

CGRSP Defense: Speaker Script and Q&A Cheat Sheet

Deck: CGRSP\_defense.pptx , 16 main slides + thank-you + 5 backups. By the per slide timings below the main talk is about 22 minutes (21:45). If the slot is 20 minutes, cut slide 3 (contributions) to one sentence and slide 7 (methods) to one sentence; that saves about 1.5 minutes, so also trim slide 5. Full 600-line Q&A lives in cheat\_sheet.md (codes such as C4, J9 below point there).

Part 1. Speaker script

1. Title (0:15)

Good morning. My thesis is the Cloud GPU Rendering Scheduling Problem: a mathematical formulation and an empirical evaluation on heterogeneous GPU fleets, using real rendering jobs from RunPod.

2. Why scheduling rented GPUs is hard (1:30)

A rendering studio rents GPUs on demand. Three things make dispatch non-trivial. The fleet is heterogeneous: RTX 3090, A5000, A4500 and A4000 differ in speed and hourly price. Availability is stochastic: stock status drives how long a node takes to provision, and execution time is log-normal. And deadlines are mixed: an assumed 20% rush jobs with 1 hour deadlines and 80% standard jobs with 8 hours, all sharing a concurrency cap of C\_max = 5. The question: which dispatch policy keeps deadlines and waits low at minimal rental cost?

3. Contributions (1:30)

Six contributions. One, a complete formulation with stock-driven availability and a bi-objective QoS-then-cost objective. Two, a discrete-event simulator calibrated on 6,627 real jobs. Three, a ten-policy comparison with 30 paired seeds and Holm-corrected tests. Four, two new schedulers, RH and CADR. Five, a regime map: where EDF and SPT trade places, the C\_max lever, monthly scale. Six, deployment rules grounded in paired tests.

4. Formulation (1:30)

The objective is lexicographic: first minimise QoS cost, which is weighted wait plus missed deadlines, and only then minimise rental cost. A missed client deadline dwarfs any rental saving, so cost acts as a tiebreaker. Constraints: each job eventually executes once, dispatch follows submission, and completion equals start plus execution. One job holds a node at a time, with at most C\_max slots. Inputs are stochastic: log-normal execution time per complexity and GPU, and provisioning delay driven by stock status and time of day. The offline problem is NP-hard and online has no bounded competitive ratio, which is why we study heuristics.

5. Worked example (1:30)

Three jobs, three GPU types. j1 is tight, 1 hour, and goes to the RTX 3090. j2 and j3 have 8 hour deadlines; only one A4000 is idle, so j3 takes it and j2 takes the cheapest slot left, an RTX A5000. At this load j1's one hour window does not bind, so all three jobs are QoS-equivalent and cost decides placement, except that j1, being tight, takes the 3090, the fastest type for low complexity. This is the lexicographic rule in action: QoS first, cost only among QoS-equivalent options.

6. Simulator (1:00)

The simulator has a job generator, a scheduler, a fleet manager and a metrics collector around a discrete-event core. Calibrated on 6,627 real jobs from July 2025 to June 2026, ten schedulers, 30 paired seeds for the headline and main sweeps, 10 for the parameter sweeps and the monthly runs.

7. Methods and validation (1:00)

Every scheduler sees identical jobs per seed, so tests are paired. Paired t-test with Wilcoxon backup, Holm-Bonferroni over a family of 24 comparisons, Cohen's d. A constraint verifier re-checks all ten policies on the seed 42 hectic run, 950 jobs each, zero violations, and the baseline sweep row reproduces Phase 3 exactly. The pipeline is deterministic with a pinned hash seed.

8. The ten policies (1:30)

Eight baselines: FIFO, SPT, EDF, LCF, Balanced, Random, SPT+Rescue and Adaptive. Two proposed. RH, Rolling-Horizon, projects the slot timeline: it keeps SPT order for jobs with slack, promotes jobs that would otherwise miss, and demotes jobs that are doomed so they do not displace savable ones. CADR classifies jobs by a scale-free critical ratio and picks the cheapest deadline-feasible node.

