

# 4Q-02 パターンに基づく移動エージェントシステム設計

小松千尋、藤田 悟、山之内 徹

NEC C&C メディア研究所

e-mail: {komatsu,satoru,yamanouchi}@ccm.cl.nec.co.jp

## 1 はじめに

移動エージェントシステム設計に対するデザインパターンの適用について、新しいパターン化手法を提案する。ネットワーク形状、外的リソース環境、及び移動エージェントの振舞いを焦点としたパターン化手法によって、システム運用時の通信性能向上、回線の耐切断性向上等、設計時に移動エージェントの利点を活かすことが可能となる。

本稿ではパターン化手法の簡単な概要を示したのち、複数ホストの検索時におけるデータ消失率に関して、パターンの入れ換えによる運用時効果の比較結果を提示する。

## 2 移動エージェントパターンとカタログ

移動エージェント技術とは、実行中のプロセスを別のコンピュータに移動させ、実行を継続させる技術であり、通信効率の向上、回線切断への耐性、環境適応性、設計容易性等の様々な効果が期待されている。これを受け、筆者らは「ネットワーク形状」と「移動エージェントの移動・通信に関する振舞い」を用いたパターン手法 [1] を提案した。これは、移動エージェントシステムの開発設計に於ける「運用時に於ける移動エージェントの効果」を重要なポイントとし、あらかじめ特徴的な効果をもたらす移動の振舞いをパターン集としてカタログ化しておくことによって、システムに対する移動エージェントの適用に的確な指針を与えるものである。

移動エージェントパターンは、オブジェクト指向におけるデザインパターンの記述方法 [2] に習い、名称、構造、適用可能性など 12 の記述項目を流用するが、「構造」には新たに定義する「トポロジー図」「通信シーケンス図」を使用する点、「適用可能性」はオブジェクト再利用性よりもシステム運用時の効果を重視する点、さらに「評価」という項目を追加し、通信性能等に関する評価式やその考察などを記述する点が異なっている。詳細なパターン記述法及び、既に収集した 13 のパターンの定義は文献 [1] に記す。

図 1 は、複数箇所のリソースに巡回アクセスする「Round」パターンのトポロジー図、通信シーケンス図を示している。トポロジー図 (図 1 左) は、対象システムのネットワーク形状、及び移動・通信を行なうエージェント、場所固定のリソースとそれらの間の移動、通信の行程を示している。通信シーケンス図 (図 1 右) は、ローカル通信、リモート通信、移動の 3 アクションを用いて、移動によるローカル/リモート通信の関係の変化を表している。

このパターン記述によって、移動エージェントによる分散システムのパラダイム変化を簡単に表現することができる。開発

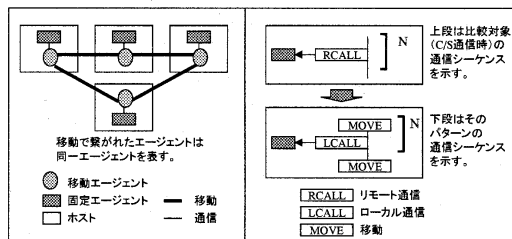


図 1: トポロジー図と通信シーケンス図

者は各々のパターンが持つ長所、短所を把握し、また「評価」の項から、本手法の重視する「運用時の利点」(通信性能等)の定量的評価を得ることができる。

## 3 運用時効果の比較実験 (耐ホスト障害性)

文献 [1] では通信効率の比較実験を主に行なったが、本稿では、ホスト障害への耐性効果に関してパターン毎の比較評価を行なう。

システムの構成例の概要を図 2 に示す。クライアントホストから、m 台のホストにある DB にアクセスし、各々の DB からデータを取得する。ここで各々のホストは、指数分布し従ったような確率で障害を起こし、使用不能状態に陥る (ダウンする) という環境を想定する。ホストがダウンしている間はホストに対する通信/エージェント移動は全て不可となり、また、ダウンする瞬間にそのホストに滞在していたエージェントは消滅する (そのエージェントが持っていた検索済データも同時に消滅する)。

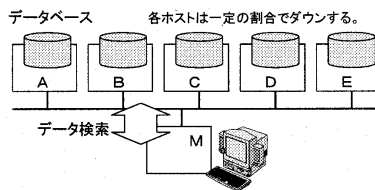


図 2: 対象システム概要図

この状況において、各ホストの DB へのアクセスを行なうパターンとして、(1)RemoteCall (C/S 通信)、(2)Round (巡回移動) の 2 つのパターン (図 3) を取り上げ、各々のパターンを使用した際に得ることの出来るデータの期待値について比較を行なう。

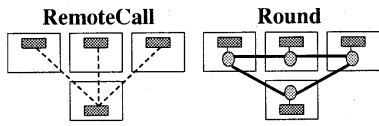


図 3: RemoteCall パターンと Round パターン

計算に使用するパラメータを、アクセスするホスト数 ( $m$ )、単位データサイズ ( $W$ )、1ホスト当たりの獲得データ数 ( $n$ )、ローカルコール1回に掛かる時間 ( $d$ )、リモートコール1回に掛かる時間 ( $r$ ) のように定義する。ホストの故障率を一概に  $\lambda$  とした場合、ある時間  $t$  に於いてホストが落ちない確率は  $R(t) = e^{-\lambda t}$  と定義できる。またダウン時における平均復旧時間を  $\tau$  とすると、平均稼働時間が  $\rho = \frac{1}{\lambda}$  であることから、ホストの稼働率  $A = \frac{\rho}{\rho + \tau}$  が定義できる。これらのパラメータから RemoteCall、Round 双方の獲得データ量の期待値  $S_{Remote}(m)$ 、 $S_{Round}(m)$  を算出する。

