



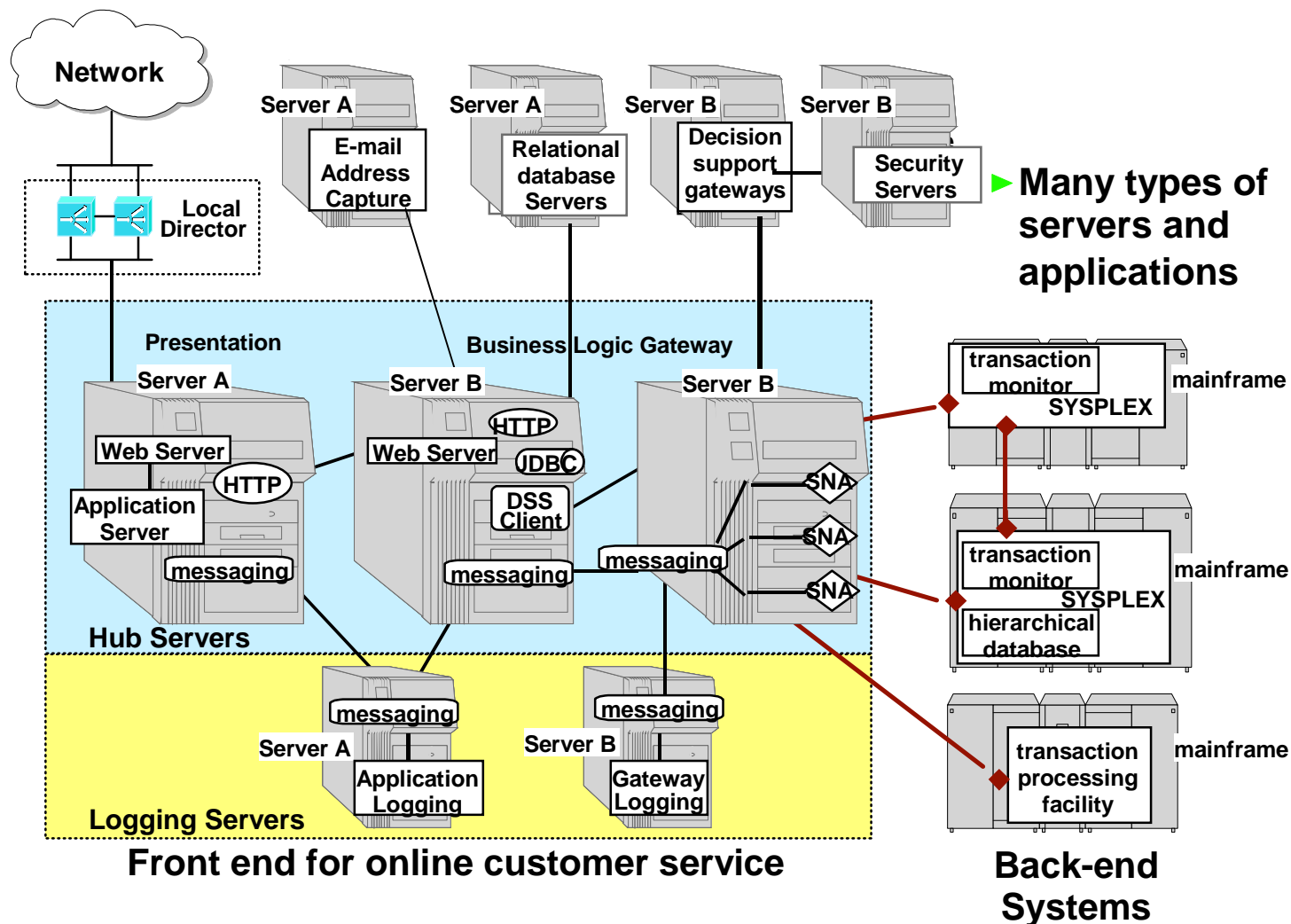
IBM T.J. Watson Research Center

Generic On-Line Discovery of Quantitative Models for Service Level Management

Y Diao, F Eskesen, S Froehlich,
J Hellerstein, A Keller, LF Spainhower, M Surendra
IBM Research, IBM Server Group

Managing Service Goals in a Complex Environment

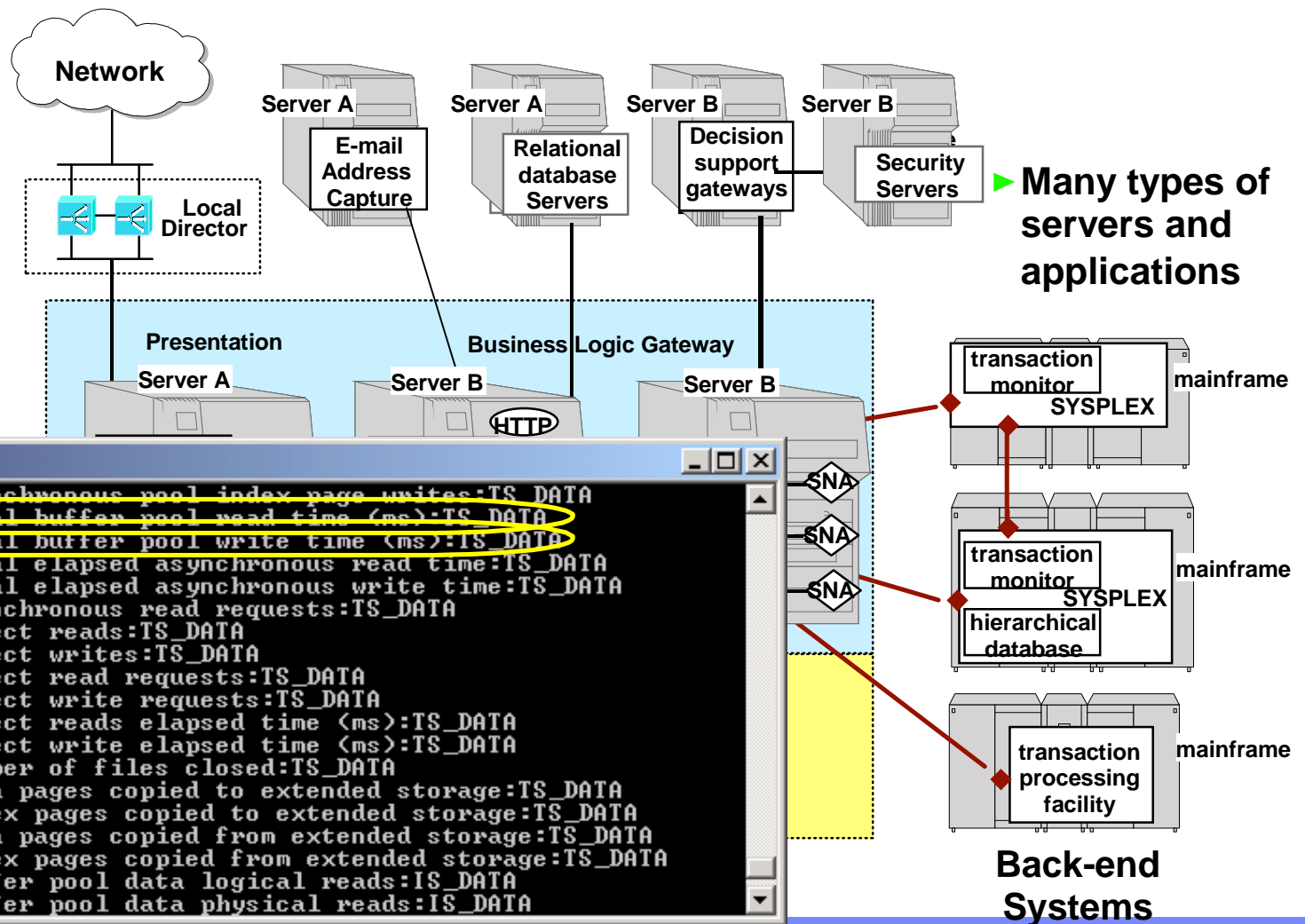
- ▶ Large number of configuration parameters and system metrics
 - WAS: ~ 100 config param
 - DB2: many 100's metrics



Managing Service Goals in a Complex Environment

► Large number of configuration parameters and system metrics

- WAS: ~ 100 config param
- DB2: many 100's metrics
- which are important?