9. Ablation (1:30)

Hectic day, 30 seeds. FIFO misses 24.85%, SPT 18.60%, adding rescue gives 12.50%. Ordering by deadline risk alone, without cost-aware placement, gives 10.28%. Full CADR 10.09%, RH 9.50%. So the gain lives in job ordering, not node choice. RH beats

SPT+Rescue on miss after Holm correction, but not on wait.

10. Hectic-day benchmark (1:30)

RH has the lowest miss, 9.5%, tied with CADR at 10.1% (p = 0.484). Their difference is in tardiness: mean tardiness per job, with on-time jobs counted as zero, is 6.4 minutes for CADR versus 30.9 for RH (about 64 versus 325 minutes per missed job). Wait: RH 140 minutes, tied with SPT at 141. So the choice is count versus severity.

11. Where EDF and SPT trade places (1:30)

Surge-day load, sweeping the tight-deadline share. EDF has significantly lower miss at 30% tight or less, the two tie at 50%, and SPT wins at 70% or more. In the shared hectic benchmark EDF beats SPT on miss, 15.81% versus 18.60%, p = 0.002.

12. Concurrency cap (1:30)

Hectic day, 10 seeds. Raising C\_max from 5 to 7 drops miss below 3% for all three policies and to 0% for CADR. At 8, our current quota, every deadline-aware policy has zero miss and FIFO 0.5%, so the policies tie there. More slots, not a better ordering, remove the saturation misses; the scheduler is the free lever when the cap binds, as it does at the default 5.

13. Monthly scale (1:30)

One continuous 30-day calendar, 10 runs, with the day-type mix from billing data. Daily arrivals stop at the next boundary, accepted jobs carry over, and the final backlog drains. RH 6.76% versus CADR 7.14% miss, tardiness 19.3 versus 6.3 minutes. No paired policy test is reported at this scale, so these are point estimates. RH's Phase 3 wait lead does not persist.

14. Snapshots contradict the assumed ranking (1:30)

I swept stock probabilities. From 0.75x to 1.00x deadline-aware beats deadline-blind and RH is lowest; RH ties CADR at 1.00x and is ahead only at 0.75x. At 0.5x and below nothing copes. At 1.25x several policies reach zero miss while RH keeps 5.4%. Our own pricing snapshots rank GPU types in the reverse of the assumed order. That is listing share, not fulfilment, and beta stays an assumption, but under that ranking CADR and SPT+Rescue tie for lowest and RH is significantly worse than CADR. So RH's lead is conditional, and CADR is the safer default.

15. Deployment guidance (1:30)

Default to CADR: ties RH on miss, has close monthly miss point estimates and much lower tardiness. Use RH as a specialist in the scarce-availability regime when the missed-deadline count is what matters; operators who prioritise queue wait should use SPT. Raise C\_max above 5 if residual misses must go to zero; at our quota of 8 every deadline-aware policy reaches zero miss on the hectic day. SPT stays a strong low-wait baseline.

16. Limitations and future work (1:30)

Simulation only, and no tight optimality bound: the exact ILP is intractable at about 950 jobs and I did not attempt it even at toy size, and the certified capacity bound shows no misses are unavoidable on the expected-service instances but says nothing about the stochastic online optimum. Hectic and surge are assumed stress loads: the busiest recorded day has 415 jobs. Availability probabilities are assumed. Single studio, Pods price snapshot, non-preemptive jobs. Future work: live-pipeline validation, spot and reclaim dynamics, learned dispatch, multi-provider fleets.

