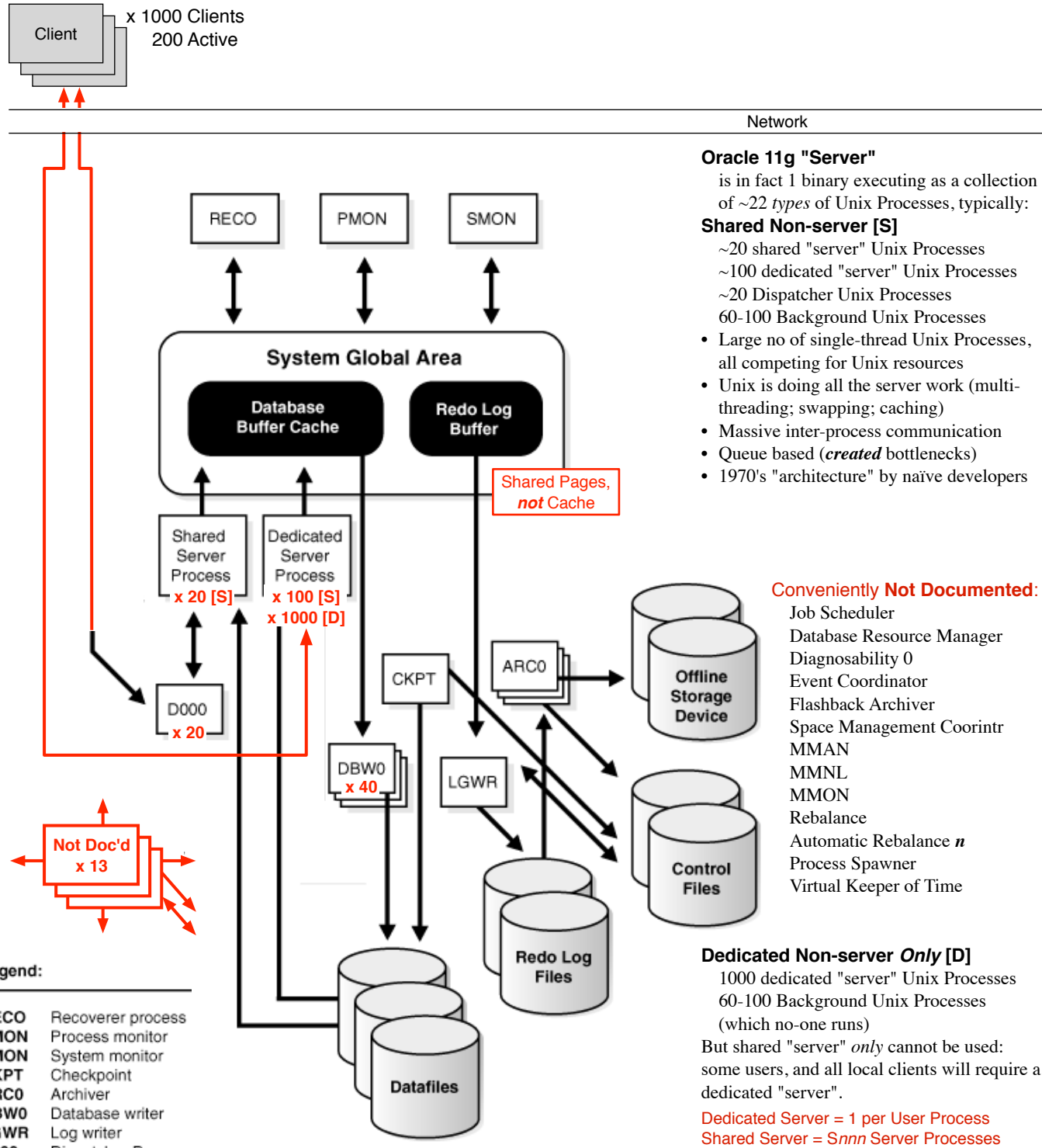