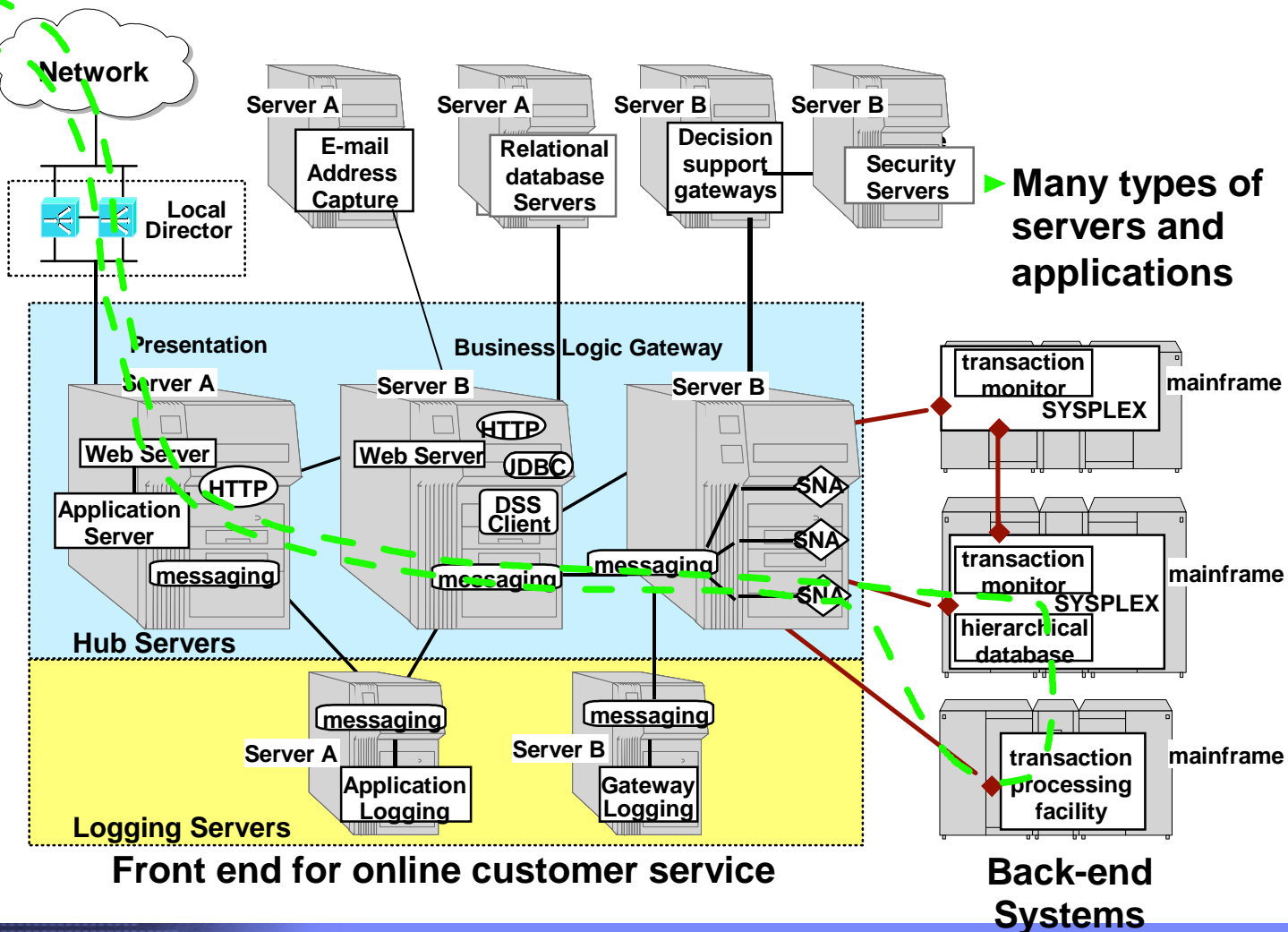


Managing Service Goals in a Complex Environment



business transaction

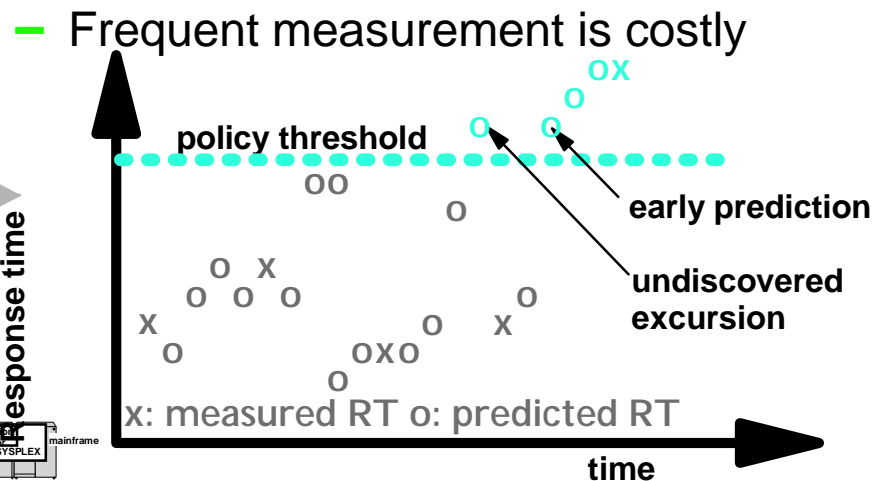
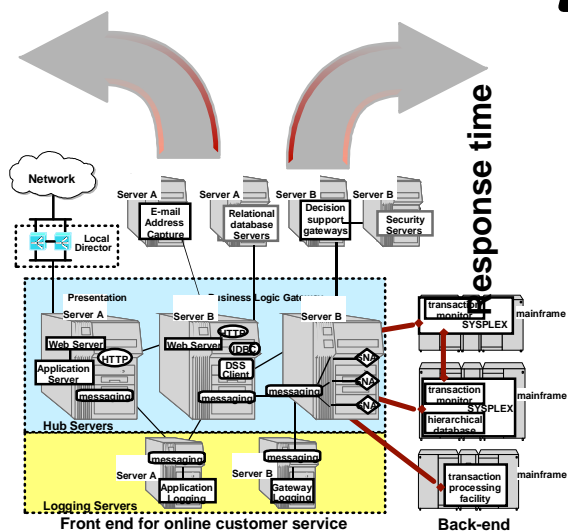
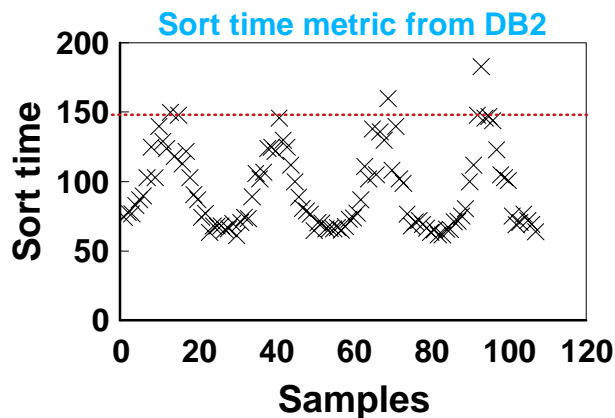
- ▶ Large number of configuration parameters and system metrics
 - WAS: ~ 100 config param
 - DB2: many 100's metrics
 - which are important?
- ▶ Business-level metrics are hard to obtain
 - response time, throughput for business transactions



