

Transiente Analyse mit HIT

25. Januar 1995
Armin M. Warda

1 Einleitung

Das Tool HASE erlaubt eine simulative Verfügbarkeitsanalyse von Rechensystemen. Es berechnet die Antwortzeitverteilung eines Dienstes, in die neben Last- und Leistungskriterien auch (Teil-) Ausfälle der Maschine eingehen.

HASE erstellt für jede einzelne der verschiedenen Konfiguration der Maschine, die sich durch (Teil-) Ausfälle ergeben, ein separates HIT-Modell, welches simulativ ausgewertet wird. In diesen Modellen werden also nur die stationären Phasen des Modellverhaltens zwischen den Ausfall- und Reparaturereignissen betrachtet.

Nun besteht natürlich ein Interesse daran, ebenfalls die transienten Phasen des Modellverhaltens zu untersuchen, die bisher vernachlässigt wurden. Dadurch soll die HASE zu Grunde liegende Annahme geprüft werden, daß transiente Phasen wenig Einfluß auf die letztendlich resultierende Antwortzeitverteilung haben und vernachlässigt werden dürfen.

2 Transiente Analyse

Das System durchläuft abwechselnd *stationäre* und *transiente* Phasen.

Stationäre Phasen sind dadurch gekennzeichnet, daß sich das System relativ *stabil* verhält, also die Zustandswahrscheinlichkeiten während dieser Phase im wesentlichen konstant und insbesondere unabhängig von der Zeit sind. Stationären Phasen ermöglichen deshalb die Untersuchung des Systemverhaltens durch Zeitreihenanalyse: aus einem einzigen Simulationslauf wird eine Stichprobe $S = \{s(t_j) | j = 1, 2, \dots\}$ des Systemzustands zu bestimmten Zeitpunkten gewonnen, aus der dann letztendlich Aussagen über das Modellverhalten, wie beispielsweise die Verteilung der Zustandswahrscheinlichkeiten oder Mittelwerte, die für die gesamte Dauer der stationären Phase gültig sind, ermittelt werden.

Im Gegensatz dazu ist eine *transiente Phase* dadurch gekennzeichnet, daß sich das Modellverhalten während dieser Phase grundsätzlich ändert. Insbesondere sind die Zustandswahrscheinlichkeiten während dieser Phase von der Zeit abhängig. Insofern hätten über die Dauer der transienten Phase gemittelte oder summierte Größen wenig Aussagekraft für eine Beschreibung des Systemverhaltens während dieser Phase. In der transienten Analyse werden aus replizierten Läufen Stichproben $S(t_j) = \{s_i(t_j) | i = 1, 2, \dots\}$ für die einzelnen Zeitpunkte t_j gewonnenen, aus denen dann beispielsweise die Erwartungswerte $E[s(t_j)]$ für bestimmte Zeitpunkte innerhalb der transienten Phase geschätzt werden. Diese Form der Analyse wird auch *Ensemble-Analyse* oder *Analyse mit festem Horizont* genannt.

3 Allgemeine Konzepte

Während bei der Zeitreihenanalyse der Initialzustand (bei „vernünftigen“ Modellen) im allgemeinen unwesentlich für das Erreichen der stationären Phase ist (man läßt einfach die transiente Phase des „Einschwingens“ bei der Aufzeichnung von Meßwerten weg), spielt er bei der transienten Analyse natürlich

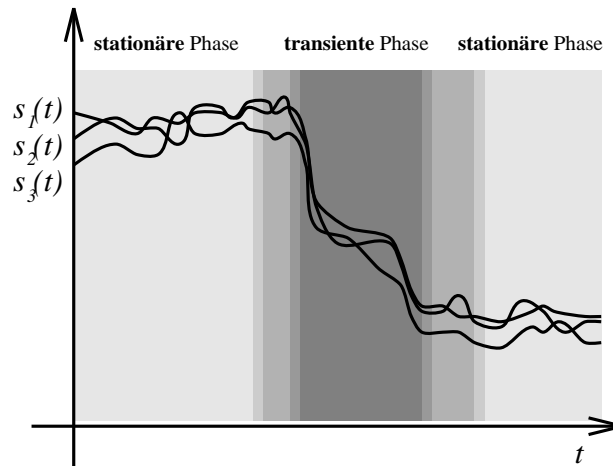


Abbildung 1: Mögliche Zustandstrajektorien $s_i(t)$ des Systems bzw. des Simulationsmodells.

