

# Archiver Hypertable Engineed (HyperArchiver)

**Mauro Giacchini**

**Epics Collaboration Meeting – Data Archiving round table  
13/10, 2010**

# Target

Make the EPICS Archiver embedded DB more reliable and safe especially in large application with a large number of PVs to store

First Approach: re-write the Rtree and double-linked-list embedded DB structure

Second approach: look for a complete replacement of the embedded DB with new one

# Why not Beauty ? (Best Ever Archiver Utility)

Beauty = Classical Archiver over Oracle DB

Pros:

SQL statements accept  
Professional Solution

Cons:

Not so fast

Really expensive (licenses)



# Why not MySQL ?

Potential scalability concerns

Designed for a single machine (not distributed)

What it takes to make it scale

**Major** engineering effort

Solutions are usually ad hoc

Solutions usually involve horizontal partitioning + replication

Solutions involve expensive hardware



# BigTable ?

Proprietary DB used by Google over their proprietary Google File System

High Performance

Large amount of data

Stable

Reliable

Distributed



Hypertable has been developed as an in-house software at Zvents Inc. In January 2009, Baidu, the leading Chinese language search engine, became a project sponsor. Rediff.com is one of the premier worldwide online providers of news, information, communication, entertainment and shopping service.

**Commercial support is available**

**Hypertable is licensed under the GNU General Public License Version 2.**

“Our goal is nothing less than that Hypertable become one of the world’s most massively parallel high performance database platforms.”

**Not RDB**



# Hadoop/Hypertable Solution Objective

- Handles applications with large datasets
- Fast fault detection and fail-over
- High throughput processing
- Centralized scheduling of tasks and execution of batch processes
- Can be deployed on low cost, commodity hardware
- Eliminates or reduces the need for expensive table joins



<b>Machines and time up</b> <b>First release of Hypertable</b>	<b>Insertion Time</b>	<b>Retrieval Time</b> (all pv parameters)
 Virtualbox virtual machine Debian Linux Stable x86 4 cores (shared among 3 VMs) 4GB RAM (1) 5.4k RPM SATA Disk \$4k (to 4 Vbox) (1/2 h data acq SNS)	<ul style="list-style-type: none"> <li>•1 min 31 sec</li> <li>•1 777 600 sample</li> <li>•19.5K samples/sec</li> </ul>	<ul style="list-style-type: none"> <li>•0.042 sec to extract</li> <li>•133K samples of 4 pv</li> <li>•3.17M samples/sec</li> </ul>
 Debian Linux Stable AMD64 - 8 cores 64GB RAM (7) 10k RPM SAS Disks (RAID 5) \$11k (1/2 h data acq SNS)	<ul style="list-style-type: none"> <li>•1min 13.6 sec</li> <li>•1 777 600 samples</li> <li>•24K samples/sec</li> </ul>	<ul style="list-style-type: none"> <li>• 0.024 sec to extract</li> <li>•133K samples of 4 pv</li> <li>•5.54 Msamples/sec</li> </ul>
 Debian Linux Stable AMD64 - 8 cores 64GB RAM (7) 10k RPM SAS Disks (RAID 5) \$11k (7dy data acq SNS)	<ul style="list-style-type: none"> <li>•11.2 hours to insert</li> <li>•597M samples</li> <li>•14.8K samples/sec</li> </ul>	<ul style="list-style-type: none"> <li>•0.031 sec to extract</li> <li>•133K samples of 4 pv</li> <li>•4.3M samples/sec</li> </ul>
 Debian Linux Stable AMD64 - 8 cores 64GB RAM (7) 10k RPM SAS Disks (RAID 5) \$11k (30dy data acq SNS)	<ul style="list-style-type: none"> <li>•37.86 hours</li> <li>•2557M samples</li> <li>•18.7K samples/sec</li> </ul>	<ul style="list-style-type: none"> <li>•0.030 sec to extract</li> <li>•133K samples of 4 pv</li> <li>•4.3M samples/sec</li> </ul>

# Oracle Vs Hypertable

Oracle **insertion** time : peak in write test (8Ksamples/sec)

Hypertable **insertion** time: 1.777.600 per 1min 13 sec (24Ksamples/sec)

Oracle **retrieve** time: 36.657 samples in 79.294 sec (~462 vals/sec)

Hypertable **retrieve** time: 57624 samples in 0.36 sec ( 160K vals/sec) in half hour

## THE OUR GOAL (hypertable 0.9.3.3 and 0.9.4.0)

- **storing:** 10000 channels per second
- **retrieve:** 1000 samples for up to 4 signal in less than 1 second