17. Thank you Questions.

Part 2. Numbers to have in your head

| Fact                                     | Value                                                                                                                                               |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Dataset                                  | 6,627 jobs, Jul 2025 to Jun 2026 (6,408 production, 218 tagged dev, 1 untagged)                                                                     |
| Fleet                                    | 2x RTX 3090, 1x each A5000, A4500, A4000; C_max = 5                                                                                                 |
| Phase 3 miss                             | RH 9.50, CADR 10.09, SPT+Rescue 12.50, Adaptive 12.49, EDF 15.81, SPT 18.60, Random 24.04, FIFO 24.85, LCF 26.14, Balanced 26.39                    |
| Phase 3 wait (min)                       | RH 140.24, SPT 141.03, CADR 145.70, SPT+Rescue 145.85                                                                                               |
| Phase 3 tardiness (min)                  | CADR 6.43, RH 30.86, SPT+Rescue 6.64, EDF 5.74, Adaptive 4.45                                                                                       |
| RH vs CADR miss                          | tied, p = 0.484                                                                                                                                     |
| EDF vs SPT miss                          | EDF lower, p = 0.002, d = 0.73                                                                                                                      |
| Phase 4 (30 days, 10 runs)               | miss CADR 7.14, RH 6.76; tardiness 6.26 vs 19.30; no paired policy test                                                                             |
| Statistics                               | 30 paired seeds, family of 24 tests, 18 significant raw, 12 survive Holm                                                                            |
| C_max (10 seeds)                         | SPT 51.5/18.9/2.5/0, RH 46.5/8.4/1.7/0, CADR 49.0/9.1/0/0 at C_max 3/5/7/10; at 8 (current quota) all deadline-aware 0, FIFO 0.5                    |
| Day types (jobs)                         | quiet 6, normal 100, surge 730, hectic 950 (stress-load assumption of the order of the busiest billing day, about 1,070 jobs); busiest real day 415 |
| Beta sweep                               | deadline-aware beats blind x0.75 to x1.00, RH lowest, tied with CADR at x1.00; x1.25 CADR/EDF/Rescue/Adaptive 0%, RH 5.36%                          |
| Snapshot-ordered ranking (beta permuted) | CADR 3.04, SPT+Rescue 3.24, RH 6.24 (CADR and SPT+Rescue tie for lowest; RH worse than CADR, p < 0.0001)                                            |
| Tests                                    | 169 unit tests, verifier checks 950 jobs in each of 10 runs, zero violations                                                                        |

Part 3. Never say

- "SPT and EDF are tied on miss." EDF is significantly lower (p = 0.002).
- "SPT has the lowest wait." It is tied with RH (p = 0.671).
- "RH Pareto-dominates SPT+Rescue." Miss yes (survives Holm), wait no (fails at rank 13 of 24).
- "RH is statistically best at monthly scale." RH has the lower miss point estimate and CADR lower tardiness; no paired policy test is reported.
- "delta\_j = 1 means missed." delta\_j = 1 means on time.
- "RH pushes doomed jobs further back" as the tardiness explanation. It is promotion: RH promotes tight jobs too late (C4).
- "Every significant result survived Holm." Only 12 of 18 did.

Part 4. Q&A, short answers

Ordered by how likely and how dangerous. Each: say the first line, then the pointer.

Model assumptions

Q1. Availability probabilities are assumed, not measured. Isn't the model made up? (A1, J5) Yes, they are stated assumptions, so I swept them from 0.25x to 1.25x. Absolute miss levels are not transferable (RH moves from about 10% to 43% between 1.00x and 0.75x, and to about 5% at 1.25x); I claim only the policy ordering, and only for 0.75x to 1.00x. Below that every policy collapses, above it the problem is easy and CADR overtakes RH. The sweep keeps the per-type ranking fixed; the data-ordered permutation covers that (J9). Next step: measure per-type fulfilment from dispatch logs.

Q2. Your snapshots rank the 3090 best and the A4000 worst, the reverse of your assumption. (J9) Conceded, and the thesis says it (5.5.8). Snapshot listing share is not fulfilment probability, so beta stays an assumption, but it is the only per-type signal I have. Under that ranking CADR 3.04% and SPT+Rescue 3.24% tie for lowest, and RH 6.24% is significantly worse than CADR. That is why CADR is the default and RH the specialist. The no-prior ablation shows RH's prior is inert (9.72% vs 9.50%, p = 0.62), so the effect is the ordering interacting with the ranking.

