オペーレーティング・システム 54－7
（１９９２．３．１３）

計算機システム性能評価用シミュレータ
― ＩＮＳＹＤＥ ―

小松俊雄　　野瀬純郎
ＮＴＴ情報通信網研究所

大規模で複雑な計算機システムの性能評価に，シミュレーションは有効な手法であるが，専用言語やモデリング技法の習得，膨大な所要工数などが障害となって，誰でも手軽に行えるという状況にはない．これらの障害を取り除くため，計算機システムの性能評価用シミュレータ：ＩＮＳＹＤＥを開発した．ＩＮＳＹＤＥはモデル条件を図表形式で入力すると，自動的にシミュレーションプログラムを作成・実行し，結果を図表形式で出力するものである．これにより，シミュレーションの経験がなくても，容易に性能評価を行うことが可能となった．本稿では，ＩＮＳＹＤＥの特徴，モデリングの方法，適用例などについて述べる．

A Simulator for the Performance Evaluation of Computer Systems
― INSYDE ―

Toshio Komatsu　　Junro Nose
NTT Network Information Systems Laboratories

Simulations are one effective method for the performance evaluation of computer system, but the large number of skilled specialists required with knowledge of simulation languages and modeling techniques is a major obstacle. The way to get around this is to make it so that anyone can conductsimulations easily, and hence the INSYDE for computer system performance evaluation was developed. The modeling conditions are input in both table and graphic form, and INSYDE automatically creates and runs a simulation program, outputting the results also in both table and graphic form. This paper describes INSYDE's characteristics and modeling techniques, and presents a sample application.

1．まえがき

計算機システムが社会に広く浸透するにつれて，システムはますます高度化，複雑化，大規模化が進み，それに伴って性能への要求条件が厳しくなってきている．このため，システムの性能・負荷・リソース間のバランス分析をもとに，ボトルネックの抽出，適正なリソース量の算出，処理能力限界の推定，業務量増加に伴う性能劣化の予測などを的確に，かつタイムリに行える手段が求められている．

性能予測の手法には，待ち行列モデルによる解析とシミュレーションがよく知られているが，大規模で複雑なシステムを高い精度で評価するには，シミュレーションが適している．しかし，シミュレーションを実行するには，一般に次のような問題がある．まず，ＧＰＳＳ，ＳＬＡＭ，ＳＩＭＳＣＲＩＰＴなどのミュレーション言語の修得が必要である．次に，モデルの検証，シミュレーション実行結果の分析にかなりの工数を必要とする．モデルは進化・発展していくのが常であり，これに対応したモデルの修正が頻繁に生じ，そのための作業量も膨大になる．また，システムの開発者／運用者と評価者が異なる場合，モデルの作成，確認のための情報交換に長い期間を要する．これは，前者のシミュレーションに関する知識が不足していることによる．

そこで，シミュレーション専用言語からの解放と評価工数の削減を目的として，計算機システムの性能評価用シミュレータＩＮＳＹＤＥ（Integrated Support Tool for System Design and Evaluation）を開発した．

ＩＮＳＹＤＥは，リソースの構成，処理の流れ，負荷の発生条件などのモデル定義情報を図表形式で入力すると，シミュレーションプログラムを自動的に作成・実行し，結果を図表形式で出力するもので，シミュレーションの経験がなくても，手軽にかつ短期間に性能評価が行える．このような評価用ツールは通信ネットワークの分野でいくつかあるが[1]-[4]，大規模なシステムへの適用は規模および精度の点で困難である．また，計算機システムを対象としたものは見当たらない．本稿では，ＩＮＳＹＤＥの特徴，モデリングの方法，および適用例などについて述べる．

2．ＩＮＳＹＤＥの特徴

シミュレーション専用言語からの解放，および評価工数の大幅な削減を図るために開発したＩＮＳＹＤＥは，以下のような特徴をもつ．これらは，筆者らが多くの情報通信システムの構築にあたり，実際に性能評価を実施してきた経験[5]-[11]を反映したものである．