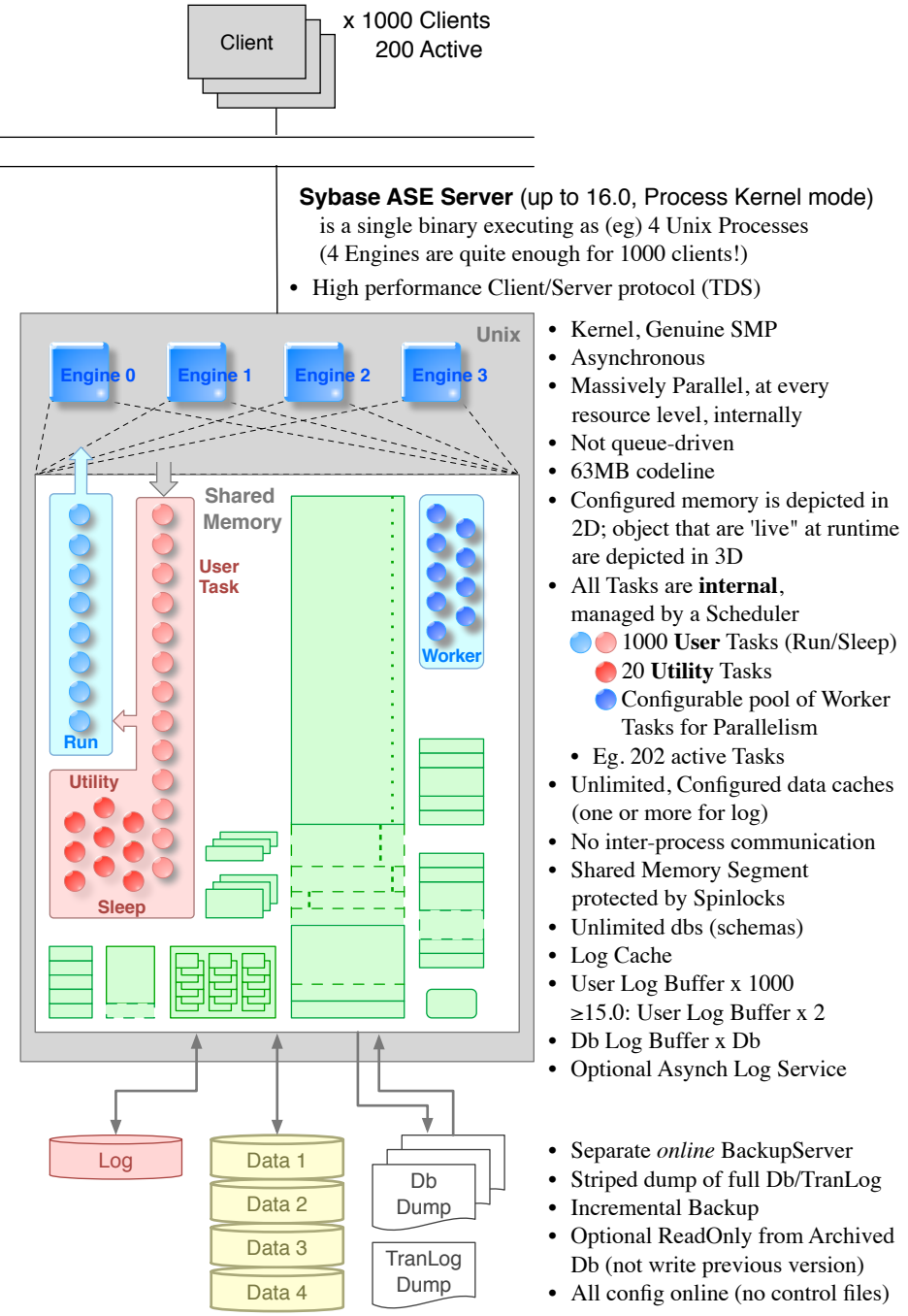