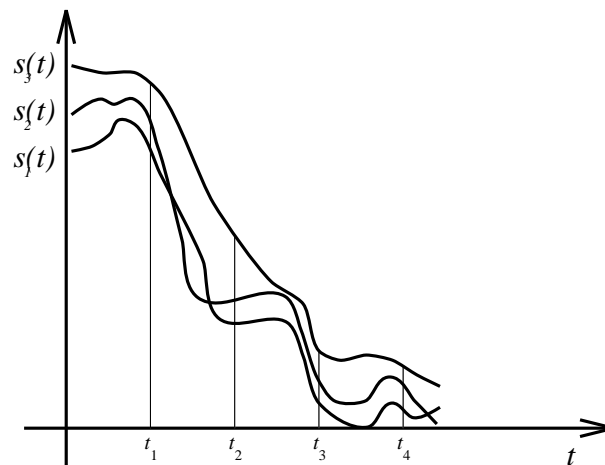


Abbildung 2: Die Grundidee der transienten Analyse: der Zustand in transienten Phasen wird an diskreten Zeitpunkten abgetastet. Aus replizierten Läufen werden Stichproben $S(t_j) = \{s_i(t_j) | i = 1, 2, \dots\}$ für die einzelnen Zeitpunkte t_j gewonnenen, aus denen dann beispielsweise die Erwartungswerte $E[s(t_j)]$ geschätzt werden.

eine entscheidende Rolle. Wichtig sind deshalb Konzepte für die Spezifikation geeigneter Initialzustände und das Einbringen des Modells in einen solchen Zustand.

Der Modelltyp für das transient zu analysierende Modell muß geeignet parametrisiert sein, damit die einzelnen Instanzen des Modells für die replizierten Läufe in die entsprechenden Anfangszustände gebracht werden können.

Sinnvoll erscheint es hier, die Parameterbelegungen in einer Datei zu halten, die pro Experiment eine Zeile mit den Parameterwerten für die formalen Parameter des Modelltyps enthält.

Hat der Modelltyp etwa folgende formale Parameter:

```

random_seed : INTEGER DEFAULT 13;
number_of_nodes : INTEGER DEFAULT 4;
mode_of_operation : STRING DEFAULT 'normal';
mean_disk_seek_time : REAL DEFAULT 32e-6;
initial_population : INTEGER DEFAULT 0;
pending_requests : INTEGER DEFAULT 0;

```

so könnte eine Parameter-Datei folgendermaßen aussehen:

```
# Parameter für Modell TEST # 21.10.1994
Armin M. Warda
  17  4  normal  32e-6  100  17
  13  3  normal  32e-6  105  16  # Dies
  43  2  normal  32e-6   97  13
ist ein Kommentar!
```

Diese Datei würde dann für drei verschiedene Läufe für die Analyse einer bestimmten transienten Phase verwendet.

Für die Generierung der Parameter-Datei bestehen alternativ zwei Möglichkeiten:

manuelle Spezifikation: der Modellierer möchte das Modell selbst in geeignete Initialzustände bringen. Dies scheint bei HASE die geeignete Methode zu sein, um transiente Phasen beim Wiederanlaufen nach Total-Ausfällen zu untersuchen.

automatische Generierung aus Simulation der stationären Phasen: während der Simulation einer stationären Phase werden „typische“ Zustände notiert. Diese Methode ist für die Analyse von Übergängen zwischen zwei stationären Phasen geeignet. Die Maschinenkonfiguration, die dem Modell im Nachfolgezustand entspricht, wird mit der Last-Situation des Vorgängerzustands konfrontiert.

Es sind also Maschinenzustände von Lastzuständen zu unterscheiden. Maschinenzustände sind bspw. die Anzahl von (nicht-) ausgefallenen CPUs, etc. also die Zustände, die durch Ausfall- und Reparaturprozesse verändert werden. Lastzustände sind im wesentlichen durch die Anzahl von noch zu bearbeitenden Requests, also die Warteschlangenpopulationen, gegeben.

4 Voraussetzungen

Es wird die HIT Version 3.3.205 (oder höher) benötigt, Version 3.3.000 hat einen Fehler im Zusammenhang mit Start- / Stop-Bedingungen für Meß-Ströme.

5 Alternative A

Zunächst wurde angedacht, mehrere replizierte Läufe der gleichen Modellauswertung synchron laufen zu lassen, indem das Modell repliziert wird und dieses vervielfältigte Modell in ein Rahmen-Modell integriert wird, vgl. Abbildung 5.

Dadurch sollte es möglich sein, die Auswertung durch HIT besorgen zu lassen.

