# The Heaviest Studies on the Smallest Box

**SynthIQ capacity under 3D tomosynthesis and high-instance-count CT — measured.**

Can a vendor-neutral DICOM load balancer survive a flood of 3D tomosynthesis and 10,000-slice CT studies arriving at once? We put SynthIQ on a dedicated performance rig — on the smallest instance a cloud will rent, **2 vCPU / 8 GB** — and measured it. The result: **70 tomosynthesis studies and 30,000 CT objects moved at 100% delivery using under 700 MB of memory.** SynthIQ does not decode or hold pixel data; it spools to disk and streams. Memory is never the wall.

*Non-device software under FD&C Act §520(o)(1)(D). Every number below is a measured result, not a projection.*

---

## Executive summary

The fear with any DICOM proxy is the big study: when a mammography unit pushes a 1 GB tomosynthesis volume, or an "Alpha CT" dumps a 10,000-slice study, does the load balancer fall over? We answered it the only credible way — on a dedicated rig, on the smallest instance a cloud will rent (2 vCPU / 8 GB), with the system-under-test isolated and instrumented.

- **100% delivery, every run.** 70 tomosynthesis studies (35 GB) and a mixed 30,000-object CT workload (30 GB) — zero failed objects.
- **Memory peaked under 700 MB** — about 8% of an 8 GB box — and **plateaued** (no leak) across 23-minute sustained runs.
- **SynthIQ's memory is bounded by concurrency, not data volume.** It never decodes or buffers pixels; each in-flight object spools to disk and streams to the backend, so the heap stays flat whether you send a handful of gigabyte objects or thirty thousand small ones.
- **It stayed responsive throughout** — the management API answered every health probe during every run; no stalls under load.
- **The only thing that scales with the box is throughput, not memory.** Extreme-instance studies move faster with more cores; memory is never the limit.

**The practical headline: SynthIQ handles every modality on a 2 vCPU / 8 GB instance with memory to spare.** You size the box for the throughput you need, not to survive the biggest study.

## §1 — The real question, and the two fears behind it

"Can it handle big studies?" hides two distinct fears that stress different resources:

**Fear 1 — the giant object (memory).** A single 3D tomosynthesis volume is ~561 MB to over 1 GB. The worry is that a few arriving at once exhaust RAM. This is the failure mode that constrains *processing* nodes, which must hold the decoded volume in memory.

**Fear 2 — the giant count (throughput & contention).** A wide-detector CT angio can be 3,000–10,000 thin slices — each small, but the number stresses the per-object pipeline: connection churn and the routing-state lookups a study-aware balancer performs for every instance. This is the contention class that can choke a naïve database-backed router.

A trustworthy answer addresses *both* axes. This paper measures each.

## §2 — Methodology

- **Dedicated rig, isolated roles.** Three machines on a private network: one generated DICOM load ("the modality"), one ran SynthIQ under test, one was the storage backend. Nothing else competed for resources.
- **The smallest instance on purpose.** The system-under-test was a **2 vCPU / 8 GB** box — so any ceiling would show first. (You cannot prove "no memory wall" on a 64 GB host.)
- **Two backends.** Runs were repeated against a fast no-op sink (to isolate SynthIQ's own behaviour) *and* a real production router (to confirm under realistic backend drain). The results matched.
- **Both stress axes.** A tomosynthesis profile (few huge multi-frame objects) and a high-count CT profile (a fleet of 1,000-object studies plus two 10,000-object "Alpha CT" studies), fired concurrently.
- **SUT-side instrumentation.** Every run sampled SynthIQ's own process memory, system free memory, CPU, temp-disk spool, and management-API responsiveness on a fixed interval — so the numbers are SynthIQ's, not inferred from the client.

## §3 — The central finding: memory doesn't track data

The single most important result: **SynthIQ's memory footprint is governed by how many objects are in flight at once, not by how big they are or how many you send.** Because SynthIQ is a routing proxy — it inspects an object's identity to make a placement decision but never decodes the pixels — each in-flight object is spooled to a temporary file and streamed to the chosen backend. The pixel bytes live on disk for the moments they're in transit; they never inflate the heap.

Peak heap stays at the floor of the box across a 7× range of data volume:

| Data pushed through SynthIQ (per run) | SynthIQ peak heap | Host RAM |
|---|---|---|
| 5 GB | 260 MB | 8,192 MB |
| 35 GB tomosynthesis | 490 MB | 8,192 MB |
| 30 GB CT (30,000 objects) | 620 MB | 8,192 MB |

Memory does not track data volume. The in-flight pixel data is on disk, not in the managed heap.