Process Non-architecture vs Architecture • Internal

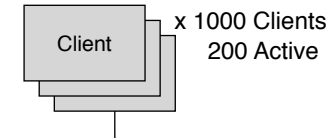
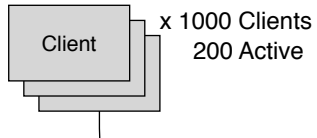


This page uses the published Oracle 11g diagram as reference; for each Oracle component shown (the level of detail in the Oracle diagram), the Sybase equivalent is shown; the features that Sybase has, that are absent in Oracle, are not shown.



More detail at [Sybase ASE Architecture](#)

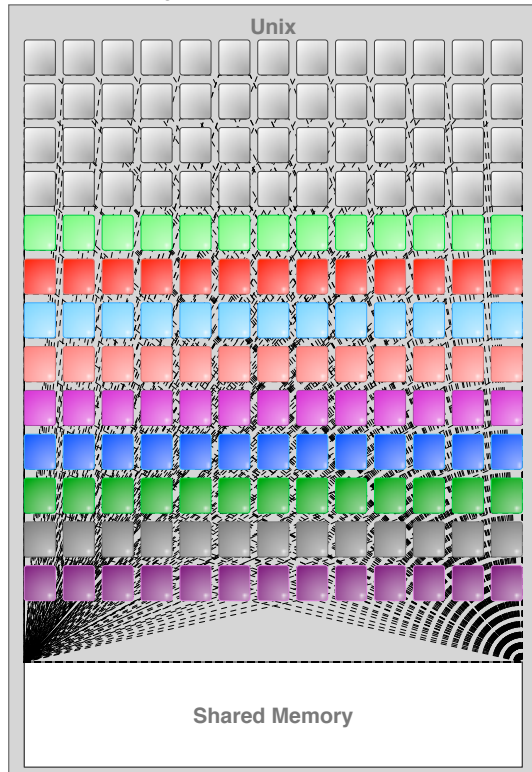
Process Non-architecture vs Architecture • Unix Level



Network

Oracle 11g Shared Non-server

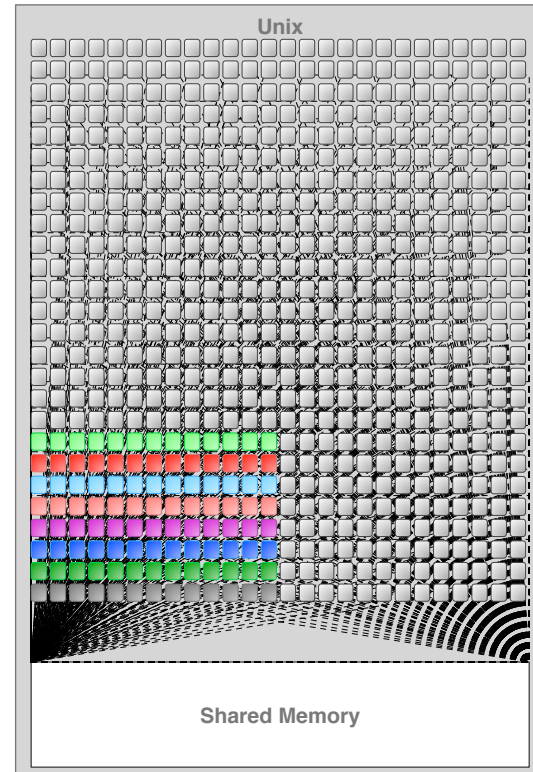
- ~50 shared "server" Unix Processes (light grey)
- ~20 Dispatcher Unix Processes
- ~50 dedicated "server" Unix Processes
- 60-100 Background Unix Processes



- Large no of single-thread Unix Processes, competing for resources
- All competing for access to Shared Memory
- Unix is doing all the server work (multi-threading; swapping; caching)
- The most expensive operation in Unix is a Context Switch, and there are billions here
- Massive inter-process communication
- 1970's style prototype, by naïve developers
- It is in fact a non-architecture; a non-server, it cannot be described as "architecture" or a "server"
- Good for small-scale databases and loads

Oracle 11g Dedicated Non-server

- 1000 dedicated "user" "server" Unix Processes (light grey)
- 60-100 Background Unix Processes

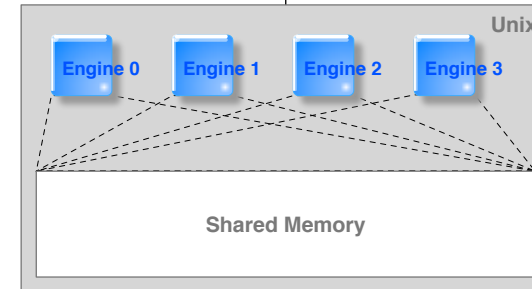


- Not the difference in scale: Client/Server "connectivity" is a joke
- Typical of freeware: Oracle, PostgreNonSQL and MyNonSQL have the same absence of architecture
- Thousands of Unix Processes, competing
- Massive inter-process communication
- Queue based (created bottlenecks: if the absence of architecture is not enough to drive performance through the floor, a queue will do that for sure)
- Selecting a machine configuration that handles thousands of processes (trillions of Context Switches) is best
- High throughput is not possible, and therefore not necessary (save the expense of a high-throughput machine).

On the previous page, the published Oracle 11g diagram obfuscates the technical facts, thus it is not readily comparable to the technical diagram provided for Sybase. A technical diagram of Oracle 11g is provided here, in order to afford that comparison.

