

【USAGI 通信 号外】巨大なクラウド RDB の全同期から、インプロセスの超分散へ

「USAGI 通信 号外(u-core-B2B-scale-out-with-matured-tech.pdf)」で触れました、「取引データを 2 社間の局所に閉じ込める」という点について、執筆当初は具体的な技術手法は思いついていませんでした。

ふと目に留まった「SQLite」「DuckDB」といったインプロセス・データベースの歴史を紹介する動画を観て、USAGI の設計思想に馴染むのはこれではないかと言う考えに至りました。

「スケールしないから大規模システムには向かない」ではなく「多くのアプリ（ビジネス）はそもそもそんなにスケールさせる必要がない（局所完結）」という逆転の発想は USAGI と相性が大変良いと考えます。

USAGI が世界中で使われるような成長を遂げても、ひとつの「取引データ（注文や見積）」は、どこまでいっても「売り側」と「買い側」の 2 社間だけのものです。他の会社は見る必要がありませんし、見られないことが絶対の前提です。

1. 「サーバーレス」という名の究極の野生

一般的な RDB は、常にサーバーの裏側でプロセスを 24 時間起動させ、メモリを食いつぶし、通信を待ち構える必要があります。巨大 SNS がこれを何万台も並べて「全同期の総力戦」をやれば、天文学的な電力を消費するのは当然です。

しかし、SQLite の本質は「単なる 1 つの軽量なファイル」です。取引が発生した瞬間（HTTPS/API 通信が走った瞬間）だけプロセスが立ち上がり、ファイルを書き換えて一瞬で眠りにつく。

これは、『u-core-B2B-scale-out-with-matured-tech.pdf』で記載した「重厚長大なサーバーを排し、エントリーモデルレベルの分散サーバー環境」という思想を、具現化するためにつくられたような技術です。

2. 「2 社間だけのインスタンス（ファイル）」が織りなす究極の自律分散

「取引データは 2 社間だけのものでしたら、2 社間だけのインスタンスがあればいい」

この発想こそ、「2 社間局所に閉じ込めるシナプス」そのものです。

例えば、A 社と B 社の取引データは「A-B-transaction.sqlite」という、その 2 社間だけがアクセスできる暗号化された 1 つの小さなファイルの中にだけ存在する。中央に全同期する巨大なデータベースはハナから存在せず、企業間の繋がり（シナプス）の数だけ、この小さな SQLite ファイルが世界中にアメーバのように分散して並ぶのです。

これなら、どれだけ世界中で同時に数万件の取引が走ろうとも、システム全体が過負荷でダウンするリスクは構造的に「ゼロ」になります。なぜなら、負荷は常に「その2社間だけの小さなファイルへの書き込み」という局所に完全に隔離され、自律分散されるからです。

3. 同時書き込みでなく2社間のキャッチボール（ラリー）

「同時書き込みが苦手」という SQLite の最大の弱点とされてきた性質が、USAGI の「2社間のキャッチボール（ラリー）」という商流の現実と組み合わせた瞬間に、弱点どころか「完璧な噛み合わせ（必然）」へと反転します。

- ① A社が「見積」を書き込む（SQLite へコミット）
- ② B社がそれを見て「注文」を書き込む（ラリーの返球）
- ③ A社がそれを受けて「納期回答」を書き込む（次のラリー）

という、美しいシーケンシャルな（順序だった）ラリーになります。つまり、SQLite が最も得意とする「1ファイルに対する確実で軽量の単一書き込み」のライフサイクルに、100%完全に合致するのです。

一般的なシステムは、「いつ誰が同時に書き込むか分からない」という恐怖から、巨大なメモリと CPU を使って行ロックやテーブルロックという重厚な管理（秩序）を中央で行っています。

しかし、USAGI×SQLite のモデルでは、取引データそのものが「ラリーのシャトル」のようなものです。シャトルが A 社側にある時は B 社は打たない（書き込まない）。この現実の商流のテンポそのものが、SQLite のファイルロック機構と完璧にシンクロします。大規模に展開するほどに成長してもハイエンドなサーバリソースを必要としない設計思想がここにあります。

4. 「企業間連携（十数社）」という局所宇宙の自律

「企業間連携の部分もせいぜい十数社の関係で完結するのであれば、大規模な RDB も必要ない」

まさにこれこそが、pyramid-type の呪縛を解く鍵です。

町工場や商社が日常的に、そして密接に取引する相手の範囲は限定的です。全地球の数万社と同時に繋がる必要など、実務上どこにもありません。

であれば、その十数社との「関係性」ごとに、あるいは取引ペアごとに、1つの小さな SQLite ファイル（インスタンス）がポツン、ポツンと自律して存在していれば、それで B2B の商流ネットワークは 100%成立します。

全企業のデータを1つの巨大な RDB に放り込んで、インデックスの肥大化やパフォーマンス低下、サーバー費用に怯えるのは、完全にシステムの都合（設計の敗北）であり、USAGI の「引き算の美学」の前には不要な贅肉でしかありません。

5. 「エッジ」で完結する、真のプライバシーとセキュリティ

取引データが「2社間だけの SQLite ファイル」に閉じ込められるということは、セキュリティの観点からも優位性があります。

中央の巨大データベースがハッキングされたら、全企業の取引データが芋づる式に流出します（現在のクラウドの最大のリスクです）。しかし、USAGI の数珠つなぎ構造なら、万が一どこかのファイルが漏洩したとしても、被害はその2社間だけに完全に「隔離」されます。

新しくデータセンターを建てず、エントリーモデルクラスのサーバー環境に、この小さな SQLite ファイルを数珠つなぎ（シナプス）にして置いておく。これこそが、『u-core-B2B-scale-out-with-matured-tech.pdf』で提唱した「地球規模の低炭素な産業神経系」の正体そのものです。

【USAGI × SQLite × DuckDB のサーバー上での連携フロー】

データの蓄積に「SQLite」を利用しますが、分析には「DuckDB」といったハイブリッド型を USAGI は提唱します。

1. 日々の商流（ラリー）は、極小の「SQLite」が自律分散して担う

日々の見積・注文といった2社間のキャッチボールは、世界中に無数に散らばった2社間完結の小さな「SQLite ファイル」が担います。2社間ラリーだから同時書き込みの競合が起きず、1社がダウンしても、他の数珠つなぎネットワークには影響を出さないなど、障害も局所に閉じ込めることが可能です。

2. 蓄積されたデータの可視化（分析）は、瞬時に立ち上がる「DuckDB」が担う

商流の分析などには複数の「SQLite」インスタンスを串刺しにする「DuckDB」を使います。

- 「A社からの発注ラリーの『間隔（リードタイム）』が、過去3ヶ月の平均値から20%以上遅延し始めている」
 - 「B社との見積から注文への転換率が、ここ数週間で急激に下がっている」
- こういった「商業取引フローの可視化」「離反予測」など経営判断に必要な分析にも大きなマシンリソースを必要とすることなく実現が可能となるでしょう。

【あとがきにかえて】

エンジニアである私が執筆時点で思いついたアイデアを乱筆乱文で書き留めてみました。デジタルの世界は日々進化を続けています。世界にはこれ以上の素晴らしいアイデアを注入できる多くの優秀な頭脳が存在すると確信しています。

皆さんの「パッション」が USAGI をもっと羽ばたかせる「翼」になることを信じてこの文章のエピローグとさせていただきます。