Q3. The busiest real day has 415 jobs. Where do 950 and 730 come from? (J2) They are stress-load assumptions: hectic is a saturating burst chosen to put the fleet in rho > 1, which is where scheduling matters most. Surge matches the average spend of surge-class billing days; hectic is a stress-load assumption of the order of the busiest billing day (about 1,070 jobs). Read

Phase 4's absolute miss as a stress month and the policy ordering as the result. Trace-driven replay is future work.

**Q4. Production showed up to 8 concurrent jobs. Why C\_max = 5? (J1)** C\_max = 5 is RunPod's default account quota, used as the baseline budget. Our current quota is 8 and the sweep {3, 5, 7, 8, 10} includes it as its own row: at 8 every deadline-aware policy, RH and CADR included, has zero miss and FIFO 0.5%. At the default 5 the scheduler matters most, so 5 is the stress case and 8 is the operating point. The saturation mechanism holds across it.

**Q5. Production ran on Serverless but you price Pods. (A7, J3)** Deliberately: persistent exclusive slots, provisioning not billed, no reclamation. Serverless supplied execution times and the default quota. Absolute costs are optimistic (about 1.5x to 2.5x), and that is why cost is only the tiebreaker; cost spread under saturation is about 1.3%.

**Q6. Poisson arrivals are not realistic. (A4)** Day types are the coarse diurnal envelope; within a day Poisson is the standard memoryless model. Non-homogeneous Poisson with the empirical hourly rate is future work.

**Q7. Is the 20/80 tight/loose split measured? (I3)** Assumed; production logs have no deadline field. The tightness sweep varies it from 10% to 90% and maps the EDF/SPT crossover, so each conclusion carries its regime.

**Q8. Complexity comes from render-time terciles, so isn't it circular? (A2, G4)** Disclosed. Logs have no scene metadata, so terciles are the proxy; SPT ordering needs only the ordinal class, but Rescue laxity, CADR's critical ratio, RH timelines and every policy's node choice use the stratum mean  $W(\kappa)$ , which the ch5 bracket stresses. Deployment would classify from scene statistics. A bracketing experiment now exists (ch5): with noisy or uninformative labels RH and CADR stay the two lowest miss point estimates but lose several points, SPT falls above FIFO when the label is uninformative, and no real classifier has been measured.

Methodology

**Q9. Why simulation, not queueing theory or an exact solver? (B1, B2)** Heterogeneous machines, stochastic availability and mixed deadlines have no tractable closed form, and I did not solve a 20 to 30 job instance exactly, so I cannot quote an optimality gap at toy size. For the bound, the certified capacity bound and a clairvoyant relaxation both give zero unavoidable misses, so observed misses measure the price of online, non-preemptive, stochastic operation. RH's 9.50% is the smallest measured price.

**Q10. Calibration is not validation. What did you validate against? (I6)** Agreed. Done: simulator reproduces fitted per-stratum means and dispersion; 169 tests plus the constraint verifier; capacity-bound cross-check; sensitivity sweeps. No independent operational trace exists, which is the simulation-not-deployment limitation.

**Q11. Why n = 30, paired tests, Holm? (B5)** Same jobs per seed make a randomised block design, so pairing removes between-seed variance. Holm controls family-wise error across a fixed set of 24 comparisons, mostly each scheduler against FIFO plus a few head-to-heads: 18 significant raw, 12 survive, largest surviving  $p = 0.002$ . Ten survivors are against FIFO; the two head-to-head survivors are SPT vs EDF on miss and RH vs SPT+Rescue on miss. Under Wilcoxon with the same Holm step-down 11 of the 12 survive: EDF over SPT on miss (Wilcoxon  $p = 0.0039$ ) does not, so that one result rests on the paired t-test and is borderline (the margin is below  $1e-4$  and depends on the tied-difference approximation). Head-to-heads outside the family, such as CADR vs SPT+Rescue on miss, or CADR vs EDF or Adaptive, are exploratory and uncorrected, so I lean on effect sizes and the sweep ordering rather than p-values.

**Q12. Why no RL baseline? (B3)** An early REINFORCE prototype on an older simulator version did not converge, so it was removed for scope; I do not use it as evidence that RL cannot work. The thesis gives three reasons for excluding RL (ch6), and it is stated as future work.