表１　INSYDEの対象モデルの規模

項　目	最大数
装置台数	10,000　台
タスク数	200　種
ファイル数	5,000　個
呼種数	200　種

①モデル定義は図表で記述する．これはシステム開発者／運用者とのレビューにそのまま使用でき，新たに別の情報交換用ドキュメントを用意する必要はない．

②システムを構成するリソースはハードウェアのみならず，タスク，テーブル，ファイル等のソフトウェアも対象とし，その数が数千の大規模なシステムにも対応可能である．（表１）

③割り込み，排他制御等の計算機に特有の複雑な処理も扱えるよう，豊富な記述法を用意している．

④さらに，図表形式では表現が困難／煩雑な条件は，ユーザがＦＯＲＴＲＡＮでオウンコーディング

することが可能である.

⑤プログラムの検証を容易にするための，エラー検出機能およびトレース機能が充実している.

⑥シミュレーションの実行結果は分かりやすく図表で表示する.

⑦シミュレーションの実行時間が，専用言語で直接プログラムを作成した場合に比較して，同程度である.

3. INSYDEでのモデリング

INSYDEでは，システムの動作を，トランザクションがいくつかのノードと，それらを結ぶアークで構成されるネットワーク上を動くモデルとして表現する．モデリングは次の3種の定義により行う.

①リソースの定義

②処理ﾌﾛｰ （ネットワーク）の定義

③トランザクションの定義

記述性と柔軟性を考慮して，①と③は表形式で，②は上から下へ記述するフローチャート形式で定義することとした.

3.1 リソースの定義

ハードウェアおよびソフトウェアのリソースについて，表2に示すような項目の定義を行う.

3.2 処理フローの定義

(1) INSYDEのノード

INSYDEがサポートする52種類のノードを表3に示す．特徴的な

表2　リソースの主な定義項目

種　類		定義項目	内　容
ハードウェアリソース		装置名称	
		装置台数	同時に処理可能な数
		多重度	サービス開始条件
		起動契機	有限の場合，溢れは呼損扱い
		待ち容量	待ち室のﾄﾗﾝｻﾞｸｼｮﾝを取り出す規則
		待ち選択規則	処理速度／固定時間
		性能	
ソフトウェアリソース	タスク	タスク名称	
		タスク枚数	タスク種別毎
		起動契機	サービス開始条件
		CPU 補足優先度	ﾀｽｸ 間の CPU優先順位
		待ち室名称	
		待ち容量	
		待ち選択規則	待ち室のﾄﾗﾝｻﾞｸｼｮﾝを取り出す規則
		窓口数	同時に処理可能なタスク枚数
		ﾀｽｸ 補足優先度	タスクを確保するための優先順位
	テーブル	テーブル名称	
		多重度	ﾛｯｸ ，ｹﾞｰﾄ 制御等の対象ﾃｰﾌﾞﾙ名
	ﾌｧｲﾙ	ファイル名称	

表3　INSYDEのノード

分　類		種類	機　能
時間遅延ノード		8	リソースを確保した状態，あるいは確保しない状態でトランザクションを遅延させる
制御ノード	開始／終端	8	システム，タスク，マクロ処理フローの開始と終端を表す．終端においてはトランザクションの消滅以外に，処理の再開，他処理へのキューイングが可能である.
	分岐	13	無条件分岐と種々の条件分岐ができる。条件分岐には，等確率で分岐，指定した確率で分岐，指定の通過回数に達したときに分岐，またテーブルやワークエリアに設定した条件による分岐がある（図1）.
	処理依頼	6	システム処理とタスク処理間およびタスク処理相互間の処理依頼ができる（図2）.
	ﾘｿｰｽ管理	9	リソースの確保／解放を行う．複数のリソースを同時に確保／解放する事も可能である．また，タスク処理において CPUへの強制割り込みができる.
	分裂／統合	4	トランザクションを複数個に分裂させて，同一方向／複数方向に進める．分裂したトランザクションは全て統合／一部統合の指定が可能である.
ユーザノード		1	ユーザが独自にノードを定義する.
マクロノード		1	予め定義したマクロ処理フローを呼び出す
測定ノード		2	任意の測定区間の統計情報を取得する.

