



STRATEGIC ESSAY

Editorial date: 23 October 2026

When Dammam Became a Factory for Seconds

Groq, specialized inference silicon, and the thermodynamics of useful AI

At a demonstration for a Saudi audience, the striking part was not that a language model answered a question about Vision 2030. Dozens of models can do that. What mattered was what Groq founder Jonathan Ross and Tareq Amin, then leading Aramco Digital, tried to show at the Global AI Summit in 2024: a voice conversation with a machine that no longer forced the human speaker to break rhythm while waiting for a reply. The companies presented a demonstration, not a published independent latency test. Still, it exposed the question around which Groq had built an entire company: what changes when the delay between two words becomes a problem in processor design itself? [S01](#)

Months later came an announcement unlike the usual news about a data center build. Groq says it brought an inference cluster in Saudi Arabia into service in **eight days in December 2024**. In February 2025, it said its Dammam center was processing requests from customers around the world through **GroqCloud**, alongside the announcement of a **\$1.5 billion Saudi commitment** to expand its infrastructure. These are figures and events reported by the company itself. They do not mean that the entire sum has been spent, that every planned unit of capacity has been installed, or that the announcement constitutes an export ledger. They are enough, however, to alter Saudi Arabia's position on the technical map: not every answer delivered to a Saudi phone must have been computed abroad, and an answer requested by a developer elsewhere may be computed on Saudi soil. [S02](#)

That opens a question deeper than the trade in servers. Why can the place capable of producing the next token in a fraction of a second matter more than the place that merely stores the model's weights? Why did a company that designed a chip for language generation make the Gulf a station in a global network, then enter a licensing and architectural relationship with NVIDIA, its apparent natural rival? And beneath the elegance of the speed figures lies a harder question: if Saudi Arabia manufactures shorter waits, where does the electricity go, and which water carries the heat away?

This is not an essay about outages, governance, or “model sovereignty” as a slogan. It examines the **industrial production of inference time**. That industry begins in the electrical quiet of a small memory array on a chip, passes through the compiler, scheduler, and thermal loop, and ends in a resource bill that can be counted. At its center is a less obvious Saudi opportunity: **to export measurable time performance, rather than merely host a file called a model**.

An answer does not arrive all at once

Suppose you ask a voice assistant about an industrial report. The listener hears sentences; the inference system sees a **sequence of tokens** selected probabilistically by a neural network. The question first enters **prefill**, when the context tokens are processed with substantial parallelism and the system creates internal representations and attention key-value state, or the **KV cache**. Then **decode** begins. The model produces one token, which becomes part of the context required to produce the next. This is not a literary division of time. It divides the physics of computation into two different operations.

Prefill is rich in matrix multiplication and large computational operations. Decode for a single request repeatedly passes over model weights and attention state, and can be constrained more by moving data through memory than by arithmetic. It is therefore not enough to say that one chip has more trillions of operations per second. The more precise question is how much computation it performs for every byte fetched from memory. Performance engineers call this **arithmetic intensity**. The **Roofline model** helps them distinguish programs limited by compute capacity from programs limited by memory bandwidth.

A factory can increase total output by adding production lines, but that does not answer the customer waiting for an order now. In the same way, **aggregate tokens per second** differ from **tokens per second for one user**. A provider may celebrate a very large figure assembled across thousands of concurrent requests while a single user still experiences a slow conversation. Four different measures lie between those perspectives: time to first token (**TTFT**), inter-token latency, queue time, and end-to-end latency from submitting the request to receiving its final token. Groq's operational documentation distinguishes these measures and also separates latency-prioritized services from batch work that does not require an immediate interaction.

S09 S10

Consider an arithmetic example unrelated to the speed of any particular chip. Generating 180 tokens at 300 tokens per second takes 0.6 seconds in theory. At 900 tokens per second, it takes 0.2 seconds, a gain of 0.4 seconds in that stage alone. This excludes sending speech to the model, speech conversion, external tool calls, and any wait before generation. Yet if a software agent performs four consecutive generation rounds of the same size, that stage-level saving may accumulate to 1.6 seconds. In agentic systems, sequential token speed compounds across steps. The economics of inference therefore differ from the economics of training a model once.

