Backfilling data via interfaces, connectors and adapters

Impact

- Seen up to 40% improvements for numeric types and up to 90% for strings and blobs
- Your mileage may vary

AVEVA

AVEVA PI Server 2024

Improving archive read cache

Significant work to optimize

- Improved query performance
- Performing cache eviction

Improves queries which span larger event counts

Properly handles larger cache sizes

- Performance doesn't degrade with larger caches like previous versions did

AVEVA PI Server 2024

Optimizing attempts to write events to read-only archives

- Backfilling from older systems
- PI Analytics performing recalculations on older data

Attempting to write 10M events to read only archives

Current releases	AVEVA PI Server 2024
90 seconds	16 seconds

AVEVA

Questions?

Please wait for the microphone.

State your name and company.

Please remember to...

Navigate to this session in the mobile app to complete the survey.

Thank you!

AVEVA

This presentation may include predictions, estimates, intentions, beliefs and other statements that are or may be construed as being forward-looking. While these forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could result in actual outcomes differing materially from those projected in these statements. No statement contained herein constitutes a commitment by AVEVA to perform any particular action or to deliver any particular product or product features. Readers are cautioned not to place undue reliance on these forward-looking statements, which reflect our opinions only as of the date of this presentation.

The Company shall not be obliged to disclose any revision to these forward-looking statements to reflect events or circumstances occurring after the date on which they are made or to reflect the occurrence of future events.

AVEVA

linkedin.com/company/aveva

@avevagroup

ABOUT AVEVA

AVEVA is a world leader in industrial software, providing engineering and operational solutions across multiple industries, including oil and gas, chemical, pharmaceutical, power and utilities, marine, renewables, and food and beverage. Our agnostic and open architecture helps organizations design, build, operate, maintain and optimize the complete lifecycle of complex industrial assets, from production plants and offshore platforms to manufactured consumer goods.

Over 20,000 enterprises in over 100 countries rely on AVEVA to help them deliver life's essentials: safe and reliable energy, food, medicines, infrastructure and more. By connecting people with trusted information and AI-enriched insights, AVEVA enables teams to engineer efficiently and optimize operations, driving growth and sustainability.

Named as one of the world's most innovative companies, AVEVA supports customers with open solutions and the expertise of more than 6,400 employees, 5,000 partners and 5,700 certified developers. The company is headquartered in Cambridge, UK.

Learn more at www.aveva.com

AVEVA