

J2EE Performance Scalability and Clustering *Part 2*

Emmanuel Cecchet
INRIA Rhône-Alpes, ObjectWeb

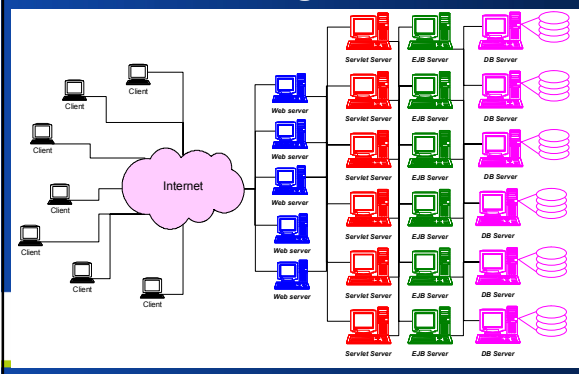


Outline

- J2EE Clustering
- C-JDBC
- Démo
- Conclusion



J2EE Clustering



J2EE Clustering

- Load balancing on Web Servers
 - hardware: L4-switch
 - software: One-IP techniques, RR-DNS, ...
- Web server to Servlet server
 - Load balancing with JK module (mod_jk)
 - Static weighted round-robin
 - Session affinity



J2EE Clustering

- Servlet/JSP server clustering
 - Tomcat in-memory session replication
 - failover ensured by mod_jk
- Servlet/JSP to EJB server
 - clustered JNDI
 - load-balancing and failover by cluster-aware stubs



J2EE Clustering

- EJB Server clustering
 - cluster stubs for load-balancing
 - transparent failover for idempotent methods
 - session-beans
 - stateless: no state to replicate
 - stateful: in-memory replication or database-based persistency
 - entity beans
 - usually database based persistency



J2EE Clustering

- Database clustering
 - Commercial offers
 - Oracle RAC (60.000\$ / cpu)
 - based on expensive SAN (Storage Area Network)
 - Open-source solutions
 - No real clustering
 - master/slave replication in MySQL
 - Postgres-R (still in alpha)
 - Clustered JDBC



Outline

- J2EE Clustering
- C-JDBC
- Démo
- Conclusion



C-JDBC motivations

- J2EE performance scalability bounded by database performance
- Large SMP are not commodity
- Database tier must be
 - scalable
 - fault tolerant (high availability + failover)
 - without modifying the client application
 - using open source databases
 - on commodity hardware



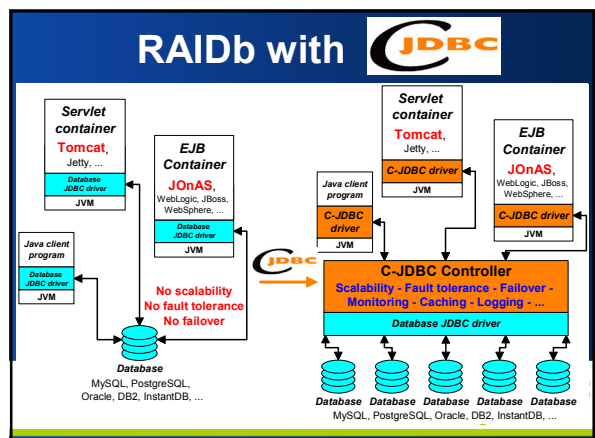
RAIDb

- Redundant Array of Inexpensive Databases (RAIDb)
 - better performance and fault tolerance than a single database,
 - at a low cost,
 - by combining multiple DB instances into an array of DB.
- RAIDb controller
 - gives the view of a single database to the client
 - balance the load on the database backends
- RAIDb levels
 - RAIDb-0: full partitioning (best performance)
 - RAIDb-1: full mirroring (best fault tolerance)
 - RAIDb-2: partial replication (best tradeoff)



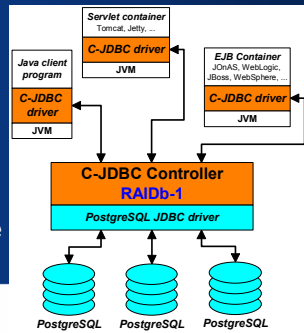
C-JDBC – Key ideas

- Middleware implementing RAIDb
- Two components
 - generic JDBC 2.0 driver (C-JDBC driver)
 - C-JDBC Controller
- C-JDBC Controller provides
 - performance scalability
 - high availability
 - failover
 - caching, logging, monitoring, ...
- Supports heterogeneous databases



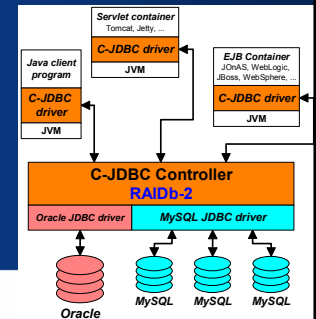
C-JDBC RAIDb-1 example

- no client code modification
- original PostgreSQL driver and RDBMS engine
- C-JDBC provides scalable performance and high availability