**Q13. Who verified this? (I13)** Plainly: the planned independent verifier became unavailable, so no second person ran it. A scripted audit on 3 and 4 September reproduced every output byte for byte from a fresh

clone, recomputed every statistic from raw arrays and regenerated every figure. It found and fixed real defects. I can walk the committee through the checklist live.

Results

**Q14. Isn't RH just greedy with extra steps? (C2)** Placement is greedy per job. The tier orders are fixed rules (SPT, EDF, demote-last); the contribution is that the urgent/normal boundary and the doomed-job test are read from the projected slot timeline instead of a hand-set threshold. Under saturation about 85% of hectic-day decision events with a busy slot (seeds 0 to 2) have t\_free collapsed to now, so RH then orders mostly as SPT with hopeless demotion. Ablation: RH cuts miss from 12.50% to 9.50% at equal or better wait.

**Q15. Why is CADR's tardiness so much lower than RH's? (C4)** Promotion, not demotion. Any job RH demotes, CADR demotes too, so the difference is not in demotion. RH promotes a job only in the last window before it cannot wait (about one service time, planned without provisioning delay); a tight job missing that window becomes hopeless and waits behind loose backlog. CADR's at-risk tier opens at  $CR \leq 3$ , about three service times early, so tight jobs leave with margin.

**Q16. Why change your recommendation to CADR so late? (C11)** The September recommendation change altered only wording; the later continuous-calendar correction updates the monthly numbers. The Phase 3 paired tests show RH's miss lead is a tie at baseline ( $p = 0.484$ ), significant only at 0.75x among the operable scales (at 0.5x and below RH is also lower, but the platform is inoperable for every policy there), and reversed under the snapshot-ordered ranking. CADR holds up across settings and has far lower tardiness. RH is still a contribution: forward-timeline triage, lowest-tier wait, load-gated anticipation.

**Q17. Monthly RH has lower miss but CADR lower tardiness. Which should I choose? (C10)** State the point estimates: RH 6.76% and CADR 7.14% miss; mean tardiness 19.30 and 6.26 min. No paired policy test is reported. CADR is the default for low miss and much lower lateness; RH is the specialist when only the breach count matters; SPT for queue wait.

**Q18. Raise C\_max to 7 and everything is near zero. Why a scheduler? (C8)** C\_max is a paid platform limit; the scheduler is the free lever. At C\_max = 7 FIFO still lags. Guidance is both: smart policy at fixed C\_max, raise C\_max if residual must reach zero.

**Q19. Cost differences are negligible. Why cost-aware schedulers? (C6)** Saturated fleets run flat out so cost converges (1.3%). Off-peak and monthly it matters (Phase 4 spread 11.9%). Cost is a lexicographic tiebreaker by design, and on this fleet its demonstrated value is nil: costs converge because utilisation is about 96% at saturation. I do not claim CADR's cost awareness saves money; RH uses a symmetric surrogate internally, which ch4 discloses.

**Q20. At 1.25x RH misses 5.36% while others hit zero. Why? (G10)** RH's projection is certainty-equivalent, so a provisioning draw beyond expectation can doom a deferred job. With abundant availability its ordering collapses toward SPT. In a verified-easy regime run CADR, EDF, SPT+Rescue or Adaptive.

**Q21. 140 minutes mean wait: acceptable? (I8)** Hectic is deliberately above saturation ( $\rho > 1$ ), so queueing sets the wait scale; the comparison is relative. At C\_max = 10 waits fall to roughly 8 to 14 minutes. Normal days are under about 8 minutes for every policy.

Theory

**Q22. Why lexicographic, not a weighted sum? (D3)** The corrected weight study uses seconds of wait and missed-job fractions before normalisation. RH wins the tested binary-miss grid, so that grid does not demonstrate a reversal; lexicographic priority is the modelling choice that prevents a cost saving from compensating for worse QoS. Lexicographic encodes that a missed deadline dwarfs any rental saving.