Sybase ASE Server (up to 16.0, Process Kernel mode)

is a single binary executing as (eg) 4 Unix Processes (4 Engines are quite enough for 1000 clients!)
The server is "dedicated", there is no other option.



- Kernel, Genuine SMP
- Asynchronous
- Massively Parallel
- Not queue-driven
- 63MB codeline
- Shared Memory Segment protected by Spinlocks
- Unlimited databases

- Architected & designed as a **Database Machine**
- **Sybase Adaptive Server Enterprise** is highly configurable (beyond the non-server sense of "configuration")
- Highest performance is gained from:
 - selecting a machine configuration for databases and high throughput
 - using a dedicated machine for ASE, and
 - configuring ASE to take full advantage of the resources on the host system
 - configuring ASE for the specific use and load

More detail at [Sybase ASE Architecture](#)

Eliminating the Waste

- For the arithmetically challenged. For the given example of 1000 connected User Processes, an architected replacement for the Dedicated non-server would be **1 Unix Process**.
- It eliminates: 1000 User Processes; 60-100 Background Processes; Inter-process communication; and Inter-process contention management.

Oracle Circus

Process "Architecture" is of course a blatant lie. A list of Unix processes clamouring for resources is not an architecture, same as a list of buildings in an university is not an architecture. Architecture means Integration, planning and design: a single coherent unit, the components of which strive to a common principle, and nothing but. The collection of 22 binaries is evidence of the *absence* of architecture. The term "server" is false. The only statement in which Oracle can be associated with "server" is this: Oracle is a collection of 22 disparate, competing Unix processes that together, attempt to perform the functions that are normally performed by a single server process.

For further information, visit Brian Ceccarelli's [Oracle Facts 1](#) & [Oracle Facts 2](#). (His website is defunct, links to saved PDFs.)

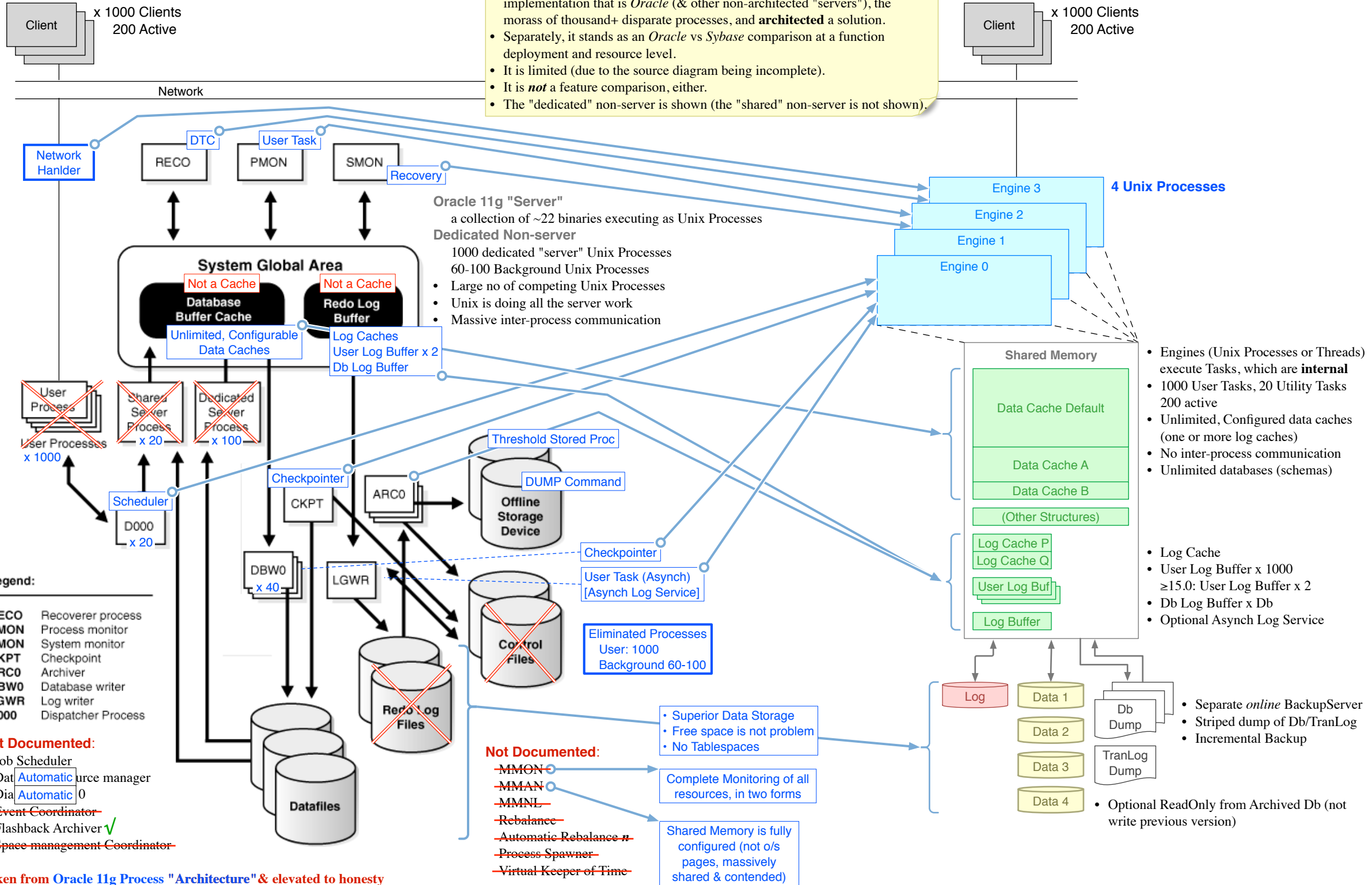
Note that the precision and order above is for diagramming convenience only; no precision or order in Oracle is suggested.