▶ Many types of servers and applications

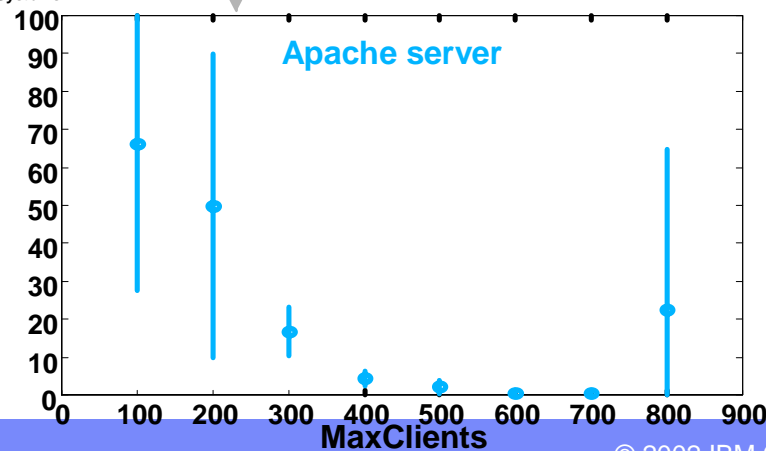
Requirements for Managing Complex Environments

- Characterize safe operating regions
 - ▶ Requirement: Know metrics that characterize service goals
- Detect service level violations early
 - ▶ Requirement: Frequent measurements or predictive models



- Optimize configuration
 - ▶ Requirement: Models that relate controls to metrics

Response time (sec.)



AutoTune Evolution: From Feasibility Study to Real Life

Published at:

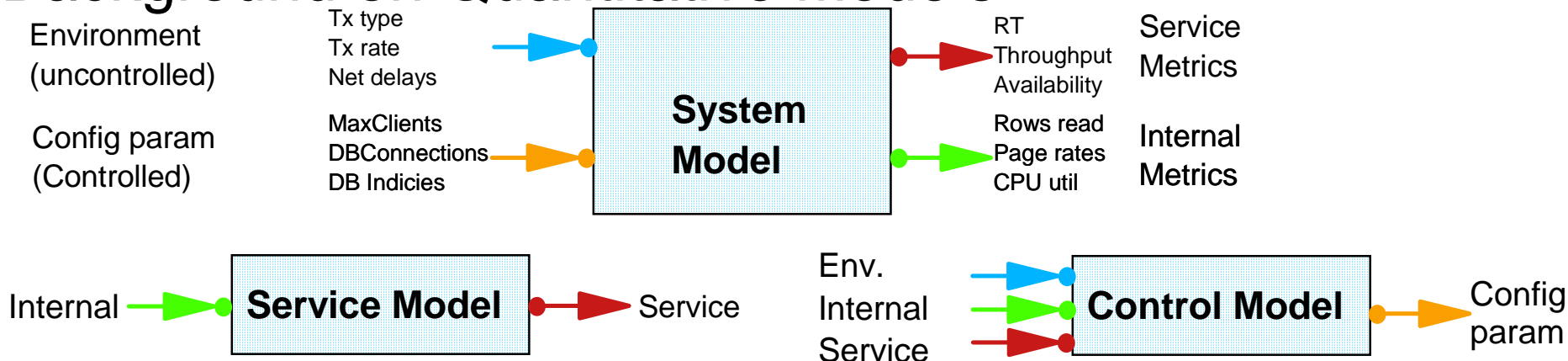
NOMS 2002

IM 2003

Research Focus	Finding "right Controller"	Real-Life Environment
Feedback Controller Type	Feasibility w/ SISO, MIMO	MIMO
Managed Resource	Apache Web Server	IBM DB2 UDB v8.1
# Metric Instances	MEM Usage, CPU Usage	~ 500
# Controls	MaxClients, KeepAliveTO (proprietary Extensions)	35 (out of the box)
Instrumentation	Apache API	CIM Model
Goals	Fixed (50% CPU Util, 60% MEM Util)	Service Levels
Resource Metrics	Known up-front	Discovered on-line
Workload Model	Synthetic (Wagon)	Real (TPC-W)

[NOMS2002]: "Using MIMO Feedback Control to Enforce Policies for Interrelated Metrics With Application to the Apache Web Server"

Background on Quantitative Models



e.g. Multiple Linear Regression

$$RT = b_0 + b_1 * \text{LogicalReads} + b_2 * \text{SortTime} \dots$$

e.g. Proportional Control

$$\text{MaxClients} = K_{p11} * (RT - RT_{\text{goal}}) + K_{p12} * (\text{ThruPut} - \text{ThruPut}_{\text{goal}})$$

$$\text{KeepAlive} = K_{p21} * (RT - RT_{\text{goal}}) + K_{p22} * (\text{ThruPut} - \text{ThruPut}_{\text{goal}})$$

Core Technologies

➤ **Metric discovery.** *Run-time* discovery of service model (especially between resource and service metrics).

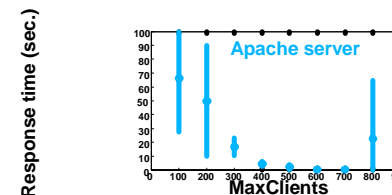
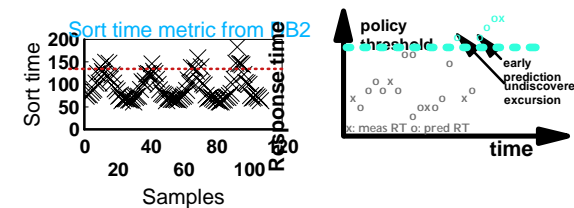
- Ex: DB2 RT determined by: exec appl, sort time, # sorts
- Uses semantic (via CIM) and statistical variable selection

➤ **Control discovery.** *Run-time* discovery of control model.