ノードの使用例を図1，図2に示す．

(2) フローの種類

タスクの処理をモデル化する場合，処理の流れが複雑になるため，次の2段階に分けて定義する．

①タスク間の情報および制御の流れを定義したシステム処理フロー

②個々のタスク内の情報および制御の流れを定義したタスク処理フロー

(3) ノードパラメータの値設定

記述をシンプルにするため，リソース番号や処理量等のパラメータは，直接ノード上に値を設定する以外に，表形式の設定（図3）と FORTRAN記述が可能である．

3.3 トランザクションの管理

トランザクションを管理する番号として，発生番号以外にトランザクションの分裂，コピーが区別できるように世代番号，識別番号を自動的に付与する（図4）．これにより，トレースが容易となる．

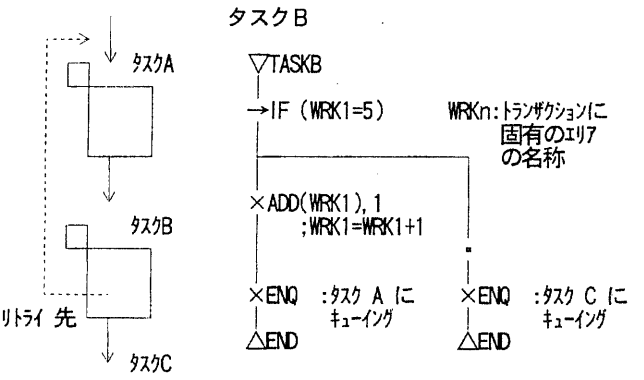


図1 分岐ノードの使用例

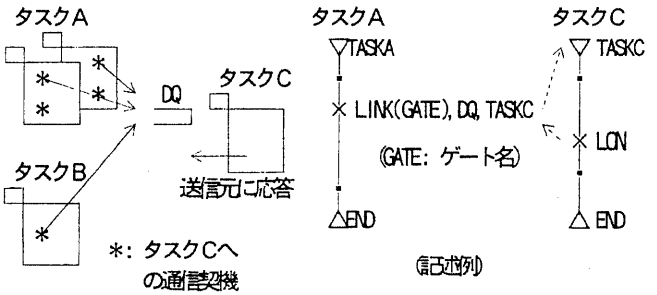


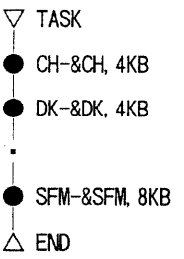
図2 タスク間通信の記述例

4．シミュレーションプログラムへの展開方法

高性能化と開発工数短縮のために，以下の方法でモデル条件をシミュレーションプログラムに展開する事とした．

①シミュレーション言語は広く実績のあるＳＬＡＭII [12] を採用する．

②種々の事象処理ルーチン（システムの状態に何らかの変化をもたらす出来事に対応した処理）を作成し，それらの組合せによってシミ



・&XX は装置番号を表示

確率対応表

&CH	&DK	PROB	⇨	&SFM
1	2	50		3
1	2	50		4
2	1	100		2
3	3	100		1

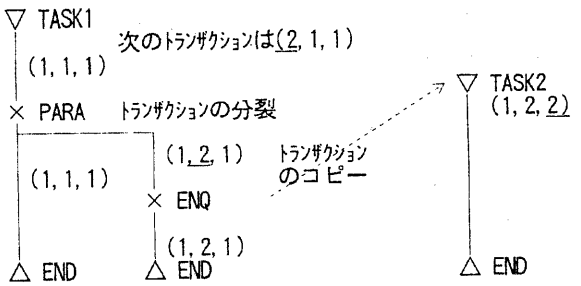
・PROBは確率値
・&CH=1, &DK=2であれば等確率で&SFM=3 or 4

図3 確率対応表によるパラメータ値の定義

ュレーションプログラムに展開する.

③高性能化のため，事象処理ルーチンはＳＬＡＭIIのベース言語であるＦＯＲＴＲＡＮで記述する.

④事象処理ロジック(事象を早いもの順に実行するなど，シミュレーション全体を司る機能)，トレース等はＳＬＡＭIIの機能を利用する.