Ubuntu 9.10 karmic Kola Intel Pentium M 740 1,73Ghz 2GB RAM 5,6K RPM sata Disk Hypertable 0.9.3.3 the new release is 0.9.4.1

	Hypertable
Insertion Time	<ul style="list-style-type: none"><li>•20 min</li><li>•11999173 sample</li><li>•42 K sample/sec</li></ul>
Retrieval Time	<ul style="list-style-type: none"><li>•1328395 sample</li><li>•8k sample/sec</li></ul> <p><b>WORSTCASE</b> 2,6 ksample/s</p>

# TIME Retrieval w/o CSS

- Is not stable, some time the extraction don't have good performance ( Probably resolved in the new release 0.9.4.1)

1132 sample: 30 ms

1331 sample: 39 ms

1332 sample: 14 ms

1331 sample: 16 ms

**53,8 ksample/s**

# TIME Retrieval into CSS

Control System Studio (SNS)

File CSS Window Help

Archive Se... Navigator

URL: jdbc:mysql://localh... Info

Pattern: 2\* Search

○ Add... ● Replace: □ Reg.Exp.

PV Name	r
200:aiExample	r
200:calcCounter	r
201:aiExample	r
201:calcCounter	r
202:aiExample	r
202:calcCounter	r
203:aiExample	r
203:calcCounter	r
204:aiExample	r
204:calcCounter	r
205:aiExample	r
205:calcCounter	r
206:aiExample	r
206:calcCounter	r
207:aiExample	r

Value 4  
Value 3  
Value 2  
Value 1

Time

200:aiExample 200:calcCounter 206:aiExample 206:calcCounter

Properties

Trace	Item (PV, Formula)	Display Name	Color	Scan Perio	Buffer Size	Width	Axis	Trace Type	Request
<input checked="" type="checkbox"/>	200:aiExample	200:aiExample	Blue	0.0	5000	2	Value 1	Area	Optimized
<input checked="" type="checkbox"/>	200:calcCounter	200:calcCounter	Red	0.0	5000	2	Value 2	Area	Optimized
<input checked="" type="checkbox"/>	206:aiExample	206:aiExample	Green	0.0	5000	2	Value 3	Area	Optimized
<input checked="" type="checkbox"/>	206:calcCounter	206:calcCounter	Black	0.0	5000	2	Value 4	Area	Optimized

Not logged in

Create table archive (pv); ← There is the possibility to create compressed table too.

Only one column where insert this field as String:

**Pv-name,  
Timestamp,  
Severity,  
ID\_Group,  
Engine,  
SempleMode,  
SemplePeriod,  
Valuetipes,  
Valeue (Array Value)**

Row key

Pv-name

timestamp

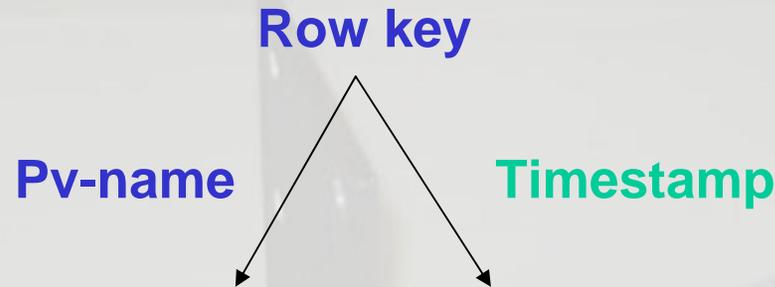
200:aiExample.1286519692093

pv

200:aiExample#c#LOW\_ALARM#c#1286519692093#c#MINOR#c#Test#c#jdbc:mysql://localhost:3306/archive#c#Monitor#c#1.0#c#double#c#4.0

Example of data output

# Row Key



**200:aiExample.1286519692093**

Example of data output

**200:aiExample.1286519692093**

**pv**

**200:aiExample#c#LOW\_ALARM#c#1286519692093#c#MINOR#c#Test#c  
#jdbc:mysql://localhost:3306/archive#c#Monitor#c#1.0#c#double#c#4.0**

Loris Giovannini, the student, who made the Java communications interfaces is closing his thesis with us and in looking to a contract... to complete this job to make the HyperArchive ready to a production system.

Please forward this info to the labs interested on this project, thanks !!

# Acknowledgments

- Kay Kasemir (ORNL)
- Doug Judd (Hypertable Core Team)
- Bob Dalesio, Ralph Lange, Robert Petkus (BNL)
- Thomas Birke (BESSY)

More information  
and Tar Ball

[www.inl.infn.it/~epics/](http://www.inl.infn.it/~epics/)