Performance Testing — Interview Prep Cheatsheet

Created by: Lamhot Siagian

Aligned to ISTQB® FL — Performance Testing: concepts, metrics, lifecycle, tasks, tools, and interview angles.

⚡ Basics & Principles

Area	Key Points
Purpose	Validate performance efficiency: time behavior, resource utilization, capacity.
Principles	Stakeholder-aligned goals • Repeatable & comparable tests • Representative env • Affordable/feasible.
Where	Web, mobile, IoT, client/server, distributed, cloud, mainframe, embedded real-time.

🧭 Types of Performance Tests

Type	Goal	Notes
Load	Expected/gradual load	Concurrency, realistic mix.
Stress	Beyond limits	Graceful degradation.
Spike	Sudden bursts	Autoscale recovery to steady state.
Endurance/Soak	Stability over time	Leaks, pool exhaustion.
Scalability	Future growth	Horizontal/vertical scaling thresholds.
Capacity	How much can SUT handle?	Users, TPS, data volume.
Concurrency	Simultaneous actions	Race/deadlock risks.

📊 Metrics & Measurement

Category	Examples	Tips
Time	Response time (avg/p95/p99), page/API latency, batch duration	Define context: data size, users, time of day.
Throughput	TPS/RPS, hits/sec, MBits/sec	Correlate with resource headroom.
Resources	CPU, memory, GC, I/O, disk, threads, connections	Minimize probe effect; use dashboards.
Business	SLA conformance, orders/hr, checkout time	Goal–Question–Metric (GQM) alignment.

🔁 Lifecycle & Risk

Phase	What to do	Risk focus
Plan	Scope, env, tools, data, risks, exit criteria	Env representativeness, load realism.
Analysis/Design	Derive objectives, select scenarios, define metrics	Architecture bottlenecks, SLAs.
Implementation	Scripts, data seeding, monitors	Think time, correlation, parameterization.
Execution	Warm-up → steady → ramp → ramp-down	Rollback plan, failure capture.
Results	Aggregate, compare to baseline, report	Residual risk to stakeholders.

📝 Performance Test Plan (PTP) Essentials

Section	Contents
Objectives	User vs technical goals; acceptance criteria & SLAs.
System & Env	Arch tiers, versions, env diffs vs prod, extrapolation rules.
Data	Production-like volume/shape; masking strategy; reset/backup.
Profiles	Operational → load (scenario mix, % per flow, arrivals).
Metrics	Minimal actionable set; p95, error rate, CPU/mem, TPS.
Risks	Tool limits, env scale, 3rd-party, observability gaps.

📈 Profiles & Load Modeling

Concept	Cheat notes
Operational profile	Real user flow(s) with probabilities; time-of-day patterns.
Load profile	Users/concurrency, arrival rate, mix %, think time, pacing.
Throughput vs Concurrency	$T \approx \lambda$ (arrival rate) • success% ; Watch Little's Law $N \approx \lambda \cdot W$.
Scenarios	Smoke (@smoke), baseline, peak, spike, endurance.

🧰 Scripting & Data

Topic	Guidance
Correlation	Capture dynamic tokens/IDs; parameterize inputs.
Think time	Use realistic distributions to mimic humans.
Data mgmt	Unique accounts, idempotent writes, cleanup hooks.
Assertions	Verify correctness (status, payload) to avoid false greens.

🚀 Execution Patterns

Stage	Why	Checks
Warm-up	Cache/JIT stabilization	Ignore initial samples.
Steady state	Stable metrics window	Use medians + percentiles.
Ramp up/down	Observe scaling & recovery	Error spikes, queue depth.
Chaos/failure drills	Resilience	Graceful errors, fallbacks.

🔍 Analysis & Reporting

Focus	How	Deliverable
Bottlenecks	Correlate latency ↔ CPU/mem/IO; link traces & logs	Hotspots by layer.
Percentiles	p95/p99 over time & by endpoint	SLA pass/violation chart.
Capacity	Find knee of curve	Max TPS at SLA.
Regression	Compare to baseline	Delta table + trend.
Narrative	Risk, causes, fixes, next steps	Exec summary (1 page).

🛠️ Tools & Monitoring

Area	Examples	Notes
Load gen	JMeter, Gatling, k6, Locust	Distributed agents, PoPs.
APM/metrics	Grafana/Prometheus, New Relic, Datadog	Dashboards, alerts.
Tracing/logs	OpenTelemetry, ELK	Span-level latency.

🧯 Common Failure Modes

Symptom	Likely Cause	Fix
Slow at all loads	DB design, N+1 calls, high network latency	Indexing, caching, batching.
Degrades at moderate load	Resource saturation, locks	Pool tuning, parallelism, CQRS.
Drifts over time	Leaks, unbounded queues	Profiling, GC tuning, backpressure.
Poor error handling	Tight timeouts, small pools	Graceful degrade, circuit breakers.

🏗️ CI/CD & Gates

Gate	Policy	Tip
PR Smoke	Light load @smoke must meet p95 SLA	Short jobs, consistent env.
Nightly	Baseline + endurance subset	Trend charts, alerts.
Pre-release	Peak + failover drills	Rollback criteria defined.

🎯 Interview Angles

Question	Strong Answer Angle
How do you size tests?	Operational→load profiles; Little's Law; SLA-driven objectives.
Which metrics matter?	Business outcomes + p95, error rate, resource headroom.
Find a bottleneck?	Correlate APM + traces + logs; bisect layers.
CI integration?	Tiered suites, env as code, perf budgets.
Data realism?	Shape, cardinality, privacy vs representativeness.

📚 Glossary (Quick)

Term	Short def
Think time	User pause between actions; affects concurrency.
Pacing	Rate at which a virtual user repeats a flow.
Baseline	Reference run for comparison across builds.
p95	95% of samples faster than this value.
Throughput	Work done per unit time (TPS/RPS/MBps).