(発生番号，世代番号，識別番号)

図4　トランザクション番号の付与方法

5．適用例

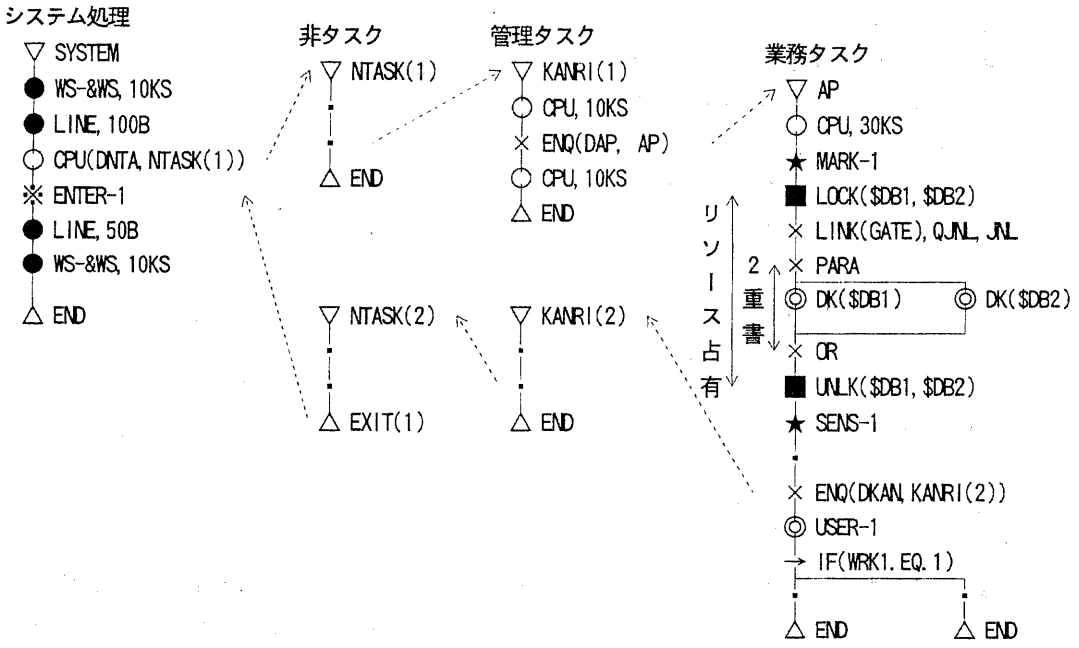
回線系設備の設計情報，故障処理情報等をデータベース化し，サービスオーダから完成までの業務を管理する大規模設備管理システムを評価対象として，ＩＮＳＹＤＥを適用した際の，処理フローの一部とユーザの FORTRAN記述例を図5に示す．また，シミュレーション結果の表示例を図6に示す.

図には複数の業務のうちの，1業務の処理の流れが示されている．ＷＳで発生したトランザクションが回線を介してセンタに到着すると，非タスクで受信処理を行い，管理タスクに処理を依頼する．管理タスクはトランザクションを振り分けて業務タスクに処理を依頼する．業務タスクでは，設計情報を更新するために，ファイル(DB1，DB2)を同時にロックし，ジャーナルタスクにジャーナル処理を依頼し，ジャーナルタスクからの応答を待ってファイルの 2重書きを行う．次に，ロックを解除する等の処理を行い，管理タスクに終了処理を依頼する．管理タスクでは非タスクに送信処理を依頼し，情報を送信元のＷＳに送る.

図から分かるように，処理フローは簡単に記述でき，処理の流れが容易に理解できる.

6．あとがき

ＩＮＳＹＤＥの特徴，モデリングの方法，適用例を示した．ＩＮＳＹＤＥの使用により，ＳＬＡＭIIで直接プログラミングした場合と比較して，評価工数を約1/5 に短縮することができた．また，シミュレーションプログラムを事象処理ルーチン（ＦＯＲＴＲＡＮで記述）の組合せに展開する事により，ＳＬＡＭIIで記述した場合に比較して，シミュレーションの実行時間を0.3 ～1.2 倍にすることができた.



