

網環境における

3G-3

プロセスのCapability 管理方式

和賀井 フミ子 田中 英彦 元岡 達

(東京 大学 工学部)

1 はじめに

これまでに、capability管理方式は、幾つかの提案や実装が行なわれてきているが、網環境においては、あまり考慮が払われてこなかった。

ここでは、分散処理向きのプロセスのcapability管理について考察する。

始めに、capabilityを網に拡げた時の問題点を調べ、その解決法を探る。次に、網構成には、不可欠なプロセス間通信にcapabilityを導入した場合の基本的構成について考察する。

2 capabilityについて

一般的には、capabilityは表形式で表わし、meta-operation を用いて操作することが多い。

表形式の作り方には、主に、C-list方式とA-list方式がある。両者の機能は同値であることが知られている。

C-listは、或る対象が使用し得る相手側のcapabilityを書いた表である。相手を直接調べなくとも、使用の可否がわかるので、処理効率が良い。

A-listは、或る対象が受け入れ可能な相手のcapabilityを書いた表である。受け入れ側にあるので、常に、正確な情報を保つことができる。

3 網環境におけるcapability

網環境においては、誤りに対する処理は、集中的環境における場合よりも難しくなるので、網環境にcapabilityを導入することは、効果がある、と思われる。

この場合、第一に考慮しなければならないことは、capabilityの変更・更新が集中的環境ほど容易ではない、という点である。例えば、変更情報を知らせる相手は変化し得る。又、伝達遅延による情報の食い違いも起こり得る。これは、網の性質上 止むを得ないことなので、その対策を講ずる必要がある。

伝達遅延の問題を避ける為には、capabilityは、アクセスされる対象のみが持ち、コピーを相手に渡さない様にすればよい。こうすれば、capability管理は集中に行なわれる。ところが、他のノードにお

いので、このA-listを用いるだけでは、着いた場所ではねつれられることがあり、効率が悪い。そこで、事前にアクセスの可否がわかる方法があるとよい。特に、網に出る前に調べる機能があると便利である。その為には、網上で不正操作を予防することも考え合わせて、各ノードに他のノードの相手に対するC-listも置く必要がある。

C-listを用いると、コピーを持つことになるので、伝達遅延によるcapabilityの不一致が起こり得るが、この場合だけは、効率は落ちて、A-listにより対処できる。又、この頻度はそれ程多くはないと考えられる。

換言すると、伝達遅延による不一致は生ずることがあっても、それがただちに誤りになることは無いような構成が可能である。即ち、複雑なcommitment control を用いなくとも、C-listとA-listとを併用することで、効率良く、誤りも生じにくいシステムが構築できる、と考えられる。

C-listの更新は、主にA-listが変わった時に行なわれる。その制御は、網環境を考慮して、二重に行ない、誤りに対処できるようにする。

A-listは情報内容の変更とともに、変更中の旨の表示を行ない、対応ノードのすべてのC-listに通知する。各C-listは、内容変更が済んだら、A-listにその旨を知らせる。A-list側では、該当するすべてのC-listから変更済の通知が戻ったら、変更済とする。

C-listの変更が途中で失敗した場合は、capability管理者にその旨を通知し、原因調査及び回復処理を依頼する。

capabilityの変更中に、情報不一致の為、実際にはアクセスが可能であるのに、できないことも起こり得る(過保護問題)が、この点は許容することにし、むしろ制御の容易さの方を重視する。

C-list変更処理は、ノード対応に、C-list管理プロセスをおき、これらの間で通信を行うことで、実現する。

4 capabilityを付加したプロセス間通信

プロセス間通信は網においては必須要素である。それだけに誤操作は許されない。誤り予防の一方法として、プロセスに対し、通信に対するcapabilityを付加することを考える。ここで対象とするプロセス間通信機構は、これまでに CUPIDシステムに実装された方式を基にする。

プロセス間通信は、通信するプロセスが互いに通信要求を出し、通信契約を交わすことにより行なわれる。

通信要求を出しても、相手が存在しなかったり、通信不能な相手の場合は、通信契約が成立せず、途中で止まったままになる。これまでは、Time Outによって解決してきたが、capabilityを導入することにより、通信要求を出した時点でチェックが可能になるので処理効率が向上する。

capabilityのチェックには、網環境であることを考慮して、C-listと A-listを併用する方式を採る。

プロセス間通信では、相手プロセスの指定のしかたも重要である。架空の相手やまちがった相手を指定することの無いように、各プロセスは、使用する相手プロセスを予め登録し、この中から使うようにする。この管理も C-listを用いて行なうことができる。

C-listと A-listとを用いたプロセス間通信の手順は以下の様になる。

プロセスが通信要求を出すと、C-listにより通信のcapabilityを調べ、可ならば、プロセスのローカル・ネームをシステムの標準名に変換し、網に出す。これを受けた側では、A-listにより、capabilityを確認して、可ならば、標準名からプロセスのローカル・ネームに変換して、該当プロセスに通信要求を渡す。

5 今後の方針

以上、考察してきたcapability付きプロセス間通信の機構の詳細設計を行ない、その後、既に実装済みの分散処理向き専用サブシステム CUPIDに組み込むことを検討する。

参考文献

- 1) Peter J. Denning : Fault Tolerant Operating Systems
Computing Surveys Vol.8 No.4 Dec.1976
- 2) W. Wulf : The Hydra Operating System
Jun.1975
- 3) 和賀井、田中、元岡 : 網向きプロセス間通信制御プロセッサCUPIDの性能測定
情報処理分散処理システム研究会16-5

