Sybase ASE Kernel • Common Misunderstandings

There is an awful lot of misinformation on the web, and these days, anyone with two fingers and a keyboard can create a blog site. On LinkedIn groups, such misinformation is propagated. Correction or refutation of such misinformation gets deleted on some groups, and people like me stop refuting same, therefore people remain in a state of misinformation. I am hoping that Pankaj's new group allows such misinformation to be deleted or refuted. Here is my first contribution.

There are many blog posts on LinkedIn about all sorts of weird fears re the ASE Kernel (all versions), and especially the new 15.7 Threaded Kernel. Rather that answer each point of misinformation or misrepresentation, allow me to post only the correct and relevant info.

1 Server Architecture

There seems to be a growing number of DBAs who administer ASE, who do not have a reasonable understanding of ASE Architecture (unfortunately some of them post blogs!). This is not an overview of that rather large problem, but I will address one common misunderstanding. This applies to all versions, all kernels.

Some DBAs have the view that many engines is better than fewer engines. ***False***.

You may have decided, when you purchased the machine, that you need 8 CPUs with 4 Threads (or Cores) each. *Marvellous*.

That applies to orrible platforms that do not have a server architecture, such as orable. You may be excused for thinking that orable is a server, because the word "server" is misused and abused and stamped 42 times on every page in the manual, but none of those pieces of paper will make a non-server a server, only a genuine server architecture will.

It simply does not apply to ASE, which has a beautiful and matured server architecture. That means you chose an incorrect configuration for the machine for ASE. You wasted an awful lot of money, and you will have machine resources that are wasted, but that in itself will not prevent you from running ASE efficiently, and getting massive throughput out of whatever box you have.

That is one mistake, bad enough, do not propagate that mistake to another area, where it does not yet damage that area; if you do, you will damage that area.

You have 32 threads, so you build an ASE Server, and configure 32 engines. ***Dead wrong***. You just propagated your mistake into a second area, without understanding that second area.

You look at the sysmon reports, and you see that 32 engines are running at 10% CPU usage each, and you think that is wonderful, you are doing well, and the machine is doing well, and the ASE Sever is doing well. ***Dead wrong***. On all three counts.

For orable:

- Dedicated Non-Server Mode: for 100 clients you may have 210 unix processes
- Shared Non-Server Mode: for 100 clients you may have 130 unix processes

(either way, there is at least one process for each client connection) because it gets the o/s to do all the normal work of a genuine server, such as context-switching, multi-threading, thread management, memory page caching (and swapping), etc. That is, (a) Unix does all the work that orable should be doing, and (b) there is a mass of Unix processes that unix has to manage.

If you examined an orable non-server with 130 or 210 processes (on orable you have to look at Unix metrics, there are no internal metrics ala sysmon, because there is no server, and because unix is doing all the work), and all of them were accruing 1% or 3% CPU usage, yes, that is a Good Thing. If you examined just the 30 orrible background processes, and they were accruing 5% each, yes, that would be a Good Thing.

But this is Sybase ASE land. A small platoon of commandos, not a battalion that cannot shoot straight.

ASE is architected as a true server. One single parent process with the first engine, and one process per engine thereafter. They do not use Unix for anything. You cannot understand what ASE is doing, or tune ASE by looking at the Unix metrics, you have to look at sysmon metrics, all 250 of which are internal to ASE. Take a look at a formatted and normalised report of 24 hours of sysmons. Orrible DBAs have only vmstat, the last 18 lines in that report.

The client connections; their workspaces (stacks); the several data caches; all of it, is internal to ASE. Each connection is an internal Thread, not a Unix process.

32 engines ***inside one server*** accruing 10% CPU usage each means 32 engines are 90% idle. ***Dead wrong***.

You can double the overall throughput and halve the response time by configuring 16 engines at 20% CPU each. Actually, it will go to 25%.

You can double the overall throughput and halve the response time ***again*** by configuring 8 engines at 50% CPU each. It will probably go to 55% each.

When you reach the highest throughput and lowest response time, you may end up with 4 or 6 engines, running at 80%. That is a Good Thing in Sybase land. That is *not* propagating your first mistake into a second area.

Sure, there are 24 idle threads on your machine, but there is nothing you can do about it (ok, you can transfer the box to orable use, and buy the correctly configured box for ASE).

- If you have accomplished a high CPU usage, do not build a second ASE Server on that box, you will be competing with yourself. ASE is architected to take full advantage of the machine, which means the two ASEs will compete worse than two orable processes, or two massive CPU hogs. Just one server per machine, please and thank you.
- For non-production systems, where the CPU usage is low, feel free to build a DEV or DR server on the UAT box, etc.
- But even then you are better off with the ASEs on separate LPARs, or Containers.
  - If you have VMWare or such, ***do not hide*** the 10% overhead, the machine resources that you lose (*insane for a machine intended as a database server, but hey, everybody's doing it, doing it, doing it*).