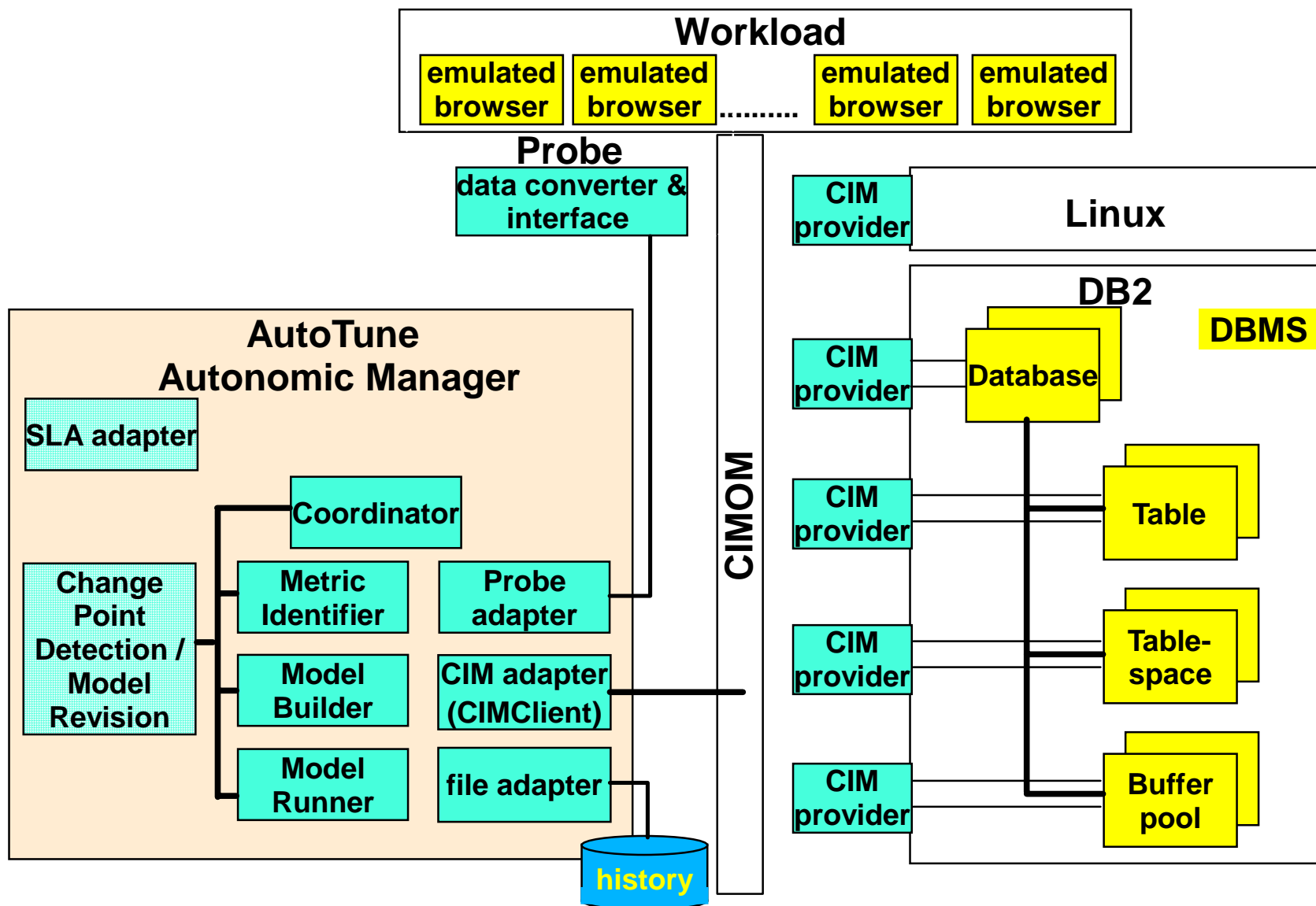
- Ex: Concave relationship between Apache MaxClients & RT
- Basis for generic control.

➤ **CIM interface for instrumentation.**

- Self-describing metrics
- Ex: In database model, statistical (metrics) and configuration parameters are distinguished
- Web services, CIM integration