Oracle vs Sybase Architecting the Processes



Taming the Circus

- What a competent engineer would do, if he inspected the naïve implementation that is *Oracle* (& other non-architected "servers"), the morass of thousand+ disparate processes, and **architected** a solution.
- Separately, it stands as an *Oracle vs Sybase* comparison at a function deployment and resource level.
- It is limited (due to the source diagram being incomplete).
- It is **not** a feature comparison, either.
- The "dedicated" non-server is shown (the "shared" non-server is not shown).



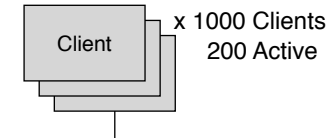
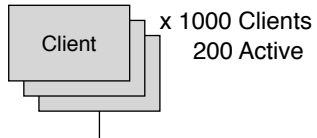
Taken from Oracle 11g Process "Architecture" & elevated to honesty

Derek Ignatius Asirvadem • 20 Nov 14

Copyright © 2014 Software Gems Pty Ltd

Oracle vs Sybase • 3 of 4

Threaded Process Non-architecture vs Threaded Architecture



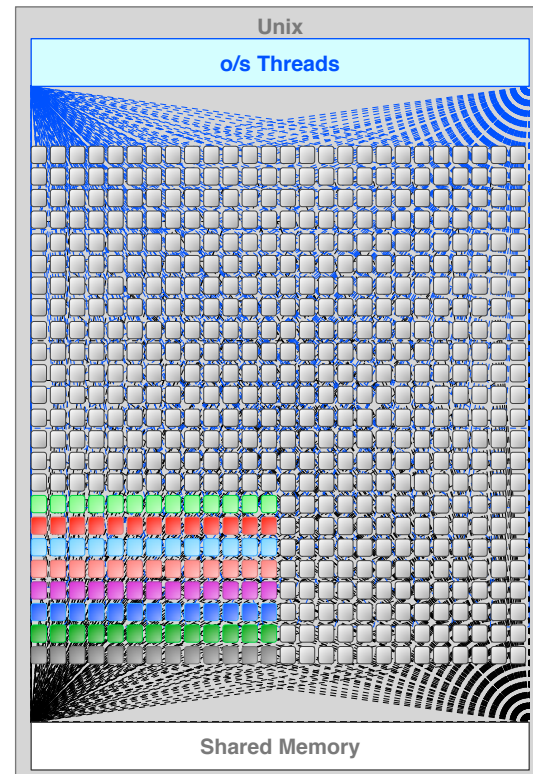
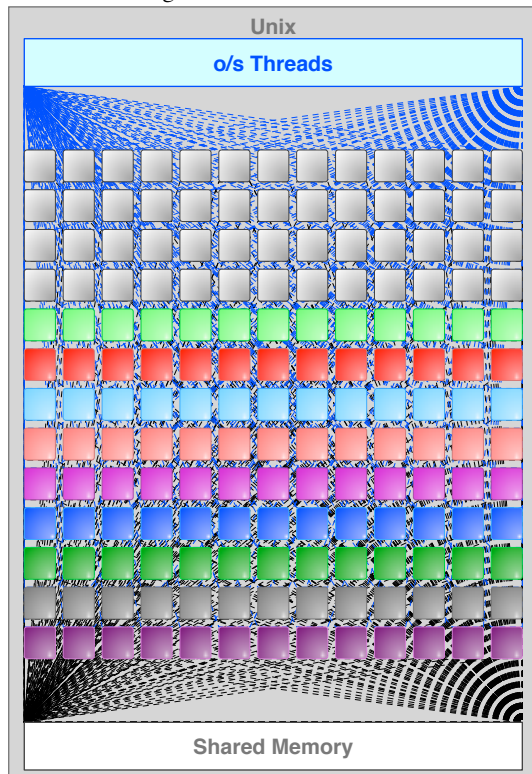
Network

