
📑 PPT Essentials

Slide 1: Title

- Amaravati Quantum Valley Hackathon 2025
- Problem Statement ID & Title: *(use official PS from portal)*
- Theme: Logistics Optimization with Real-World Data
- PS Category: Software
- Team ID & Name: *(your registered details)*

Slide 2: Idea Title / Proposed Solution

Idea Title:
“Rust-Powered VRP Solver with Real-World OSM Integration”

Proposed Solution:

- A high-performance **Vehicle Routing Problem (VRP) Solver** implemented in **Rust**.
- Integrates **OpenStreetMap (OSM) real-world data** for accurate routing.
- **Parallel computing (Rayon in Rust)** to accelerate distance calculations and VRP solving.
- Outputs **GeoJSON routes** for direct visualization in map UIs.

Innovation & Uniqueness:

- Goes beyond abstract coordinates → routes tied to **actual road intersections**.
- Exploits **classical parallelism** for sub-second solving on large datasets.
- Scalable to millions of nodes (tested with 74M+ OSM nodes).

Slide 3: Technical Approach

Technologies:

- **Rust** (safe, high-performance systems programming)
- **Rayon** (data-parallelism library)
- **OSM/PBF parsing** for road networks
- **GeoJSON Export** for mapping/visualization

Methodology (Workflow):

1. Parse OSM data → extract roads.
2. Map depot/customers → nearest road intersections.
3. Generate VRP instance (vehicles, capacities, constraints).
4. Solve using parallelized VRP algorithms:
    - Greedy Nearest Neighbor
    - Clarke-Wright Savings
    - Multi-Start Solver (parallel execution)
5. Export → JSON/GeoJSON → visualize routes.

(Insert flow diagram: OSM → VRP Solver → Parallel Algorithms → Optimized Routes → Visualization)

Slide 4: Feasibility & Viability

- **Feasibility:**
    - Proven prototype with 100% test pass rate.
    - Handles real datasets efficiently (233m–795m routes computed in sub-second).
    - Scalable with parallel computing (Rayon).
- **Risks & Challenges:**
    - Large OSM datasets can be memory-heavy.
    - Requires efficient filtering & preprocessing.
    - Balancing route optimality vs computation time.
- **Mitigation:**
    - Parallelization to reduce computation bottlenecks.
    - JSON/GeoJSON exports for lightweight integrations.
    - Extendable to time-windows, real-time traffic.

Slide 5: Impact & Benefits

- **Impact:**
    - Urban delivery, logistics planning, emergency response.
    - Directly usable by businesses (routes → GPS).
    - Research & urban planning for traffic flow.
- **Benefits:**
    - **Economic:** Reduces fuel/time costs.
    - **Environmental:** Optimized routes → lower emissions.

    - **Social:** Faster deliveries, emergency services efficiency.

Slide 6: Research & References

- Parallel computing in VRP optimization (Rayon + Rust)
- Real-world coordinate routing with OSM
- Benchmark: 74M+ OSM nodes processed successfully
- Classical algorithms: Clarke-Wright Savings, Greedy Nearest Neighbor.

⚡ Positioning (to make it look promising)

- Emphasize **“Classical Computing at Scale”**: your solver shows how **parallelism on CPUs** can tackle logistics problems that many assume need quantum computing.
- Highlight **“Sub-Second Solving”**: strong performance proof.
- Stress **real-world grounding**: not just abstract VRP → OSM-based realistic routes.
- Show **scalability path**: from hackathon prototype → production-ready logistics platform.