Why Groq moved the schedule into the chip

Groq calls the chip at the heart of its proposition a **Language Processing Unit**, or LPU. The term may be closer to a company-defined category of specialized computing than to a category settled in computer architecture textbooks, but the engineering behind it deserves attention. Modern GPU systems are powerful and general-purpose. They use high-bandwidth memory (**HBM**), tensor cores, schedulers, instructions, and software layers to support a wide range of deep-learning workloads. Groq instead chose an architecture in which **the order of data movement and operations is largely known before execution begins**. [S04](#) [S05](#)

In a more conventional processor, the system decides at runtime which ready resources can execute which work. In **statically scheduled dataflow**, the compiler assigns operations and data movement to a timetable. Results move through processors, memories, and links like material through a production line that knows when each item should reach each station. This image explains the disclosed architecture; it does not imply that an entire external service becomes deterministic. Queues, networks, contention among users, and model variation remain real sources of latency variance. Predictable data movement inside the machine nevertheless removes some decisions that hardware would otherwise make during execution. [S04](#)

Groq also chose on-die **static random-access memory**, or SRAM, rather than relying primarily on fetching weights from high-bandwidth memory outside the processor die. SRAM holds bits in fast transistor cells and does not require refresh cycles like DRAM, but every bit consumes more silicon area. The trade buys speed and low latency at the cost of storage density. That trade, rather than the isolated slogan of “ten times faster memory,” is the point of the architecture. Groq cites more than 80 terabytes per second of SRAM bandwidth for one chip generation and compares it with roughly eight terabytes per second for some GPU/HBM configurations. This is a **vendor comparison conditional on how the path and generation are defined**. It does not judge every GPU or establish that complete inference is ten times faster for every model. [S04](#)

The harder engineering question sounds simple: **where do the weights go?** A hypothetical 70-billion-parameter model at 16-bit precision requires about 140 billion bytes for weights alone, or **140 decimal gigabytes**, before attention state, temporary space, replication, and coordination across chips. At eight bits per weight, the raw size falls to 70 gigabytes, but deployment still carries costs in numerical precision, quantization scales, the KV cache, and partitioning. NVIDIA's 2026 **Groq 3 LPX** rack design contains 256 accelerators and **128 gigabytes of aggregate SRAM**, with a claimed 40 petabytes per second of in-rack memory bandwidth. Even in this newer generation, the uncompressed 16-bit weights of a 70B model can exceed the SRAM in one rack. That does not mean the system cannot serve such a model: weight partitioning, quantization, other memory, accelerators, and components all matter. [S07](#) [S08](#)

The closer data sits to compute, the less time the system spends fetching it. The corresponding obligation is to treat software as a **map of weight placement, memory use, and communication across chips**. The local technical capability worth developing therefore begins beyond the chip purchase: compiler-aware placement, quantization that preserves application quality, scheduling variable-length batches, and measuring how often KV state must move between resources. A country need not invent a chip overnight to develop teams expert in these operational layers.

Conceptual comparison of GPU and LPU memory and dataflow

Figure 1. An original explanatory diagram, not a manufacturing schematic or a measurement of either architecture's performance. Architectural sources: [S04](#), [S07](#), and [S08](#).

Three real episodes, and three different kinds of “success”

First, from the demonstration stage to Dammam. In September 2024, Aramco Digital and Groq announced cooperation on a large inference center and proposed linking it to the **nawat** digital-services marketplace. A voice conversation appeared on stage. It demonstrated a narrow interaction, not that all planned infrastructure existed. Then came the reported installation of a cluster in eight days in December, followed by the February 2025 announcement that GroqCloud was serving customers from a Dammam center. These events differ materially. A cooperation agreement creates a possibility; a demonstration establishes a bounded scenario; an announced cloud service indicates an operating product for developers. None of the public documents reviewed here provides an electricity profile, a water-use profile, or a count of tokens produced in Dammam alone. [S01](#) [S02](#)