## §4 — Fear 1, tested: 3D tomosynthesis (the giant object)

Batches of 4 multi-frame tomosynthesis objects per study (~512 MB/study) at 10 concurrent studies, then sustained, then repeated to check for leaks — through a **real production router** backend, on the 2 vCPU / 8 GB SynthIQ:

| Run | Studies / payload | Delivery | SynthIQ peak heap |
|---|---|---|---|
| 10 concurrent | 10 / 5 GB | 100% | 274 MB |
| Sustained churn | 50 / 25 GB | 100% | 478 MB (plateaued) |
| Leak check (repeat) | 10 / 5 GB | 100% | 490 MB (held) |
| **Cumulative (one process)** | **70 / 35 GB** | **100%** | **490 MB** |

Memory rose to steady-state during the sustained run and then **held flat** — it did not ratchet up across the repeat, the signature of no leak. Throughout, ~7 GB of RAM stayed free and CPU was never pinned. The disk spool absorbed the in-flight volume (sawtoothing up to ~19 GB and draining in waves) while the heap stayed under 500 MB.

## §5 — Fear 2, tested: high-count CT (the giant count)

To stress the *count* axis we simulated a busy CT environment: ten 1,000-object studies **plus two 10,000-object "Alpha CT" studies**, all arriving concurrently — 30,000 objects, ~30 GB, with a new routing decision and routing-state lookup for every object.

| Stream | Studies × objects | Payload | Delivery |
|---|---|---|---|
| CT fleet | 10 × 1,000 | 9.8 GB | 10,000 / 10,000 (100%) |
| Alpha CT | 2 × 10,000 | 19.5 GB | 20,000 / 20,000 (100%) |
| **Combined** | **12 / 30,000** | **~30 GB** | **100%, 0 failed** |

Over a 23-minute sustained run, SynthIQ's heap peaked at **620 MB and plateaued**; ~6.9 GB of RAM stayed free; CPU was never pinned; and the management API answered **every one of 282 health probes**. Critically, SynthIQ absorbed **30,000 routing-state lookups** — the exact contention pattern that overwhelms database-backed routers — without a stumble, because its study-affinity store records a placement *once per study* and serves the rest from a fast cache.

Few-huge or many-small, the answer is the same: **heap under 700 MB, 100% delivery, on a 2 vCPU / 8 GB box.**

## §6 — "Why does SynthIQ use so little memory?"

Because it is a **router, not a renderer.** A SynthIQ placement decision reads an object's identity attributes — study, series, modality, SOP class — to choose which backend should receive it. It never decompresses pixel data, never assembles a volume in RAM, and never holds a study to "process" it. Each object is received to a temporary spool file and streamed straight out.

The consequence is the flat line in §3: the only memory in play is the per-object working set of however many objects are crossing the wire at one instant — bounded by concurrency, a few hundred megabytes — not the size of the studies or the total volume. This is the architectural difference between SynthIQ and a processing node (a PACS, a VNA, an AI pipeline), which must hold the decoded study and is sized for it.

## §7 — What *does* scale with the box: throughput

Memory is settled. The lever that remains is throughput. SynthIQ opens a fresh, fully-released association per forwarded object — a deliberate reliability choice that eliminates a class of stale-socket failures — so an extreme-instance study (a 10,000-slice Alpha CT) clears at a rate governed by per-object handling and available cores. On the 2 vCPU floor, that single 10,000-object study took about 23 minutes; on more cores it parallelises and clears proportionally faster. **Throughput scales with cores; it is never gated by memory.**

Capacity grows the same way it does for the rest of the platform — **by adding nodes, not bigger boxes.** A SynthIQ pool load-balances DICOM across multiple instances, and its weight-aware placement steers heavy studies toward backends with the most headroom. The result is linear, predictable scale-out with no forklift upgrade — and an architecture that is highly available by design rather than dependent on a single oversized node.

## §8 — Recommended configurations

Because memory is never the constraint, SynthIQ sizing is about throughput and concurrency, not survival of the biggest study. RAM is comfortable at 8 GB for any modality; cores set the pace.

| Site profile | Workload | Per instance |
|---|---|---|
| Entry / branch | any modality mix, modest concurrency | 2 vCPU / 8 GB / SSD |
| Standard | busy enterprise, mixed CT/MR/DBT | 4 vCPU / 8–16 GB |
| High-throughput | high instance-rate, many concurrent high-count studies | 8 vCPU / 16 GB |

- **2 vCPU / 8 GB handles every modality at 100% delivery** — the entry tier is not a compromise on correctness, only on throughput.
- **Add cores for throughput**, especially where extreme-instance studies must clear quickly.

- **8 GB RAM is ample** across the board; high-IOPS storage is unnecessary, though a fast temp volume helps the spool.
- Beyond a single instance's throughput band, **add a SynthIQ pool node.**

## Conclusion

The worry that a DICOM load balancer will buckle under big studies does not survive measurement. On the smallest instance a cloud will rent, SynthIQ moved 70 tomosynthesis studies and 30,000 CT objects at 100% delivery, using under 700 MB of memory, staying responsive throughout. Because it routes rather than renders, its memory is bounded by concurrency, not by the size or count of the studies — so **2 vCPU / 8 GB is enough for every modality, with room to spare.** Size SynthIQ for the throughput you want; you will never size it to survive the biggest study.

---

*Synthology Healthcare Solutions Group · SynthIQ Capacity Whitepaper · 2026-06-07. Measurements from a dedicated, isolated performance rig; the system-under-test was a 2 vCPU / 8 GB instance. Non-device software under FD&C Act §520(o)(1)(D).*