Prototype: DB2 on Linux platform with TPC-W workload

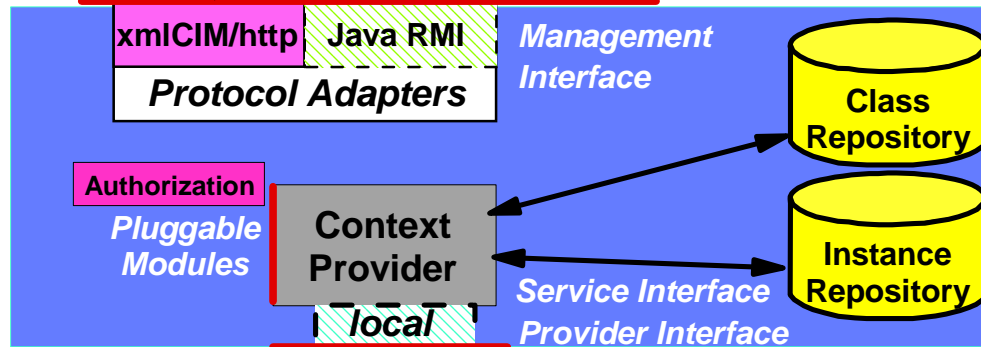


Implementing the CIM Models

CIM Object Manager

- forwards requests & responses
- handles naming & authorization

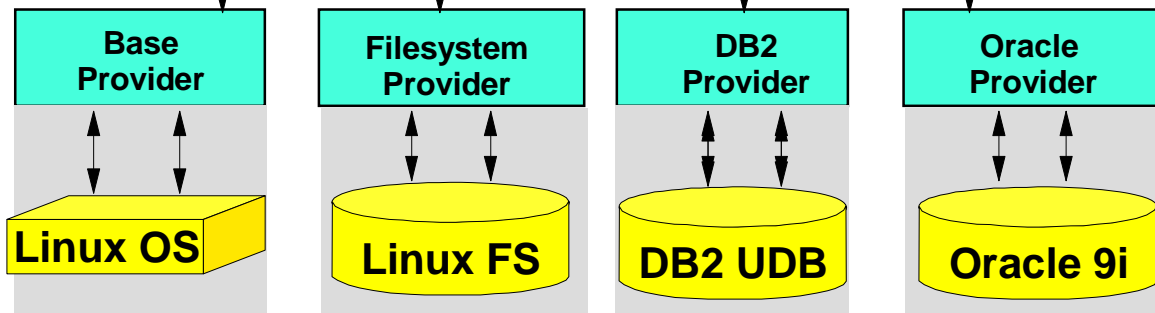
To Manager



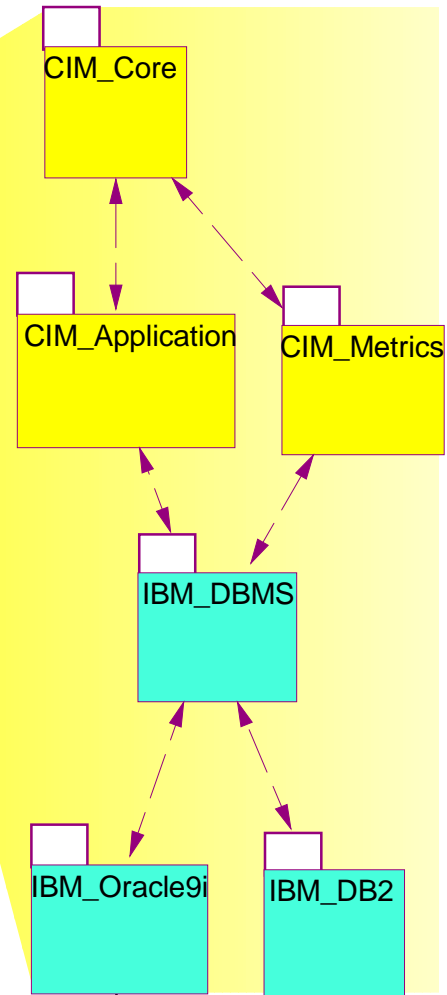
Abstraction layer



Glue code for access



Managed resource

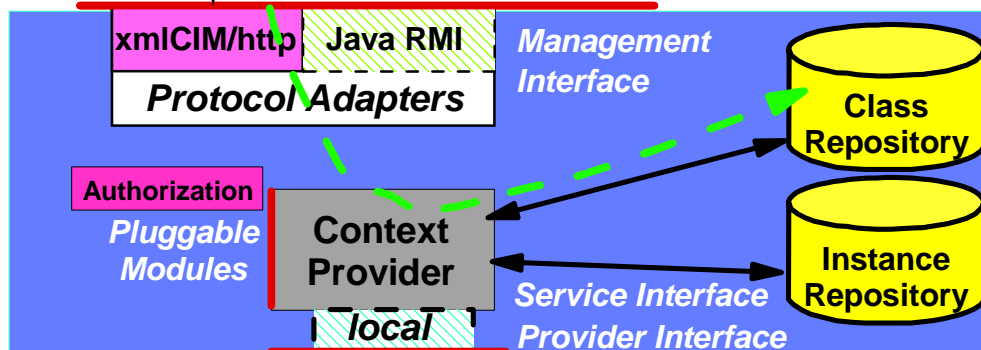


Implementing the CIM Models

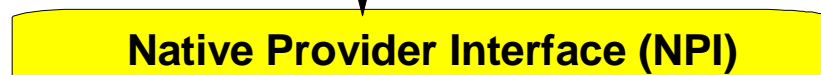
CIM Object Manager

- forwards requests & responses
- handles naming & authorization

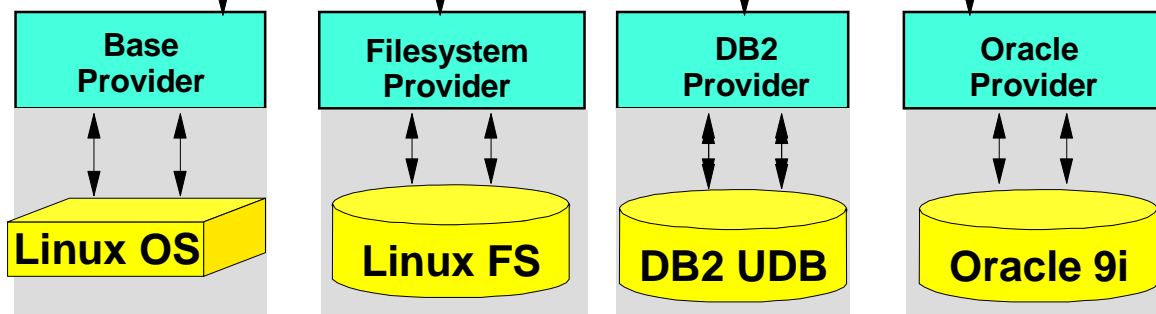
To Manager



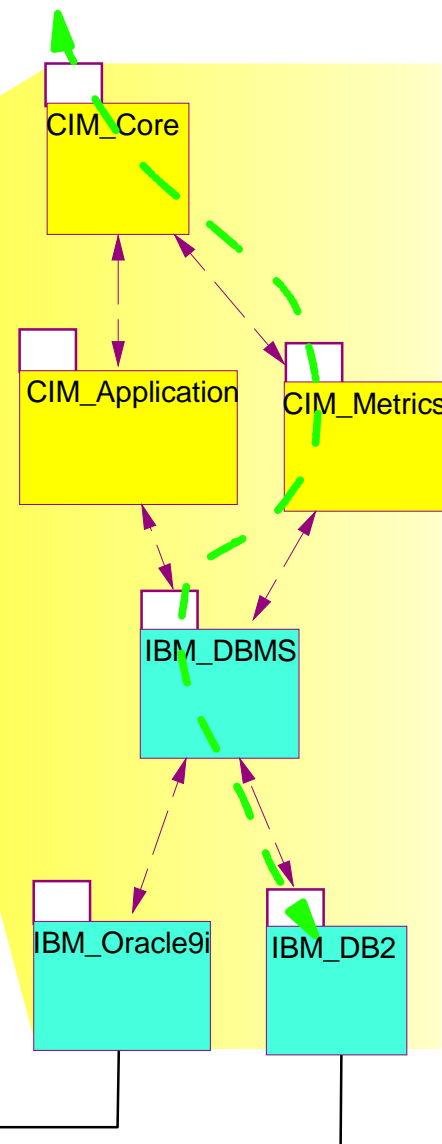
Abstraction layer



Glue code for access



Managed resource



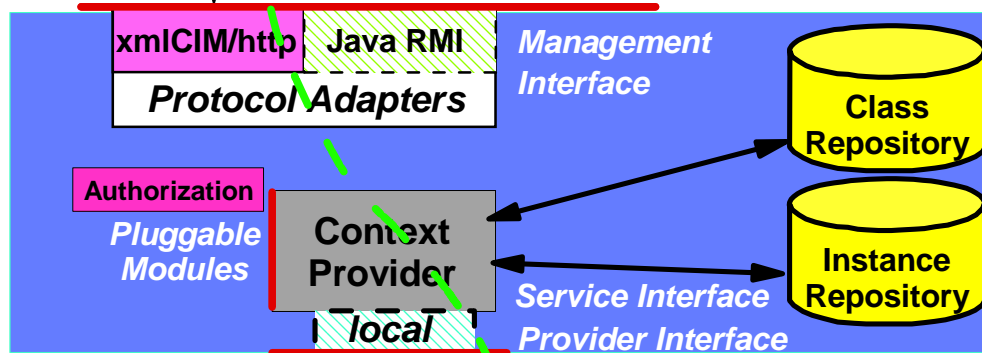
Implementing the CIM Models

Get metrics

To Manager

CIM Object Manager

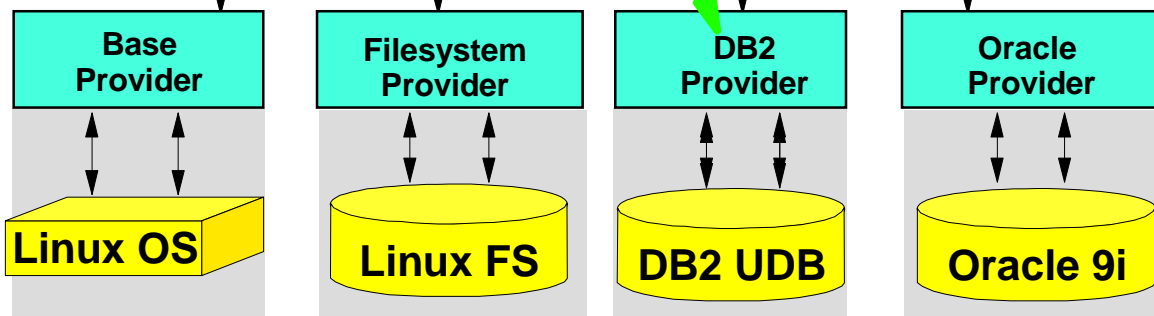
- forwards requests & responses
- handles naming & authorization



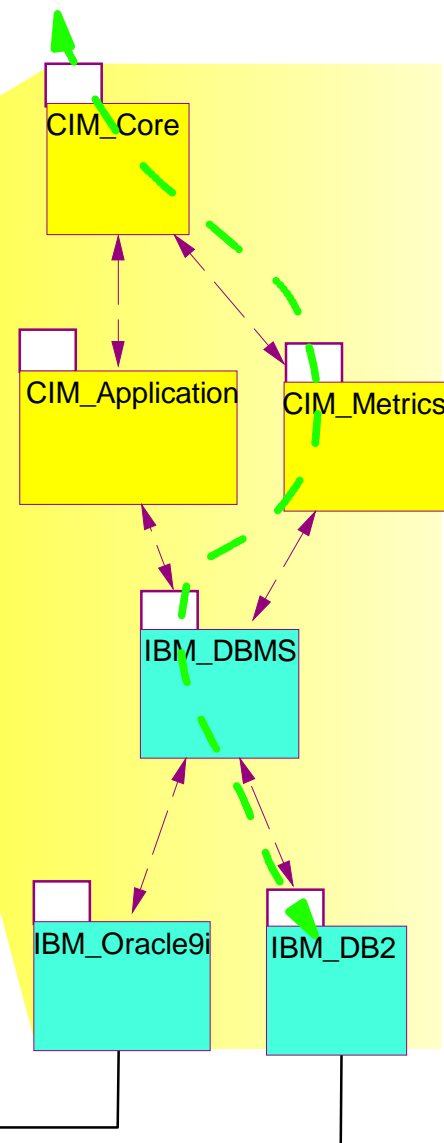
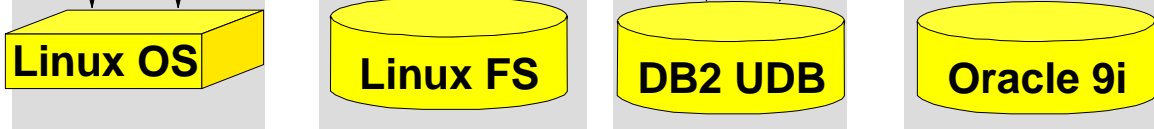
Abstraction layer