Second, a local model does not require a locally manufactured chip to run locally. At LEAP 2025, Groq demonstrated Saudi Arabia's **ALLaM** model alongside reasoning and speech-generation models in English and Arabic. In May, Groq said it had become an official inference

provider to Saudi company **HUMAIN**. The first episode demonstrated models and use cases; the second announced a supply relationship. The distinction matters: **a model may be developed in Saudi Arabia**, while **the processor is designed and manufactured in the United States**, according to Groq's disclosure, and **the request is computed in Saudi Arabia**. These are three different layers of the value chain. Sharing a national label does not make them one. [S02](#) [S03](#)

Third, the rival became part of the machine. In December 2025, Groq announced a **non-exclusive** inference-technology license with NVIDIA. Jonathan Ross, Sunny Madra, and engineers moved to NVIDIA, while Groq remained an independent company according to its statement. In August 2026, NVIDIA announced that **Groq 3 LPX** had entered full production and named **Nebius** as the first cloud provider to adopt it. Groq separately said it **planned** to be among the first adopters, working with Dell. "Entered production" must not be read as "installed in a Saudi center." No published evidence reviewed here establishes such an installation. [S06](#) [S07](#) [S12](#)

NVIDIA's new design combines two computing families: GPU-based **Vera Rubin NVL72** systems and specialized LPX racks. NVIDIA describes the components as working across model layers to serve long contexts and fast generation. This is not a story in which one chip wins and the other retires. It is an architectural acknowledgment that **inference is not a single operation that necessarily belongs on one kind of silicon**. Dense context processing may suit one mode of computing while sequential token generation suits another, with assignment determined at the level of work, network, and memory. NVIDIA also reported about **3,400 output tokens per second** for *Gemma 4 31B* with a 100,000-token context, citing Artificial Analysis. That figure belongs to the specified model, test, and platform. It is neither a promise to every customer nor a Dammam result. [S07](#) [S08](#)

A small geographic inconsistency also argues against drawing a confident map. Statements from 2025 describe operations from **Dammam**, while Groq's current platform page lists **SA-1 — Riyadh** among a network of 13 global centers. The page does not explain whether this is a different operational address, a new site, or a change in how access points are classified. **We cannot count Dammam and Riyadh as two distinct centers or assume that one replaced the other.** We know that Saudi Arabia has become an announced operating node in Groq's network. We do not know its exact composition or present workload distribution. [S02](#) [S08](#)

A token can leave the chip and still arrive too late

If a processor generates one thousand tokens per second but spends too long waiting in an admission queue, the application loses something absent from the specification sheet. If an agent calls four tools across distant continents, faster decode cannot remove network or tool-

execution time. A **request-time map** is therefore more useful than a single headline rate: network round-trip time, queue time, prefill, the gap between decode tokens, tool waits, and the rate at which complete tasks satisfy their quality requirements.

Groq's own commercial documentation supplies a practical picture without requiring us to invent an experiment on its systems. It defines **performance** tiers for tight response requirements, **on-demand** access for ordinary requests, and **flex** service that trades some guarantees for broader processing limits. It also provides a separate **Batch API** with different windows and limits. These products do not show that Groq operates a thermal scheduler. They establish that a real inference service can recognize **different time values for different work**. A live phone conversation cannot wait for cooler weather. Summarizing a large archive or running a model-evaluation suite may have hours of flexibility, if the contract and data permit it. [S09](#) [S10](#)

That suggests a more exact commodity for a Saudi cloud operator: **a quality-accepted token delivered within a known deadline, with the electricity and water required to produce it accounted for**. A raw token count can mislead. A system may generate unwanted length, repeat a failed answer, or finish a batch after its deadline. A chip's idle draw and cooling power do not appear in the token count at all. The industry needs three intersecting meters: useful work, valid time, and physical resources.

Weather can become an input to the algorithm

Computer scientists call placing a job in a suitable time slot **scheduling**. But what happens if outdoor air enters the job's cost function? The temperature displayed on a phone is not enough. Cooling engineers distinguish **dry-bulb temperature**, the familiar reading of sensible heat, from **wet-bulb temperature**, which reflects the potential for evaporation at the available humidity. Hot, dry air may support evaporative cooling differently from warm, humid coastal air. A coastal night is not necessarily a "cool night" in terms of the air's capacity to accept more water vapor.