Being a true server architecture, Sybase ASE will run the best (highest overall throughput; lowest response time per connection), with the fewest engines that can do the job. And yes, that means a high CPU usage each (internally). And examine the vmstat by all means as well.

Do not believe me, take an hours sysmon; correct the configuration on your server; take another hours sysmons; and compare the two sysmons. This confuses the hell out of orable DBAs, because they have never experienced a true server (and they have so many years of being bombarded with "server"). So the falsehood that needs to be corrected, in order to obtain an accurate technical understanding, so that you can administer both orrible non-servers-branded-as-servers and administer Sybase ASE, effectively, is that that oribble golum-like creature is not a server by any stretch of the imagination, you have been lied to for decades. *Not* that ASE is weird or difficult to tune.

It seems no one reads manuals these days, but the Sybase manual clearly states what I have stated above. Sure, for serious configuration or serious tuning, you are better off getting someone like me, but for the basics, genuine qualifications and the manual is enough. And this is "Basics".

2 Configuration Errors

Before we go into the next item, we need to address a couple of caveats.

1. First, completely totally address [1]. If you don't you will be approaching the Threaded Kernel (and please, get this) not with an incorrect Process Kernel mindset (which is incorrect configuration, bad enough), but with an orrible, ignorant mindset (which is grossly incorrect).
2. Second, if you have any problems, any bottlenecks, any configuration errors, on your ASE Server, you must correct them first. If you migrate to 15.7 without fixing them:
    a. those problems will be carried with you,
    b. and they will be worse.

If you are new to the DBA role, and especially if you do not understand [1], which means you will have fear and uncertainty about [3], those problems (that have just gotten worse on 15.7) will appear to be 15.7 problems. Ooh, ooh, back out the upgrade, go back to 15.5, the tsunami was only a monsoon then. Save yourself the upgrade plus downgrade effort: correct the problems first.

2.1 I/O Bottleneck

There can be many bottlenecks, I will address just one, as an example. A properly configured ASE (any version) will be high CPU Usage ***and*** I/O bound. That means we have fixed up all the configuration issues, such that the ***only*** thing limiting the speed of ASE is the only moving part in the machine: the disks. I won't get into the incorrect disk configurations; RAID configurations; and idiotic SAN configurations, but yes, those have to be configured correctly first. So assuming that they are, there is no reason that ASE should be doing anything else (in the idle time) except waiting for I/O. The configuration capability within ASE re I/O handling is (a) manyand varied, and (b) mature. So there is no reason that ASE should be handling I/O at anything less than the highest speed that you can squeeze of of the disks (whatever configuration they may be).

> *An incorrect 15.5 configuration is a 150kg runner running in the 500m race. Get him fixed up, so that he can compete with the others. Do not change the race, adding runners will only have the effect of crowding the race. Orable is a 150kg runner running in the 40km race: it doesn't matter, they all come in at about the same time,* **hours** *from the start.*
>
> *If you carry an incorrect configuration over to 15.7, it is a 150kg runner running in the 100m race. The others have left the stadium, and the poor guy is still running. Get him fixed up,* and *get him placed in the 500m race, before attempting a 100m race.*

Make sure your server is (at least) configured to handle the maximum I/O throughput that the machine is capable of. That means, monitor and tune the following:

- `disk i/o structures`

  Set it to whatever you have determined to be the maximum number of outstanding I/Os, along the lines of:

  `max online engines x number of devices x 16`

  and monitor it. Do not waste memory allocating structures that are not used.

The next tranche of I/O configuration is the Devices, and Object Placement, which results in I/O Load Distribution. Since that is a subject in itself, and very few DBAs understand it (eg. most believe in the myth of the SAN, despite the mountain of proof otherwise), let alone configure it, I will not get into it here. The above is the minimum, the basics.

3 ASE 15.7 Architecture

ASE 15.7 is Threaded Kernel, period. Process Kernel is not an option, it is only there for backward compatibility. I will not be posting details about the non-option.

If you do not understand the architecture of ASE 15.5, you will not be able to understand the architecture of its advancement in ASE 15.7, so go back, and get that first.
If you need to understand Process Mode, look at the detail available about 15.5, not 15.7; the manuals do not (and should not) detail the non-option.