Glue code for access

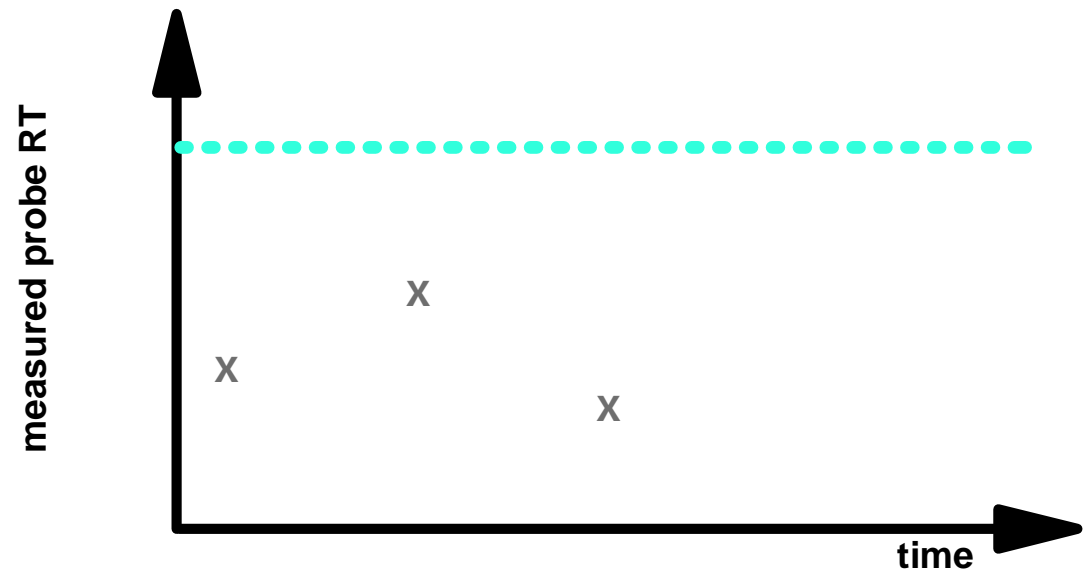
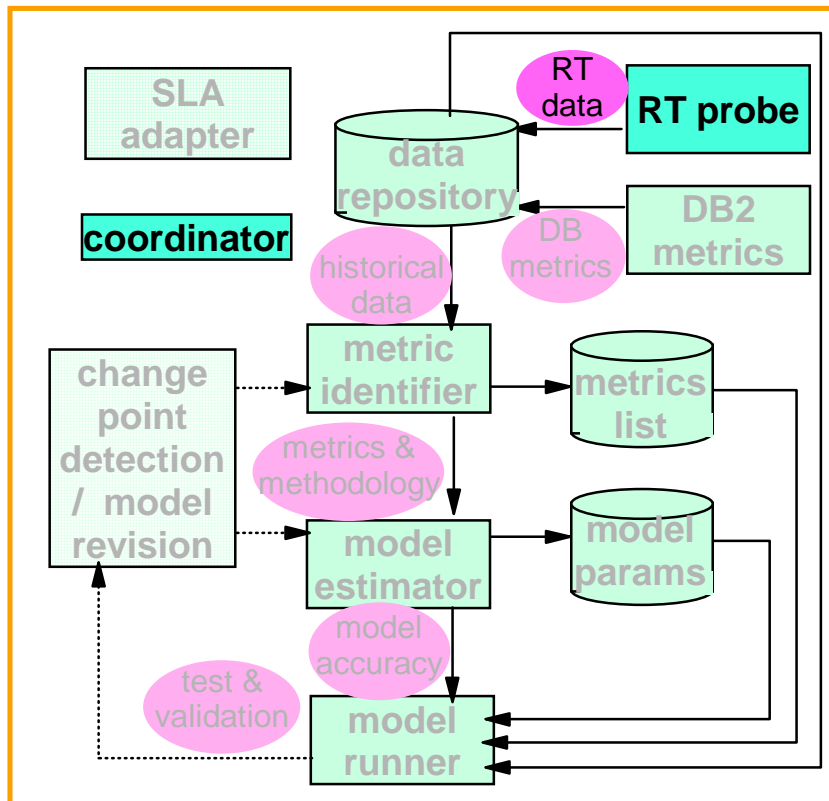


Managed resource



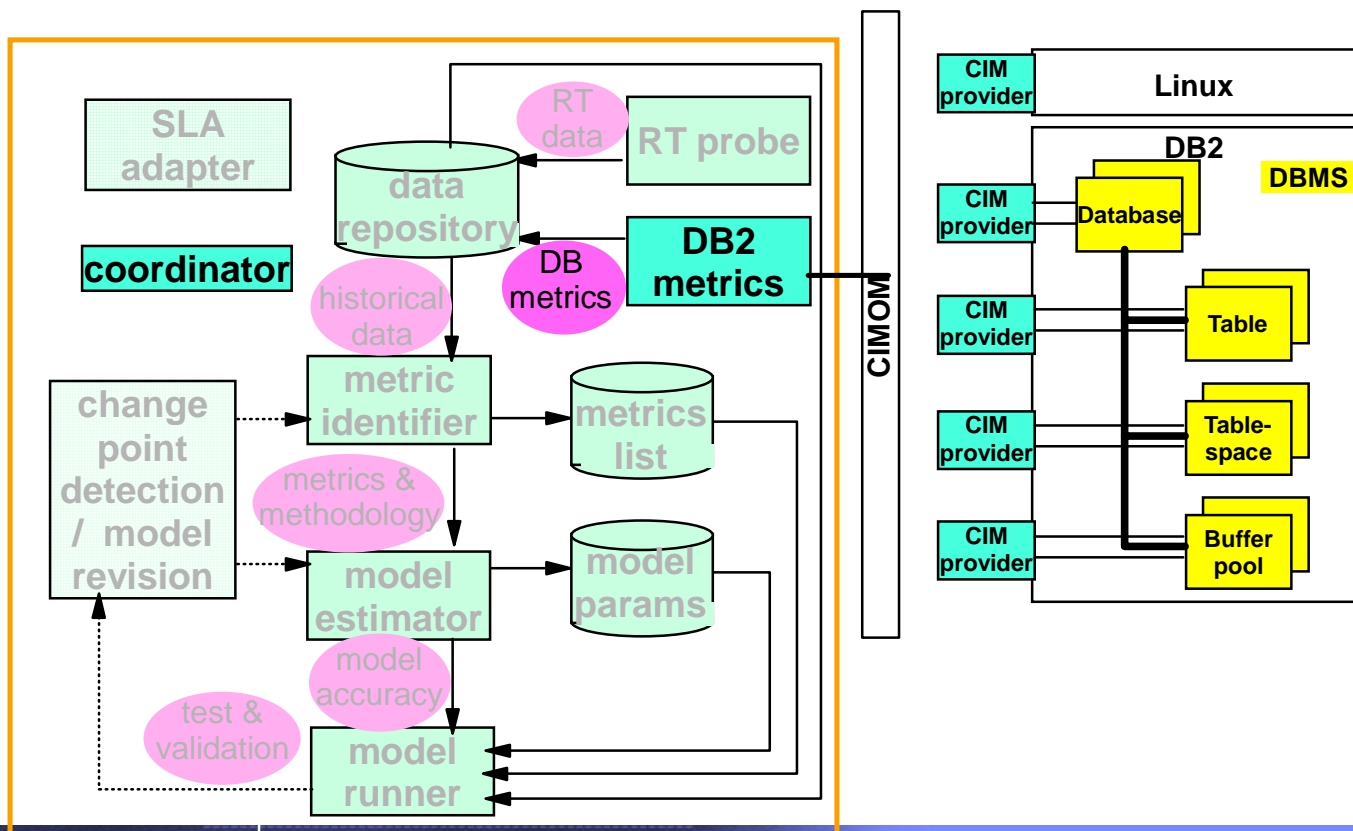
1. Data collection from Response Time Probe