Electricity does not disappear inside an inference center. Most of it eventually becomes heat that must be carried away from processors, memory, and power supplies. Liquid can collect heat through **cold plates**, then move it through pumps and heat exchangers to equipment outside the facility. A closed liquid loop inside the rack **does not prove that a center consumes no water**, just as liquid cooling does not imply that every facility uses evaporative towers. The water consequence depends on heat rejection, make-up water, treatment and discharge, and the electricity source. [S11](#) [S13](#)

A **dry** heat-rejection system can transfer heat to the air without significant on-site evaporative water, but may require more electricity when outdoor air is hot. An **evaporative** system may reduce cooling electricity under some conditions while consuming on-site water. Neither method wins in every hour and place. The US Department of Energy defines **PUE** as facility energy divided by IT-equipment energy and site **WUE** as facility water per kilowatt-hour of IT energy. Neither measure alone says how many useful tasks were completed or how much water was consumed in producing electricity beyond the facility boundary. [S11](#)

A simple physical observation gives the scale. Evaporating one kilogram of water around 25°C requires about **2.44 megajoules** of latent heat, while one kilowatt-hour equals 3.6 megajoules. Under the **purely idealized** assumption that evaporation removes all the heat from one kilowatt-hour, theoretical evaporation is about **1.47 liters of water**. This is not the WUE of any center, water per question, or an estimate that includes blowdown, losses, and the electricity supply chain. It is an energy relation for a phase transition, useful because weightless-looking bits still produce physical heat. [S14](#)

A reproducible experiment: can a scheduler move heat across time?

For this work, we ran a small computational test that was **fully executed and entirely synthetic**. We created a 48-hour horizon and two invented weather series: one “hot and dry,” the other “humid coastal.” We did not call them Riyadh and Dammam because **we did not obtain hourly Saudi observations**. They are plausibly shaped mathematical scenarios, not weather records. Each hour includes interactive work that must run immediately and flexible batch work that may be deferred by up to nine hours. We set an IT-energy ceiling, a fixed idle load, and separate electricity and water functions for dry and evaporative cooling. For each water weight, a separable first step selected the lower-cost cooling mode in every hour under linear costs and free mode switching; a **linear program** then assigned batch work to hours while preserving every deadline and **exactly the same quantity of useful work**. This is an analytical simplification, not a realistic model of cooling-plant controls. The coefficient that trades one liter of water against one kilowatt-hour is a mathematical weight, not a Saudi market price. The full assumptions, equations, and executable materials are available in the reproducibility archive.

In the synthetic hot-dry scenario, a policy that starts batch work at the first opportunity and selects the lower-electricity cooling mode used **5,192.71 kWh and 4,829 liters of water** over 48 hours. Optimizing batch timing and cooling for electricity alone reduced energy to **5,179.88 kWh**, with about **4,167 liters** of water. Adding a modest water weight of **0.02 kWh per liter** moved the solution to **5,188.14 kWh and 3,559.52 liters**. Against the minimum-electricity

solution, that buys a reduction of roughly **607 liters** at an additional **8.26 kWh**. This is not a real savings rate that can be generalized. It is conditional on cooling functions designed for the research exercise. S17

The synthetic humid-coastal scenario produced a different trade-off. The minimum-energy solution used **5,245.08 kWh and 4,893.67 liters**. With the same water weight, it used **5,263.52 kWh and 3,640.80 liters**: about **18.43 additional kWh** for roughly **1,253 fewer liters**. All 48 batch jobs met their windows in both scenarios, and useful work remained about 4,394.2 units. The result therefore shows a possible mechanism rather than an artifact of dividing unlike totals. Its lesson is not that “cooler is better,” but that **the time of useful work can be aligned with the marginal electricity and water cost of rejecting its heat**. S17

Synthetic electricity and cooling-water frontier under two invented weather profiles

Figure 2. Results from an executed linear program using invented inputs. Every point completes the same work without missing deadlines. Both axes are units of the synthetic model, not measurements from a Saudi facility.