If you understand what I have described re [1] Server Architecture, fully, you will understand *why* configuration errors and problems get worse with a Threaded Kernel. Unfortunately, I cannot detail that here, but I will provide an overview that technically qualified readers may understand.

3.1 Threaded Kernel is Not New

ASE on Windoze has always been Threaded Mode. It is not new to Sybase Engineering, who wrote it. It is very mature. Sure, the platform teams are different, but they are not working with a new codeline, we do not have to worry about the bugs that come from a new codeline. But due to ASE being very tightly bound to the specific variant of Unix, in order to maximise its SMP architecture, yes, there are new bugs for each variant.

> *Ok, ok, so don't download GA, get ESD#3 at least. SP100 is even better.*

For Unix deployments, it is just one increment of advancement over the 15.5 Kernel. And that one incrment is a natural (not foreign) progression of the 15.5 architecture (which is why I say, understand that first).

It has been in production at many (hundreds) of SAP sites for over two years. ***Deal with it.*** Sybase has been presenting it technically at SUGs for over two years. Bloggers who have just found it are, as per evidence, more than two years behind the times.

3.2 Less is More

(Except on orable of course, where largesse is more.)

Kernel Mode accomplishes three things:

1. First, it drastically reduces context-switching, which, on a properly tuned server was "the wall" that we hit. We reached for it; if we were good, we got to it quickly; but we could not overcome it: we could tune it up to the wall, but that was the limit.

   Sybase Engineering have overcome that wall, congratulations!

   That means 8 engines at 80% CPU usage now run at 60%. In ASE-15.5 language, That means you should change the number of engines to 4. Those among you with three digits in the IQ department will see where it is going.

2. Second, to accomplish that, to eliminate context-switching, it has to eliminate the number of processes (context switching means switching to another thread, another engine, another Unix process), eliminate the one process per engine architecture. So 15.7 is One Unix Process. Threads are now fully integrated hardware Threads. Threads are managed internally. One engine per thread. All configurable. That is the specific overview.

   If you really understand why 8 ASE engines (not orrible processes) running at 80% CPU usage will produce more throughput than 32 engines running at 10%, it will be just a small step for you to understand why 4 engine Threads at 90% in a single process, will produce even more throughput.

3. Most of the context switching was due to I/O issues: threads having to sleep; then having to wake up when the asynch I/O or network I/O completed; etc. That, too, is eliminated. We have a couple of special threads that are configured for I/O only. It cannot eliminate the I/O Wait Times (that is the disk configuration or network latency), but the I/O Overhead has been completely eliminated. And because of the elimination of context switching, we need very few of those I/O only threads.

   Engineering will find more to eliminate in the next major release, but this, the current identified bottleneck, is gone.

Threaded Summary. ASE 15.7 executes as one Unix Process, which is the next increment, or the highest extent, of the current ASE architecture. Processes are now Threads. Even more integrated with h/w Threads. ASE 15.5 needs very few Processes; ASE 15.7 needs still fewer Threads.

3.3 Throughput & Speed

ASE 15.7 is ***much*** faster than 15.5. In my configurations, most code runs 2 ***times*** faster, some 5 ***times*** faster, and some at 12 ***times*** faster. Throughput is massively increased, but I do not want to post figures, because my configurations are not bedded down (I have limited windows and change control, because the servers are production).

Your mileage may vary is not accurate, because it covers only part of the reason. More important, note that:

- I have no configuration errors in my 15.5 ASEs, and
- I understand Server Architecture, therefore I can actually tune 15.7 to execute better than it does, out of the box.

But for new DBAs, out of the box, plus addressing the [2] caveats is definitely good enough, 15.7 will definitely be faster than 15.5. And not if your threads are accruing low CPU usage.

4 Caveat re Bloggers

There are some people who are very busy with their two fingers, and who don't have much grey matter in-between. They are prolific bloggers. If you are a blogger, you get to create your own view of the world on your blogsite, and you can delete any comments that you don't want, so you can enforce that world view on your bogsite. The world is a bit bigger than that, but if you stick to your blogsite and avoid the world, you run the very real risk of being isolated from reality. Same as anyone in an asylum, really.

If you google this subject you may find weird info; self-contradicting info; wild fantasies; fear and loathing; tests that do not apply; concerns that do not apply. Asylum dwellers are very afraid of anything they do not understand. Unfortunately, you will not find test that *do* apply.

Again, I can't address all the nonsensical posts, but I will address one relevant example.

One blogger has posted two "benchmarks" of ASE 15.5 vs 15.7/Threaded.