- ▶ Data may only be collected at infrequent intervals
 - e.g. Keynote agents typically probe web sites every 15 - 60 min
 - multiple agents typically used (e.g. Keynote Systems uses 25 - 100 agents)



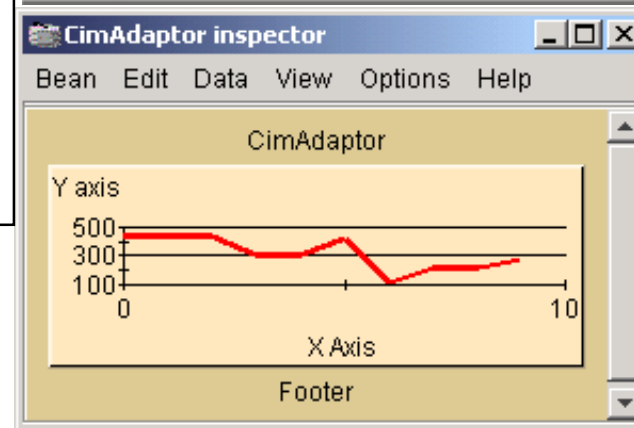
2. Collection of Internal Metrics from DB2 via CIM Providers

- Internal DB2 metrics collected by Autonomic Manager through CIM interface
 - data may be collected at relatively frequent intervals without excessively loading the system



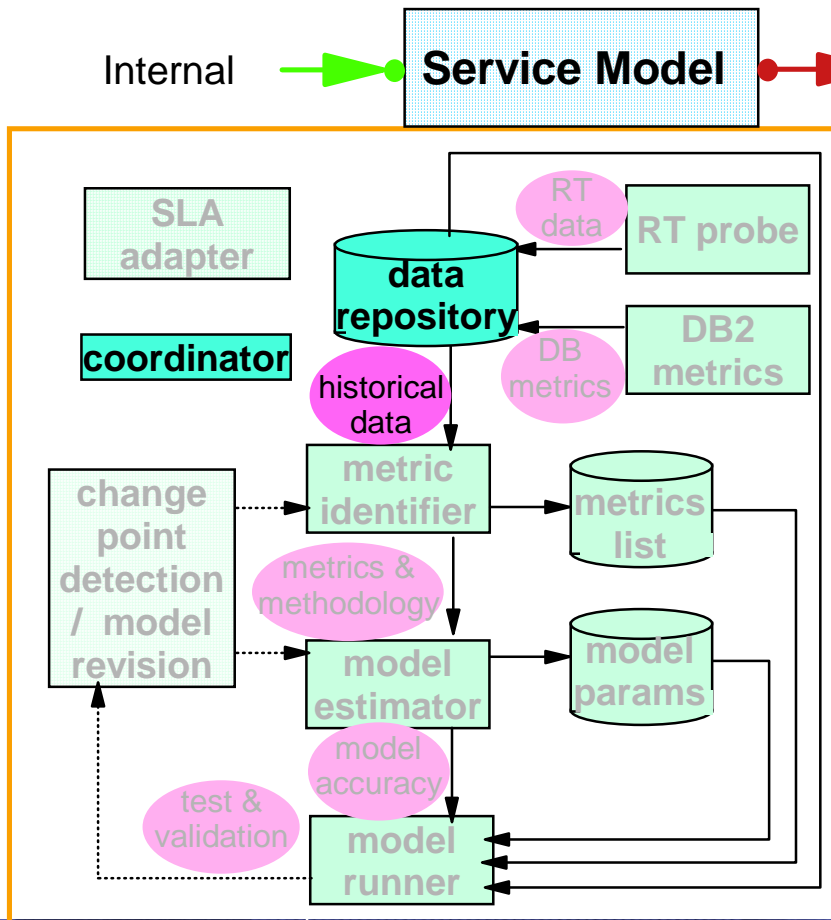
```

C:\WINNT\system32\cmd.exe
Value: IBM DB2 Data Base Management System Tablespace Statistical Data. IBM and
DB2 are registered trademarks of the International Business Machines Corporation
Name: IBM_DB2TableSD.InstanceID=IBMDB2="tpcu", DBNAME="TPCV", TABLE="TEMP (00001,
00002)">IndexReads
Value: 2255
Name: IBM_DB2BufferSpaceSD.InstanceID=IBMDB2="tpcu", DBNAME="TPCV", DPOOL="IBMDEP
0130P">IndexReads
Value: 0
Name: IBM_DB2TablespaceSD.InstanceID=IBMDB2="tpcu", DBNAME="TPCV", SPACE="USERSP
001">PoolIndexReads
Value: 0
Name: IBM_DB2TablespaceSD.InstanceID=IBMDB2="tpcu", DBNAME="TPCV", SPACE="USERSP
001">PoolIndexReads
Value: 0
Name: IBM_DB2TablespaceSD.InstanceID=IBMDB2="tpcu", DBNAME="TPCV", SPACE="SYSCAT
SPACE">InstanceID
Value: IBMDB2="tpcu", DBNAME="TPCV", SPACE="SYSCATSPACE"
Name: IBM_DB2TablespaceSD.InstanceID=IBMDB2="tpcu", DBNAME="TPCV", SPACE="SYSCAT
SPACE">PoolIndexReads
Value: 0
Total properties: 452
Number of metrics: 353
    
```

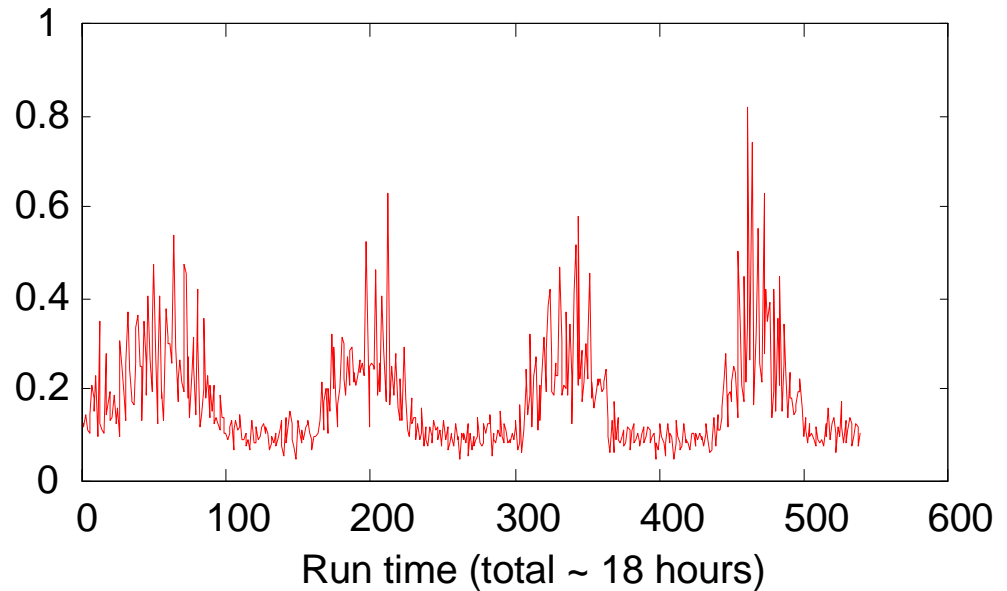


3. Historical Data for Model Building

- Model is built using historical data
 - ▶ Data collected over extended period from response time probes and DB
 - mixed workload (vary buy/browse) and varied (# emulated browsers)