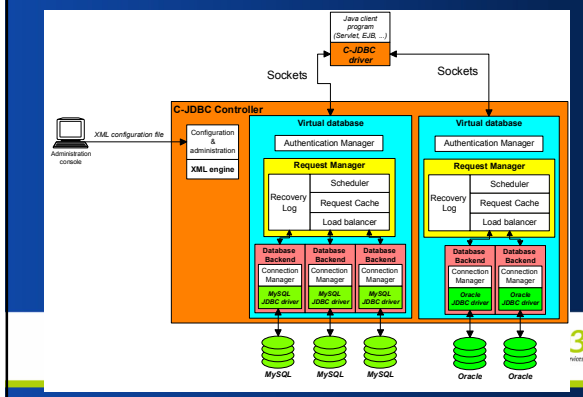


C-JDBC RAIDb-2 example

- unload a single Oracle DB with several MySQL
- add caching, fault tolerance, and monitoring for free



Inside the Controller

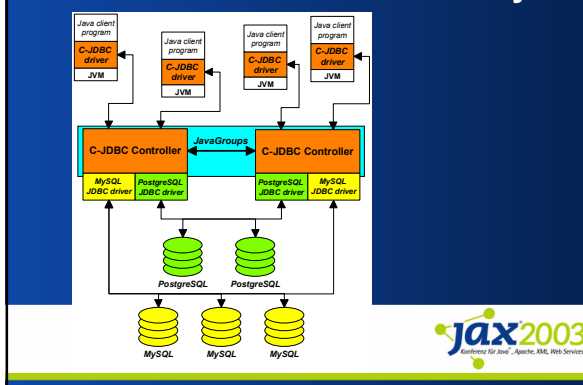


C-JDBC scalability

- Horizontal scalability
 - prevents the controller to be a Single Point Of Failure
 - distributes the load among several controllers
 - coherency ensured by group communications
- C-JDBC Driver
 - multiple controllers automatic failover
 - `jdbc:c-jdbc://node1:1099:c1,node2:1234:c2/myDB`
 - connection caching
 - URL parsing/controller lookup caching



C-JDBC – Horizontal scalability



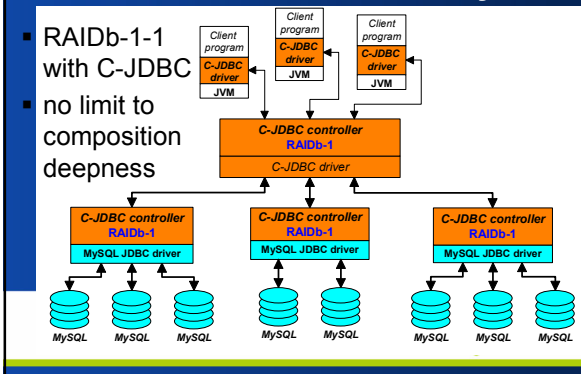
C-JDBC scalability

- Vertical scalability
 - allows nested RAIDb levels
 - allows tree architecture for scalable write broadcast
 - necessary with large number of backends
 - C-JDBC driver re-injected in C-JDBC controller



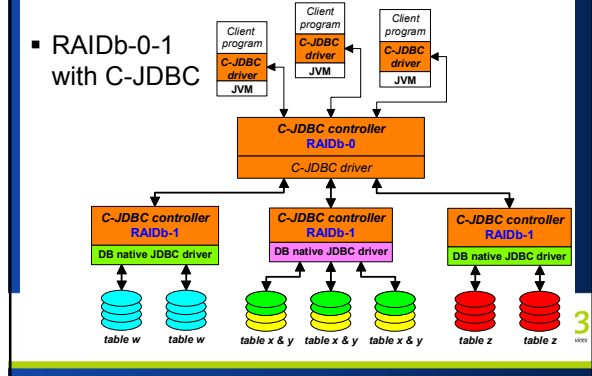
C-JDBC vertical scalability

- RAIDb-1-1 with C-JDBC
- no limit to composition deepness

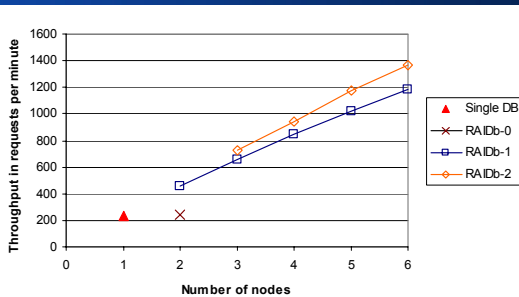


C-JDBC vertical scalability

- RAIDb-0-1 with C-JDBC



Performance - TPC-W



Fine-grain caching

- Cache hit rate with TPC-W
 - browsing mix

	Throughput	Response time	Hit rate
No cache	9.1 req/s	3.30s	
Table	12.9 req/s	1.96s	12.6%
Column	16 req/s	1.36s	48.8%
Column+ single-row	16 req/s	1.35s	49.2%

