

DEVELOPMENT OF A NEW JAVA CHANNEL ACCESS LIBRARY JCAL

Hiroyuki Sako¹, Hiroshi Ikeda²

¹JAEA (Japan Atomic Energy Agency), ²VIC (Visible Information Center, Inc.)

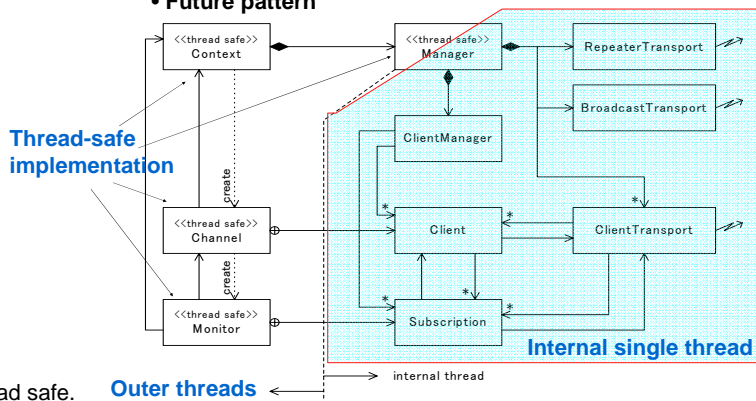
Abstract : Java channel access library JCA (Java Channel Access library) has been widely used for device control applications in Java. Especially for high-level applications in the J-PARC linac and RCS (Rapid-Cycling Synchrotron) control systems, which are unified in Java, the pure version of JCA implementation (CAJ) is desirable. However, JCA and CAJ have instability problems and vulnerability of the codes. To overcome the issues, a new compact Java channel access library, JCAL (Java Channel Access Light library) has been developed. A special care is taken to design the code architecture in order to keep thread safety and code robustness. The main part of the library is designed to work in a single thread, with the other threads for the monitor call-back. By adopting such a simple design, robustness and stability is realized. An adapter library for JCA API, JCA-JCAL, has been also implemented to plug in JCAL easily to existing Java applications using JCA-JNI and JCA-CAJ. Benchmark tests have been carried out and compared to JCA, which show comparable performance.

JCA Issues

- **Unstable behavior**
 - Connection lost
- **Thread unsafe**
 - Insufficient or inconsistent synchronizations
 - Anti-pattern : invoking wait method without a condition loop
 - Anti-pattern : starting thread in its constructor
- **Vulnerable internal structure**
 - **Broken encapsulation**
 - Returning mutable fields without defensive copy
 - Exposing the reference this in the constructor
 - **Strong inter-dependencies among packages**
 - Difficult to maintain and extend
- **Too concrete implementation in API**
 - **API should be more abstract**
 - For maintenance
 - User do not need to know implementation details
 - **Users are forced to use problematic implementations**
 - DBR (fundamental data structure) is not immutable nor thread safe.
 - Hard to repair it to keep backward compatibility

JCAL (Java Channel Access Light library)

- Single-threaded architecture for thread safety
- User thread can be multi-threaded
- Threads for monitor callback
- **Abstract API**
- **Dbr: immutable fundamental data**
- **Future pattern**



JCAL classes

Class	Functions
Context	Library environment
Channel	EPICS channel (API)
Monitor	EPICS monitor (API)
Manager	Manager to control the internal thread
ClientManager	Manager for Client and Subscription
Client	Inner class of Channel, accessed only by the internal thread
Subscription	Inner class of Monitor, accessed only by the internal thread
RepeaterTransport	Communication with Repeater (UDP/IP)
BroadcastTransport	Communication in broadcast (UDP/IP)
ClientTransport	Communication with server (TCP/IP)

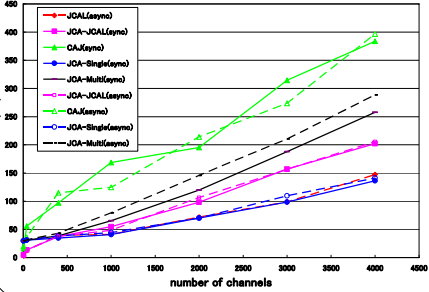
Benchmark Test

Environment A (ideal)
 Soft IOC: Dell dimension 4500C
 Pentium 4 2.4 GHz
 Client: Dell PowerEdge 830
 CPU: Pentium D 3 GHzx2
 Memory: 1GB

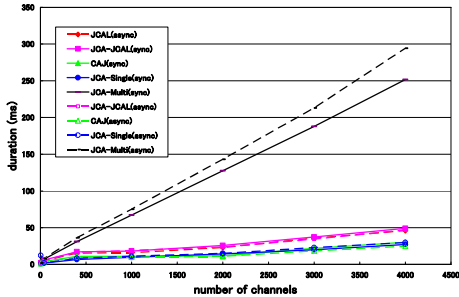
Environment B (realistic)
 Beam monitor IOCs: VME Advme7501AR
 Client: IBM ThinkPad Lenovo R60
 CPU: Genuine Intel T2300 1.7GHzx2
 Memory: 1GB

Test in Environment A

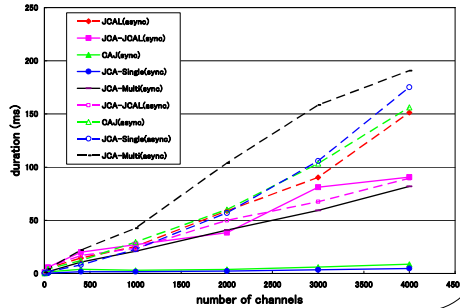
Connect test



Get test



Put test



environment	test	Average time / channel (msec)				
		JCAL	JCA-JCAL	CAJ	JCA-JNI (single)	JCA-JNI (multi)
A	connect	0.037	0.051	0.099	0.035	0.072
B	connect	0.31	0.34	1.01	0.59	0.63
A	get	0.011	0.012	0.0061	0.0074	0.074
B	get	0.040	0.071	0.026	0.031	0.101
A	put	0.038	0.022	0.029	0.044	0.048
B	put	0.16	0.37	0.15	0.15	0.18

Results

- JCAL is comparable with JCA-JNI (single-threaded) and CAJ.
- JCA-JNI (single-threaded) is fastest.
- JCA-JCAL is similar to JCAL
- JCA-JNI (multi-threaded) is slow.

Summary and Outlook

A new pure Java channel access library JCAL has been developed for Java applications in J-PARC control system. The design and implementation of JCAL is thread safe and robust. Benchmark tests show comparable performance with JCA-JNI and JCA-CAJ.