Interactive work never moves in this experiment, so the frontier is not purchased by slowing a live conversation. Only work with a real scheduling window can move. A thermal scheduler is therefore not a universal switch for all requests. Its potential is determined by the **time flexibility that actually exists**. When there is none, software cannot pretend to change the weather.

Batch work shifted through a synthetic day

Figure 3. An example of batch timing in the synthetic hot-dry scenario. The lines come from the local computation; they do not depict actual Groq or HUMAIN workloads.

From the compiler scheduler to the city scheduler

There is a useful conceptual link between what Groq does inside the chip and the proposal outside it, but it is **a similarity in coordination logic, not an identity of function**. At the first level, the compiler orders matrix operations and moves results through chip units. It cannot wait for an entire city to cool before producing the next token for a live user. At the second, the service scheduler assigns an evaluation batch or document-processing job to an hour, site, and cooling configuration. One acts over clock cycles or nanoseconds; the other over minutes or hours. Both ask the same question at different scales: **which work can move without violating its conditions of correctness and time?**

Intellectual restraint requires acknowledging the work already under way. **Balancing Bits and Drops**, released as a preprint in 2026, examines workload scheduling under water use and local water stress. **PowerSlider**, presented in August 2026 on SGLang, addresses instantaneous power limits and reports experimental goodput for phase-aware GPU frequency controls in its own setting, not in a Saudi center and not on an LPU. The proposed contribution here is not to invent “environmental scheduling” from nothing. It is to **connect time-varying cooling conditions to the actual specifications of inference work in Saudi facilities, then measure deadline-valid work through comparable telemetry.** [S15](#) [S16](#)

A real operations team would begin by recording a request trace: prefill size, output-token count, KV-cache size, number of tool steps, request start and first-token time, the user’s maximum acceptable delay, and how often the task had to be retried. The physical layer would add rack PDU energy, pump and fan state, heat-exchanger efficiency, dry- and wet-bulb temperatures, make-up water meters, and cooling-flow limits. Better quantization or shared **prefix caching** may matter more than moving a job to the evening. Separating prefill and decode across specialized resources may precede any thermal intervention. Rhetoric cannot rank these measures; useful output per unit of resource can.

Where the idea itself can fail

A prudent engineer may object that moving batches into another hour concentrates demand, raises the peak, or reduces server utilization. If chips remain powered while idle, the expected saving may never appear. Moving **KV state**, or reloading a model into less congested memory, can consume more electricity and time than the thermal saving. Cooling performance may depend on supply-water temperature as much as outdoor air. A policy that looks best for site electricity may look worse once the water used to produce off-site electricity is included, which a local WUE reading cannot reveal.

There is a sharper economic objection. If Groq and its peers make inference faster, applications may consume more tokens rather than reduce their total bill. When an agent replies faster, a user can ask it to perform more cycles of reading, experimenting, and checking. **Energy per token** may fall while **total energy consumption** rises, or the workload mix itself may change. A 2025 Lawrence Berkeley National Laboratory study found that estimated workload-level water consumption could vary by more than four orders of magnitude across assumptions and facilities, driven by server efficiency, the water intensity of grid electricity, utilization, cooling systems, and climate. That is not a Saudi measurement. It is a scientific warning against inventing a fixed number of “liters per question.” [S13](#)

Finally, when the task is a live voice response for a user on another continent, network proximity may matter more than the cooling advantage of a distant site. A location able to export compute globally is not necessarily the best location for every request. Some

applications are sensitive to network delay; others tolerate batch work that leaves and returns later. A mature industry needs **two different commercial workloads**: immediate work that buys fractions of a second, and flexible work that buys lower resource cost within a deadline. The first should not carry every water experiment that the second can test without damaging the user experience.

What Saudi Arabia could build between 2027 and 2035

Before claiming any global title, a Saudi inference center could run one serious industrial experiment. Select three real workloads with clear service contracts: live voice conversation, a multi-step coding agent, and model-evaluation batches with an eight-hour deadline. Run each on a GPU platform and an LPU platform when model compatibility, licenses, and availability permit, while holding constant **quality, valid-token counts, and request-length distributions**. Measure p95 TTFT, inter-token latency percentiles, useful work per hour, rack and facility electricity, site water, and the water associated with electricity generation when traceable data exist. Repeat the experiment across seasons rather than on one demonstration day.

