OSS and transaction processing

Dr Mark Little
Engineering Director, Red Hat

OSS TPM vendor

- **Implementing a TPM is not trivial**
- **There are cooler things to do**
  - High speed messaging, clustering …
- **Failures? What failures?**
  - If you assume failures never happen, it becomes easier!
    - Ignore durability and recoverability
    - Participants always do as they are told
- **Most applications only have a single participant anyway**

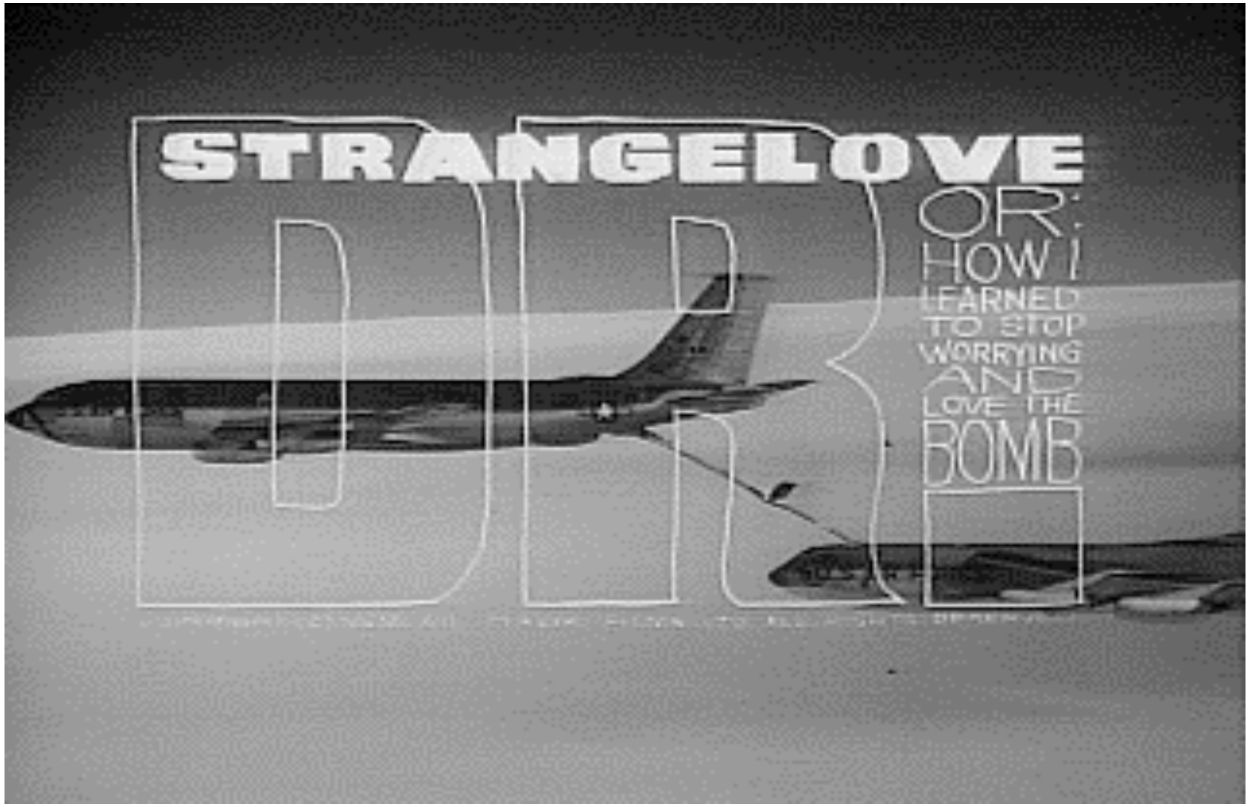
OSS RM vendor

- **Most applications only have a single database or RM**
- **Failures never happen**
  - “Not even close to the 80/20 case”
- **Performance is important for customers**
  - Reduce disk access
- **There are cooler things to do!**
  - E.g., clustering

OSS customer

- **Transactions? What are those?**
- **Failures rarely happen**
  - Always a hard sell for transactions
- **“Oooo! This stuff is cheaper than IBM :-)”**
  - “Hey, what’s the worst that can happen?”

Sometimes ignorance is bliss!



OSS was immature, but now …

- **Many customers regularly run with multiple one-phase only participants in the same transaction**
  - – And still expect atomicity and recoverability!
- **Lack of education on the problem**
  - – Often shocked when risks are explained
  - – Sometimes want atomicity and recoverability with no impact on performance
  - – Often want to keep using same multiple one-phase participants
    - “The customer is always right”.

OSS maturing

- **TPMs with support for recovery**
  - JBossTS (ex HP-TS) now in OSS
- **RMs support for 2PC**
  - MySQL
  - Derby (aka CloudScape)
  - Various JMS implementations

Compensating transactions?

- **Sometimes customers can’t migrate quickly to more mature environments**
  - Problem will exist for many years to come
- **Forward compensation transactions may offer a medium-term solution**