（ノードUSER-1の FORTRAN記述）

内容：自タスクで待っているﾄﾗﾝｻﾞｸｼｮﾝ数（<50個）を見て後続の処理の流れを制御する

```
      SUBROUTINE  USREVT(LCODE, CFLOW, IDENT, IFLG)
#INCLUDE 'USCOM. INC'          宣言文の展開
      CNAME='DAP'              業務タスクの待ち室名
      LNUM=1                   待ち室の番号=1
      IF (RWWAVE(CNAME, LNUM). LT. 50) THEN      RWWAVEはINSYDEが提供する関数
                                                 指定の待ち室の平均待ち数を表示
          WRK1=1               ワークエリアに1 を設定
          GO TO 200
      ELSE
          WRK1=2
  200 IFLG=1                   IFLGはﾄﾗﾝｻﾞｸｼｮﾝの消滅フラグ
      RETURN
```

図5　大規模設備管理システムへの適用例

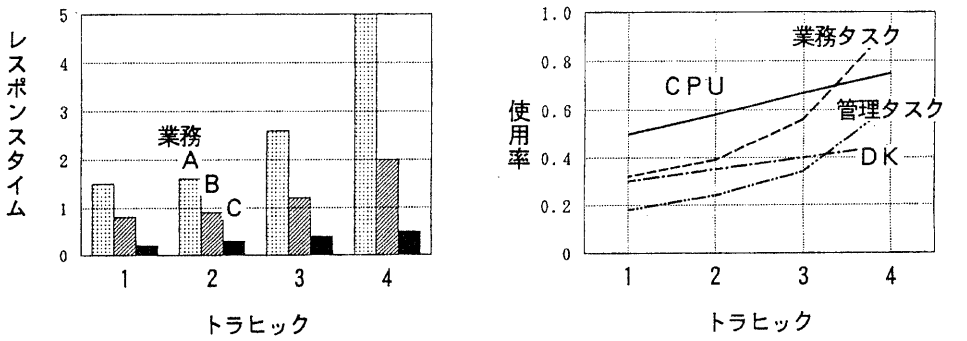


図6　シミュレーション結果の表示例

〔参考文献〕

(1) K. BHARATH-KUMAR, P. KERMANI: Performance Evaluation Tool (PET): An Analysis Tool for Computer Communication Networks, IEEE JSAC, VOL. SAC-2, NO. 1, Jan., 1984.

(2) M. A. MARSAN, G. BALBO, G. BRUNO, F. NERI: TOPNET A Tool for the Visual Simulation of Communication Networks, IEEE JSAC, VOL. 8, NO. 9, Dec., 1990.

(3)住田, 小沢, 稲守: シミュレーション支援システムTEDAS-S, NTT R&D, Vol. 38, NO. 12, 1989.

(4) A. DUPUY, J. SCHWARTZ, Y. YEMINI, D. BACON: NEST A Network Simulation and Prtotyping Testbed, COMMUNICATION OF THE ACM, Vol. 33, No. 10, Oct., 1990.

(5)小松, 野瀬：シミュレーション手法による大規模コンピュータシステムの性能評価, 電子情報通信学会システムのモデリングと性能評価第二種研究会, 1989.

(6)小松, 野瀬：メッセージ交換型システムのスループット評価, 情報処理学会オペレーティング・システム研究会, 46-2, 1990.

(7)小松, 野瀬：バンキングシステムにおける中継コンピュータの性能評価例, 情報処理学会第41回全国大会, 6D-7, 1990.

(8)小松, 永塚, 野瀬：バースト型負荷に対するシステム評価, 情報処理学会第42回全国大会, 6K-7, 1991.

(9)小松, 杉森, 野瀬：ネットワークサービスシステムの性能評価例, 電子情報通信学会春期全国大会, D-158, 1991.

(10)小松, 野瀬：ネットワークサービスシステムのレスポンスタイム評価, 情報処理学会オペレーティング・システム研究会, 50-9, 1991.

(11)野瀬：シミュレーション技術の情報通信システムへの適用, 計測と制御, Vol. 30, NO. 2, 1991.

(12)森戸, 相沢：SLAMIIによるシステム・シミュレーション入門, 構造計画研究所, 1986.