Fine-grain caching

- Cache hit rate with TPC-W
 - shopping mix

	Throughput	Response time	Hit rate
No cache	12.8 req/s	3.11s	
Table	13.5 req/s	2.58s	3.5%
Column	19.0 req/s	0.93s	30.0%
Column+ single-row	20.2 req/s	0.84s	30.4%

Outline

- J2EE Clustering
- C-JDBC
- Démo
- Conclusion

Outline

- J2EE Clustering
- C-JDBC
- Démo
- **Conclusion**



Conclusion

- J2EE Clustering
 - One-IP or RR-DNS
 - Apache + mod_jk
 - Tomcat + in memory session replication
 - EJB Server
 - distributed JNDI
 - cluster-aware stubs
 - C-JDBC + any RDBMS



Questions ?

- **Apache/Tomcat** : <http://www.apache.org>
- **JOnAS**: <http://www.objectweb.org/jonas>
- **CJDBC** : <http://c-jdbc.objectweb.org>



Bonus slides



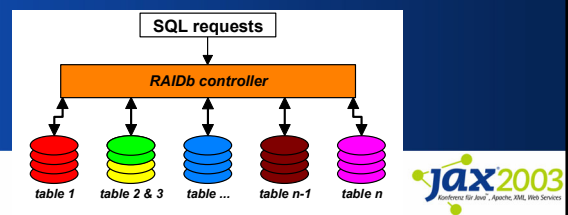
RAIDb - Definition

- Redundant Array of Inexpensive Databases
- better performance and fault tolerance than a single database, at a low cost, by combining multiple database instances into an array of databases
- RAIDb controller
 - gives the view of a single database to the client
 - balance the load on the database backends



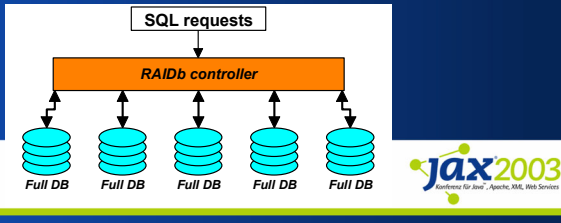
RAIDb levels

- RAIDb-0
 - partitioning
 - no duplication and no fault tolerance
 - at least 2 nodes



RAIDb levels

- RAIDb-1
 - mirroring
 - performance bounded by write broadcast
 - at least 2 nodes



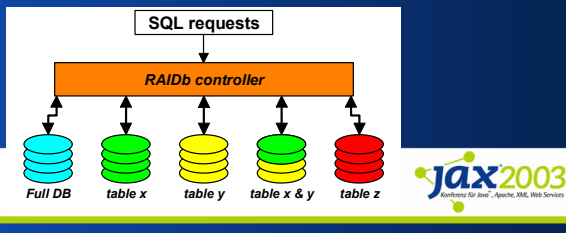
RAIDb levels

- RAIDb-1ec
 - mirroring + error checking
 - error checking
 - read request sent to multiple databases
 - replies compared
 - result returned only if a quorum is reached
 - at least 3 nodes



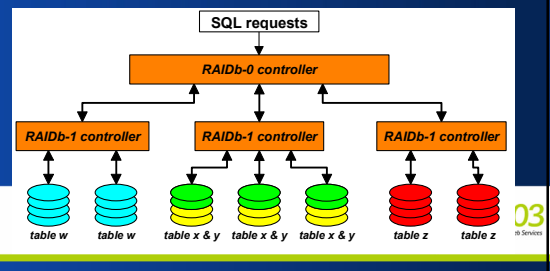
RAIDb levels

- RAIDb-2
 - partial mirroring + partial partitioning
 - at least 2 copies of each table
 - at least 3 nodes



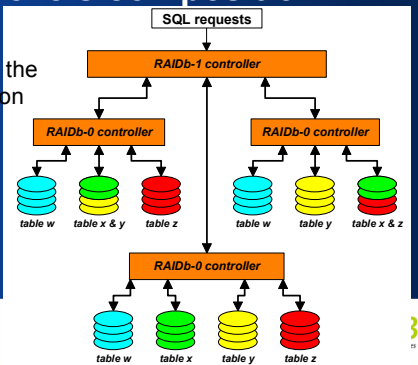
RAIDb levels composition

- RAIDb-0-1
 - RAIDb-0 at the top level
 - RAIDb-1 underneath



RAIDb levels composition

- RAIDb-1-0
 - no limit to the composition depth



Horizontal scalability