- One "test" using 10 engines and a second using 20 engines (on both sides). If you understand this article, you will realise that that is ***dead wrong*** for 15.5, and ***frozen solid*** for 15.7. It compares 10 and 20 x 150kg runners in the 500m race with 10 and 20 x 150kg runners in the 100m race. The results are relevant only to people who have never seen a race, who are only concerned for their fat children; who have never experienced a correctly configured ASE server. They are comparing a *bad* 15.5 configuration with a *terrible* 15.7 configuration. And the results ? Who cares.
- Instead of preparing a statement once, and executing it a million times (each), he prepares the statement a million times and executes it once (each). So the test may be relevant to demented developer who lives in a cave, but it is by no means a general case or relevant to the Sybase community.
- The vertical scales on the charts do not match. Either he has never written a comparative report, or he is purposely misrepresenting the results to us.
- All the charts are back to front, just like the grey matter. None of them are labelled, so the 15.5 may well be mixed up with the 15.7 stats, we will never know. They are all the same anyway, because the performance of 150kg runners on any test is, well, the same.
- He has weird expectations, such as, Large I/O stats in a PageSize Pool that has a Large-I/O Pool configured, and it is somehow a Sybase bug that they are zero. Move the horse to back paddock, and then complain that it is missing from the front paddock. But he is waiting for "bug" to be fixed, before he accepts 15.7. Two years and still waiting.
- After months of posting fear and loathing about architecture that he evidently (that means, by virtue of the evidence that he himself has posted) does not understand, now he contradicts himself completely and discusses the nice smell of the cheetah. Must have taken his first DBA course. But no retraction. ASE 15.7 has not changed. The only thing that might have changed lies in-between those two fingers. He has forgotten that he posted the opposite, and that both his opinions cannot be true. More important, both opinions have the same credibility, zero.
- No one else has those problems. Refer Sybase figures. Refer my figures. But in the isolated enforced "reality" of the cave, deleting all comments, sure, that is "real". The 150kg runners perform the same.

Smells, yes, but not the cheetah.

Info that is unreliable or confused or contradictory should be trashed, do not waste your time. Remember orable, the heavy marketing is always the opposite of the truth; the truth does not need marketing. Self-promotion is always a sign of mental illness.

I hope Pankaj does not allow that here. Otherwise this group will deteriorate quickly into a marketing one, and lose its relevance to genuine technical people.

5 Summary Overall

The reason Server Architecture is not commonly understood and the very real obstacle to having it be understood is, large-market-share non-servers have been bombarding the faithful for decades, marketing their non-architecture as a "server". Correct that, and genuine understanding of Server Architecture will naturally flow from it.

Do not follow bloggers who contradict themselves and post masses of misinformation. Beware of self-promoters. Learn to discern misinformation from information.

If you are experiencing problems with ASE 15.7, or you have reservations about the architecture, you need to step back and cover the specific areas mentioned above. Understand 15.5 architecture first, then proceed to 15.7.

Downgrading is only due to upgrading without proper administrative oversight, or ignorance, both of which are easy to correct.

5.1 Questions

Please make your questions specific, and real: I will not waste time answering hearsay or opinion, because it will merely prove that you do not understand some technical aspect, and it will turn into an education session (time-consuming for me, and embarrassing for you). If in doubt, you are better off asking a question, than making a statement. I always respond to genuine problems (as opposed to what you think will happen if you had a server) and specific technical questions, as and when I have time.

6 Further Reading

1. First, understand Server Architecture in general, and ASE 15.5 Architecture in particular. If you are an orrible bunny, take a red pen and strike out the word "server" in those manuals: it will seriously help you to understand, and therefore administer, that freeware.
2. Read the New Features Guides, every one of them, from the version that you do know, upwards, incrementally. Some of them have great info.
3. The manuals cannot be expected to deliver education, only product specific info, but you must cover those. The 15.7 manuals do provide 15.7 info. Yeah, there are missing bits, and yeah, they do not match up with the NFGs, but that's life. They are still streets ahead of the orrible manuals, mostly because *they do not lie*.
4. Peter Thawley has a couple of presentations available in PDF. Of course, reading the PDFs only means that you miss the presentation, the explanation. He is famous for providing masses of low level info, and not enough top-down info, but he is the only Sybase person providing correct info, so the **PDFs are essential reading**. His style may have faults but we have known him and loved him for decades. I will save you the annoyance of chasing down links on the SCN blogsite that are broken, and have been broken for months, here are direct links.

- The first covers 15.7 completely (my post covers the Kernel only; go to page 28 for that)
  A Technical Look Inside ASE 15.7
- The second covers the Threaded Kernel only, in more detail.
  A Technical Look Inside ASE 15.7'S "Hybrid-Threaded" Kernel

Regards
Derek Ignatius Asirvadem