Oracle 12 Shared Non-server

- ~50 shared "server" Unix Processes
- ~20 Dispatcher Unix Processes (light grey)
- ~50 dedicated "server" Unix Processes
- 60-100 Background Unix Processes

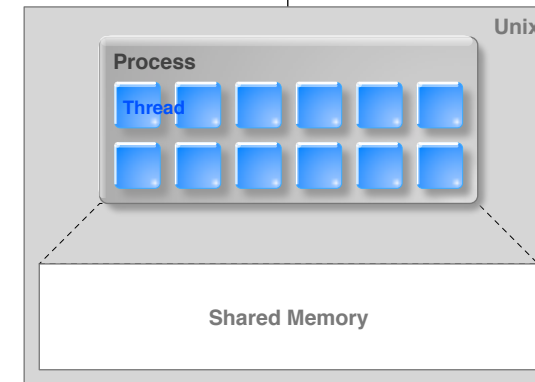
Oracle 11g Dedicated Non-server

- 1000 dedicated "user" "server" Unix Processes (light grey)
- 60-100 Background Unix Processes



Sybase ASE Server (up to 16.0, Threaded Kernel mode)

is a single binary executing as (eg) a single Unix Process with 12 Threads (quite enough for 1000 clients!). A natural progression from the Process Kernel architecture.



- Just as as fewest Engines in Process Kernel mode perform the best, the fewest Threads in Threaded Kernel mode perform the best; high *internal* CPU Usage is desirable
- Threads are configured and pooled, in order organise resources, and to avoid saturating the machine.
- Limitations resulting from Process binding eliminated
- Various efficiencies gained
- Disk, Network & Process Latency eliminated
- Need for Spinlocks reduced

- Architected & designed as a **Database Machine**
- *Sybase Adaptive Server Enterprise* is highly configurable (beyond the non-server sense of "configuration")
- Highest performance is gained from:
 - selecting a machine configuration for databases and high throughput
 - using a dedicated machine for ASE, and
 - configuring ASE to take full advantage of the resources on the host system
 - configuring ASE for the specific use and load

More detail at [Sybase ASE Architecture](#)

Eliminating the Waste

- For the arithmetically challenged. For the given example of 1000 connected User Processes, an architected replacement for the Dedicated non-server would be **1 Unix Process with 12 Threads.**
- It eliminates: 1000 User Processes; 60-100 Background Processes; Inter-process communication; and Inter-process contention management ... all fighting over (eg) at least 128 Threads.

- O/s Threads are executed by *all* Oracle Unix Processes (except "dedicated" User Processes). The 11g circus has progressed to a rave party. This is a simple extension of the diagram on the previous page: the technical diagram required would highlight the progression of insanity, but it requires an A3 page.
- No Thread pooling or limiting is provided, Thread saturation is "normal", an additional, new point of contention, a "feature".
- As always, in order to misrepresent the product, Oracle has private definitions of technical terms. The latest is their use of "multi-threaded" which alludes to Threads. The misrepresentation is to hide the fact that Oracle has never had, and this version does not have, Multi-Threading. Any Multi-Threaded operation that the thousands of competing Unix Processes (or the competing o/s Threads) do obtain, is provided outside Oracle, by Unix.
- Their introduced confusion re "Unix process" vs "Oracle process", to obfuscate "function" is noted, but not expanded. Another simple lie to keep the natives stupid and confused.
- Selecting a machine configuration that handles thousands of processes (trillions of Context Switches) is best
- The more Threads (or Cores, etc), the merrier: there will never be "enough".

For further information, visit Brian Ceccarelli's [Oracle Facts 1](#) & [Oracle Facts 2](#). (His website is defunct, links to saved PDFs.)

Note that the precision and order above is for diagramming convenience only; no precision or order in Oracle is suggested.