RemoteCall は移動を行わず、各々のホストにリモート通信を行なう。稼働しているホストに対しては、 $n$  回の通信を終えるか、又はホストがダウンするまで通信を継続する。最初のアクセス時に既に稼働していないホストはスキップする。通信がリモートであるため1回当たりの通信時間 ( $r$ ) は大きい。このとき、 $S_{Remote}(m)$  は以下ようになる。

$$S_{Remote}(m) = A \sum_{i=1}^n WR(r_i)m = A \sum_{i=1}^n W e^{-\lambda r_i m}$$

一方 Round の場合、エージェントがホストに移動して巡回検索を行なう。エージェントは稼働しているホストに移動するが、そのホストでの  $n$  回のアクセスが完全に終了する前にホストがダウンした場合、これまでに検索したデータを全て失ってしまう。ただし高速なローカル通信の為、アクセス時間 ( $d$ ) は短い。このとき、 $S_{Round}(m)$  は以下ようになる。

$$\begin{aligned} S_{Round}(m) &= \sum_{i=0}^m (m-i) \binom{m}{m-i} (1-A)^i A^{m-i} R((m-i)dn) W n \\ &= \sum_{i=0}^m (m-i) \binom{m}{m-i} (1-A)^i A^{m-i} e^{-\lambda(m-i)dn} W n \\ &= A e^{-\lambda dn} (A + (1-A)e^{\lambda dn})^{m-1} m W n \end{aligned}$$

パラメータ  $A = 0.90$ 、 $W = 10$ 、 $n = 10$ 、 $r = 10$ 、 $d = 1$ 、 $\lambda = 0.01$  及び  $0.03$  として、ホスト台数  $m$  を変化した場合の  $S_{Remote}(m)$ 、 $S_{Round}(m)$  の変化をグラフにしたものが図 4 である。RemoteCall は台数に比例してデータを獲得できるのに対し、Round の場合、移動先でホストダウンに遭遇すると全てのデータを失う為、ホスト数が増加する毎に消滅率 (獲得データが 0 になる率) を高めてしまうため、期待値の増加率が減少していき、ある時点で期待値が RemoteCall と逆転する。

前記  $S_{Remote}(m)$ 、 $S_{Round}(m)$  について、 $e^x$  を  $x = 0$  の近傍で近似すると、変形後の式は以下ようになる。

$$S_{Remote}(m) = Amn(1 - \frac{r(n+1)}{2})W$$

$$S_{Round}(m) = A^m(1 - m\lambda dn)m n W$$

$$\frac{S_{Remote}(m)}{S_{Round}(m)} = \frac{1}{2A^{m-1}} \frac{2 - (n+1)\lambda r}{1 - m\lambda dn}$$

ここで  $A$  をほぼ 1 とすれば、

$$= \frac{2 - r\lambda(n+1)}{2 - 2m\lambda n}$$

となり、これは  $m_{cross} = \frac{r}{2d}$  の時ほぼ 1 となる、すなわち、RemoteCall と Round の獲得データ量に関するトレードオフは、リモートアクセス時間とローカルアクセス時間の比及び検索ホスト台数で見積もることが出来る事が分かる。先の計算例は  $\frac{r}{d} = 10$  であるため、 $m_{cross} = 5$  となり、これを越えるポイントで両者の得失が逆転する。同様に  $\frac{r}{d} = 100$  であれば  $m = 50$  が目安となる。この値は  $\lambda$  が十分小さい場合、 $\lambda$  の大小にほとんど影響されない。

ここでは耐障害性、特にホストの障害に関して議論したが、このように様々なパターンとその適用指針を提示することにより、システム開発者は、対象システムに合う設計パターンをよりの確に判断することができる。

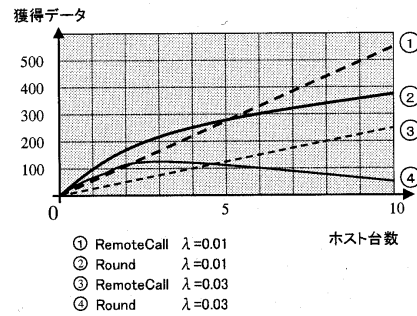


図 4: ホスト数に対する獲得データ期待値

## 4 おわりに

本稿では、移動エージェントのデザインパターンに、運用時に於ける移動エージェントの利点を活かす為の新しいデザインパターンの記述手法を提案した。実際に耐障害性 (データ消失率) の観点から各パターンの評価を提示し、設計時に採用すべきパターンは、システム要件や環境条件を十分考慮した上で適切なものを選択すべきであることを示した。

移動エージェントの振舞いをパターン化しその利点、欠点をカタログ化することは、システム設計者にとって大きな意義がある。今後はパターンの洗練と追加及び、設計レベルから実装レベルに落とす技術についても研究を行なう予定である。

## 参考文献

- [1] 小松千尋, 藤田悟, 山之内徹: パターンに基づく移動エージェントシステムの設計手法: 電子情報通信学会 AI99-49, 1999.
- [2] E.Gamma, R. Helm, E.Johnson, J.Vlissides: Design Patterns: Addison-Wesley, 1995