This is not a call to create a Saudi LPU industry from nothing in three years. Groq's original disclosure was explicit about developing, manufacturing, and assembling its chips in North America. The higher-value local question is whether Saudi teams can become excellent at **heterogeneous inference compilation**, context and KV-cache placement, and tests that make energy and water **per correct task completed within its deadline** auditable. Can they build software that selects site and platform by expected output length, token-transfer costs, and weather, then export that software and operational expertise alongside inference service? That is a research, development, and operations proposition, not a result already achieved.

A more modest starting point requires only synchronized measurement and an A/B project inside one center, assigning a small share of flexible work to a **weather-aware scheduler**. If it reduces water without harming customer response or shifting the burden to more water-intensive electricity, the benefit is real. If it fails, the result remains valuable because it identifies the binding system constraint. "Success" must reside in useful work and physical resources, rather than a pleasing motion on a chart.

In the previous century, some industries exported energy embodied in aluminum, metals, and fertilizers: electricity helped transform matter into a saleable product. In the coming decades, another industry may transform electricity, silicon, and cooling into **paid cognitive time**: milliseconds removed from a voice interaction, or thousands of inference attempts completed by a scientific agent before a project deadline. This is a proposed economic analogy, not an accounting equivalence between industries. In each case, advantage lies in converting resources into a saleable output, and waste heat or memory wait becomes a defect in the product rather than a number confined to an equipment room.

The argument returns to the first scene. Groq demonstrated a conversation that appeared faster than usual. The strategic question left behind concerns more than the speed of an artificial voice: **what if Saudi Arabia could manufacture shorter waits for the world and know, task by task, the electricity and water cost of that time?** A data center would then become an engineered facility with a defined yield; the chip would be one of its instruments; and the scheduler would read the weather alongside sentence length. A gleaming rack photograph cannot answer the question. The answer requires experiments that move from tokens to completed interactions, from those interactions to meters, and back to a result another engineer can recalculate.

References

Vendor announcements are cited as vendor statements, not as independent verification. The synthetic computation is distinct from Groq, HUMAIN, any Saudi facility, and observed Saudi weather.

- **S01.** [Groq and Aramco Digital, partnership announcement, 12 September 2024.](#)
- **S02.** [Groq, Dammam operations and Saudi expansion commitment, 10 February 2025.](#)
- **S03.** [Groq, HUMAIN inference-provider announcement, 27 May 2025.](#)
- **S04.** [Groq, “The Groq LPU Explained,” 7 March 2025.](#)
- **S05.** [GroqChip Processor product sheet.](#)
- **S06.** [Groq, non-exclusive NVIDIA inference-technology license, 24 December 2025.](#)
- **S07.** [NVIDIA, Groq 3 LPX full-production announcement, 24 August 2026.](#)
- **S08.** [Groq platform and infrastructure page.](#)
- **S09.** [Groq production service tiers and Batch API documentation.](#)
- **S10.** [Groq Prometheus request-latency metrics.](#)
- **S11.** [US Department of Energy, *Cooling Water Efficiency Opportunities for Federal Data Centers*, 9 January 2019.](#)
- **S12.** [Groq, planned adoption of NVIDIA Groq 3 LPX and Vera Rubin, 2026.](#)
- **S13.** [Lei et al., “The water use of data center workloads,” 2025.](#)
- **S14.** [US National Bureau of Standards, measurements of water’s latent heat near 25°C.](#)
- **S15.** [Talukder et al., “Balancing Bits and Drops,” preprint, 2026.](#)
- **S16.** [Li et al., “PowerSlider: Exploiting Phase Asymmetry,” preprint, 2026.](#)
- **S17.** Reproducibility archive: code, synthetic inputs, computed outputs, and run documentation.
- **S18.** [Saudi National Center for Meteorology, weather-data API documentation; the API was not queried for this work.](#)