**Q23. NP-hardness and competitive ratio? (D1, D2)** Offline CGRSP generalises weighted job scheduling on unrelated parallel machines with deadlines, strongly NP-hard (Pinedo). Online: Baruah et al. (1992) show under overload no algorithm guarantees more than a quarter of the clairvoyant value, even on one

preemptive processor. So I claim no worst-case guarantee and benchmark against FIFO.

**Q24. Why non-preemptive? (A6)** Frames render in seconds to minutes; checkpoint-restart costs more than it saves, and on-demand instances are not reclaimed.

**Q25. Is it novel given SkyNomad, REACH, Chugh et al.? (J7, G2)** On the combination. The spot literature models mid-job revocation; CGRSP is on-demand where scarcity is provisioning delay at acquisition. The critical ratio is classical (Pinedo cited); the claim is the combination of scale-free triage, doomed-last demotion and stock-filtered cheapest-feasible placement.

**Q26. What would you do differently? (F)** Non-homogeneous arrivals from the empirical hourly rate, a pre-render complexity classifier, a tighter stochastic lower bound, multi-platform validation, a preemptive extension.

Added after the full audit

**Q27. At your real quota of 8, what does your scheduler buy the studio?** At 8 every deadline-aware policy, RH and CADR included, has zero miss on the hectic day and FIFO 0.5%, so there is no miss benefit. RH even has the highest wait there (34.9 versus 24.7 min for SPT). The scheduler matters when the budget binds, as at the default 5, and below saturation on tardiness; at 8 the studio is already clear of a hectic day. These figures are the 10 seed sweep; the 30 seed Phase 3 numbers are at C\_max 5.

**Q28. Why CADR over EDF, Adaptive or SPT+Rescue?** At hectic saturation CADR is far below EDF (uncorrected paired test,  $p$  about 3e-5). Against Adaptive and SPT+Rescue the differences are exploratory and uncorrected; CADR vs Adaptive is borderline (paired  $t$   $p = 0.031$ , Wilcoxon  $p = 0.058$ ), so I do not claim CADR beats Adaptive. Below saturation EDF and Adaptive are as good or better on miss; CADR's case is steady performance across regimes plus low tardiness. The EDF advice at  $\rho < 1$  is relative to SPT and does not displace CADR as the default.

**Q29. What happens to the 1 hour rush jobs themselves?** Under RH in hectic overload 31% of rush jobs miss (loose jobs 4%). The 95th percentile tardiness averages about 5.8 hours across seeds and reaches 8.3 hours on the worst seed; for CADR it averages about 1 hour and peaks at 1.7. A CADR tight versus loose miss split is not in the stored outputs, so I do not quote one.

**Q30. Your capacity bound says a miss free schedule exists. Why do you miss 9.5%?** The bound is on expected-service instances with the fastest GPU and no provisioning delay, so it shows no miss is forced there, not that an online non-preemptive policy can reach zero. The miss rate at C\_max 5 is the price of online operation under the cap; extra slots absorb provisioning holding time and variance, which is why 8 reaches zero. I did not compute a hindsight bound on the realised stochastic instances, so how much of the 9.5 points a better scheduler could remove is unknown.

Part 5. Backup slide map

| Slide | Content                  | Use when asked about                     |
|-------|--------------------------|------------------------------------------|
| 18    | Scheduler lineage figure | how the ten policies relate              |
| 19    | Phase 3 full results     | any wait, miss, tardiness or cost number |
| 20    | Deadline sweep, miss %   | the EDF/SPT crossover detail             |
| 21    | C_max sweep, wait / miss | capacity, RH wait at C_max 10            |
| 22    | Phase 4 monthly results  | monthly numbers, cost spread             |

Part 6. Pre-defense checklist

- Defence and cover date 22 October 2026 (author decision). Confirm time limit and Q&A length with Prof. Hwang or the department office.
- Slides 11 and 14 content leaned on the audit critic's summary of ch5; skim ch5 5.5.2 and 5.5.8 once before the day.
- Open cheat\_sheet.md sections J (data exposures) and C (results) for a final read.