1 Introduction

Use Cases — Fundamental generalizes the many individual Use Cases from various application domains. This will appear in P3335 Section "10 – Applications and Best Practices".

2 Summary over Use Cases

A *Use Case* (UC) is a black-box property or capability that the system being standardized must or must not possess to meet the goals of the purchasers and users of that system. We have always had use cases by one name or another, and a major part of developing a standard was collecting and collating these implicit use cases into an overall picture. The collating process clarifies the terminology and reveals many unsuspected conflicts, gaps, and overlaps. The TimeCard Architecture (P3335 Section 05 – Architecture) emerged from the development of the TimeCard Use Cases[1].

"A *distributed system* consists of a collection of distinct processes which are spatially separated, and which communicate with one another by exchanging messages. … A system is *distributed* if the message transmission delay is not negligible compared to the time between events in a single process[2]".

All UCs are distributed systems (with shared data) that may be geographically large with multiple facilities connected by communication links, the system collectively performing matrix math on immense *dense* (not sparse) matrices.

Partial and total order[2] issues don't matter much for massive matrix math with noisy data but do matter for distributed databases and end-to-end control. Because the matrices to be solved are inherently very noisy, we will assume that partial ordering is always sufficient. If required, total ordering will be handled independently by bespoke application software not addressed here.

Here, *large* can mean from kilometers by kilometers to Solar-System scale[3], and *immense* can mean hundreds of billions (100*10^9) to many trillions (10^12) of rows and columns (mathematical dimensions).

Performance scaling and parallelizability[4] depend on maximum propagation delay, matrix dimensionality, shape, and noise content.

Signal propagation time is limited by the larger of electronics delay and the speed of light delay, and the delay at their crossover is important in smaller systems.

Scaling laws are useful for understanding and generalizing the overall behavior of these various approaches and algorithms. Scaling laws are compactly expressed in "Big O" notation[5].

The shape and mathematical dimensionality of the governing matrix to be solved may be 1D (vector), 2D, 3D, ... 10^12 D ... . While the traditional default is 2D matrices (often images of some kind), there are also applications where some of these matrices may be vectors or tensors.

---

[1] "TimeCard Architecture (Section 5) Draft (20250225).pdf"
[2] "Event Ordering in Distributed Systems (20230728).pdf"
[3] The speed-of-light delay between Earth and Pluto is about five hours and varies somewhat with the variation in relative planetary positions over time.
[4] .< https://en.wikipedia.org/wiki/Amdahl%27s_law >
[5] .<https://en.wikipedia.org/wiki/Big_O_notation> This is the capital letter, not the number zero.

The matrix shape matters in that the overall computational scaling is generally dominated by the largest physical dimension.

Algorithms for noise-filled hyperdimensional vectors scale linearly with vector length: O[N]. Because the “data” here is uncorrelated random noise, one can arbitrarily break the search up into multiple parallel searches by partitioning the vector for a roughly proportionate speedup.

Fast Fourier Transforms scale as O[N*Log(N)] (for one dimension), but cannot be computed efficiently by a large number of parallel processors due to the required internal data flows.

Dense matrix multiplication and inversion scales as O[N^3] if the matrix is square, or O[M*N^2] if rectangular (where M <= N). This does not parallelize all that well. Tensors follow a similar rule with more dimensions, but matrix shape still matters.

There are also exponential O[2^N] and factorial O[N!] scaling cases, but none of the UCs discussed herein involve these, or ever will, as they would be totally impractical.

3 Observations

The general observation is that the computational complexity scaling law depends on the mathematical dimensionality (fewer is better), how square the matrix is (the closer to a vector the better), the noise level (partial order works better with more noise), and the required degree of ordering (partial is required and thus assumed).

The product of communications bandwidth and the square of delay is necessarily roughly constant due to the interaction of the Inverse Square Law with Shannon’s Information Theory.

Relativistic corrections to propagation delay of light is important in PNT GNSS systems and the like, but do not affect the data communicated.

4 Operating Environments

TimeCard instances may be used in systems whose largest physical length is anywhere from a few meters to the diameter of the Solar System, with propagation delays to match. These systems operate at temperatures from the Solar corona (~10^6 Kelvin) to the dark sides of the outer planets (~50 Kelvin). Some such systems are in very noisy industrial estates, with heavy shock and vibration, plus industrial processes like shipbuilding or oil refining. No single hardware design can span such immense ranges of conditions[6].

Valid external time or frequency reference data (such as from GNSS systems) may be unavailable, and external reference use may be forbidden in some applications. The fundamental time reference need not be a named timescale like UTC or TAI.

5 References

None. Footnote references are purely informative and so belong in the P3335 Bibliography.

[6] Mine “TimeCard Use Cases (20240409).pdf ” for specific environments.

6 Notes

Created on 26 February 2025 from TimeCard Use Cases (20240409) for inclusion in the P3335 standard. Updated periodically thereafter.

7 Acronyms

2D = Two Dimensional, **GNSS** = Global navigation satellite system (such as GPS), **I/O** = Input / Output, **Km** = Kilometer, **Km^2** = Square Kilometer, **PNT** = Position, Navigation, and Time, **RF** = Radio Frequency, **TAI** = Universal Atomic Time, **TC** = TimeCard, **UC** = Use Case, **UTC** = Coordinated Universal Time, **WG** = Working Group