Dieser Ansatz erwies sich hauptsächlich deshalb als nicht praktikabel, weil eine große Menge von Modifikationen in der Modellbeschreibung notwendig wären. Ein Auswertungsobjekt in HIT, das an einer Array-Komponente definiert wird, ist nämlich immer an einen bestimmten Array-Index gebunden. Will man alle Elemente des Arrays gemeinsam messen, so muß eine neue, künstliche Komponente zur Messung definiert werden, die von der Array-Komponente „enclosed“ wird und die Messung realisiert.

Es soll hier nicht weiter auf diesen nicht praktikablen Ansatz eingegangen werden.

6 Alternative B

Es wird eine Experiment-Serie definiert, die replizierten Läufe werden also im Gegensatz zu Alternative A nacheinander gestartet.

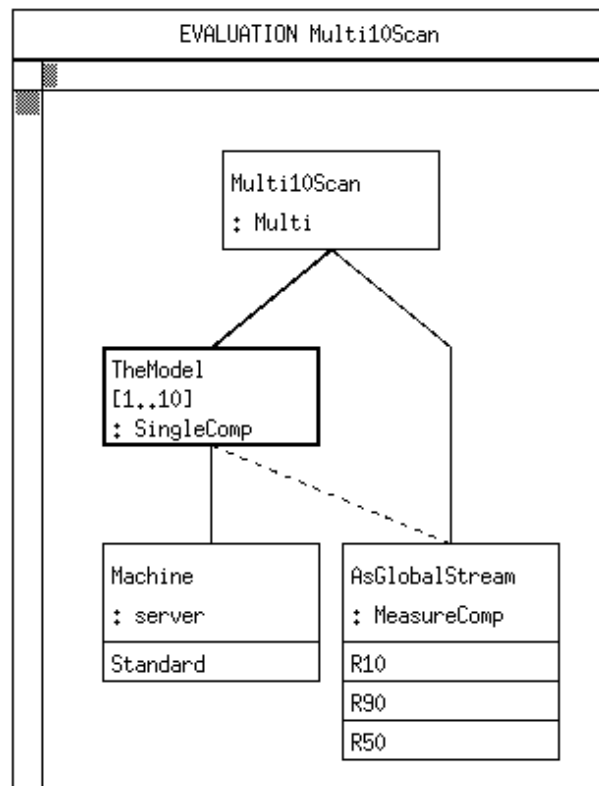


Abbildung 3: Die Modell-Hierarchie mit den Auswertungsobjekten. Das Modell, dessen transientes Verhalten zu untersuchen ist, wird als Array-Komponente eingebettet (Alternative A).

Die einzelnen Läufe könnten dabei auch als eigenständige HIT-Experimente generiert werden, um die Auswertung gleichzeitig auf verschiedenen Rechnern durchzuführen.

Für jeden der Läufe werden Leistungsmaße durch Auswertungen von Auswertungsobjekten der Form

```

MEASURE
  POPULATION
AT
  R_<t>
DUE TO
  ALL
ESTIMATOR
  MEAN
START
  MODELTIME <t> - <delta_t>
STOP
  MODELTIME <t> + <delta_t>;
  
```

ermittelt.

Die Population zum Zeitpunkt t wird also durch die mittlere Population im Intervall $[t - \Delta t, t + \Delta t]$ angenähert.

Die Zusammenfassung der Ergebnisse der einzelnen Läufe kann

1. in der Experiment-Beschreibung durch Verwendung von `get_result()`
2. extern, durch ein C-Programm

erfolgen.

Letzteres ist zu empfehlen, da es größere Flexibilität erlaubt und eine automatische Generierung von HI-Slang-Code für die Auswertungsobjekte sinnvoll ist.

Als größtes offenes Problem bleibt die Bestimmung der Länge des Meßintervalls. Hierzu könnte man eine Simulation starten, die solange läuft, bis die Mittelwerte ein gewisses Konfidenzlevel L erreichen. Das dabei simulierte Modellzeitintervall $[0, T]$ schließt die transiente Phase dann sicher ganz ein.

Zusammenfassend könnte die transiente Analyse folgendermaßen ablaufen:

1. Generierung von Last- und Maschinenparametern für die transiente Analyse
2. Generierung eines HIT-Modells zur Bestimmung von T
3. Auswertung des Modells
4. Generieren der HIT-Modelle für die transiente Analyse
5. Auswertung der Modelle
6. Zusammenfassen der Ergebnisse

6.1 Beispiel

Ein konkretes Beispiel wurde bisher noch nicht untersucht.

7 Schlußfolgerungen

Alternative B scheint am besten geeignet zu sein und wird deshalb weiter untersucht.