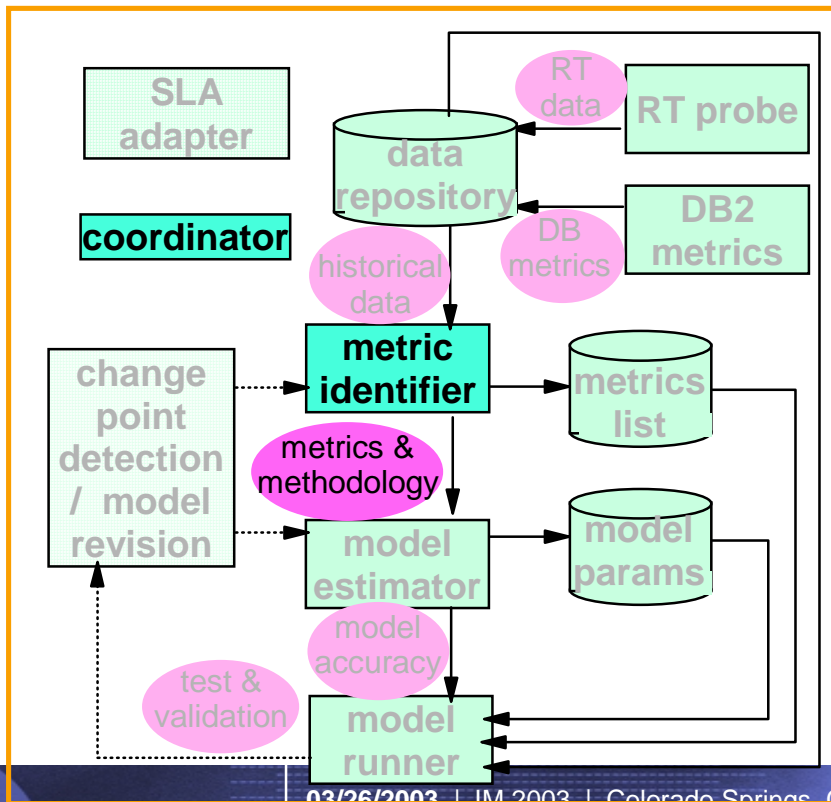


Response time



4. Metric Discovery: Find Relevant/Useful DB Metrics

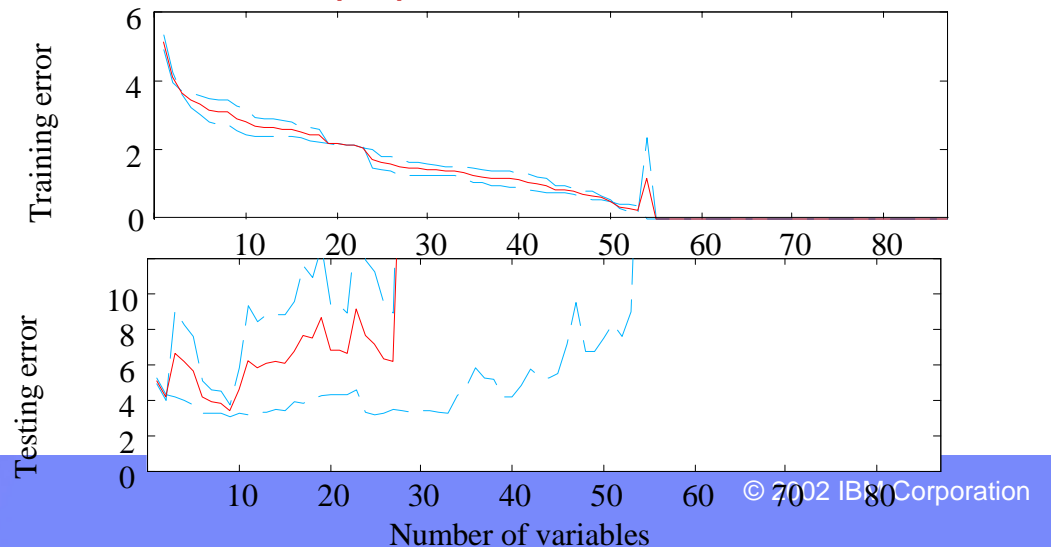
- Eliminate constant metrics
 - reduce # from 470 to ~ 150
- Some metrics are counters:
 - ▶ include 1st order differences
 - doubles # of metrics (~ 300)
- Elim. highly cross correlated metrics
 - reduce # to ~ 90
- Eliminate metrics with very low correlation to response time
 - reduce # to ~ 70
- Do stepwise model regression
 - ▶ Useful metrics selected based on significant reduction of model error
 - reduce # to < 10



[10] Database Snapshot:Appls. executing in db manager currently

[299] Tablespace Snapshot:Total buffer pool read time (ms):TS_DATA

[39] Database Snapshot:Total buffer pool write time (ms)



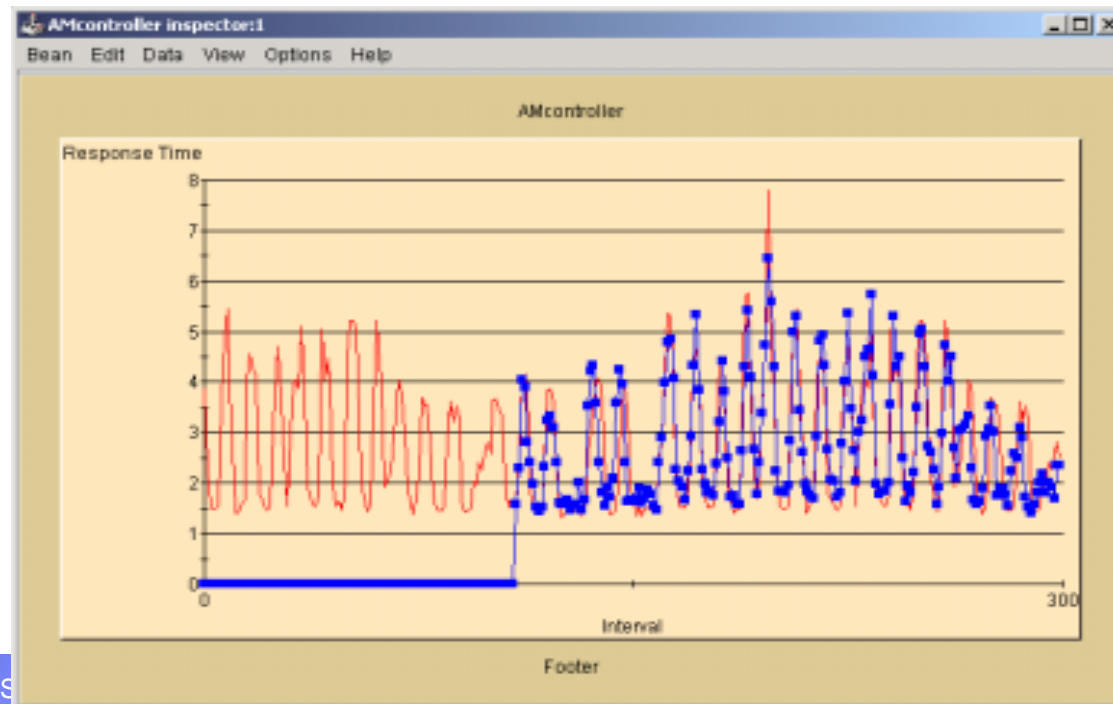
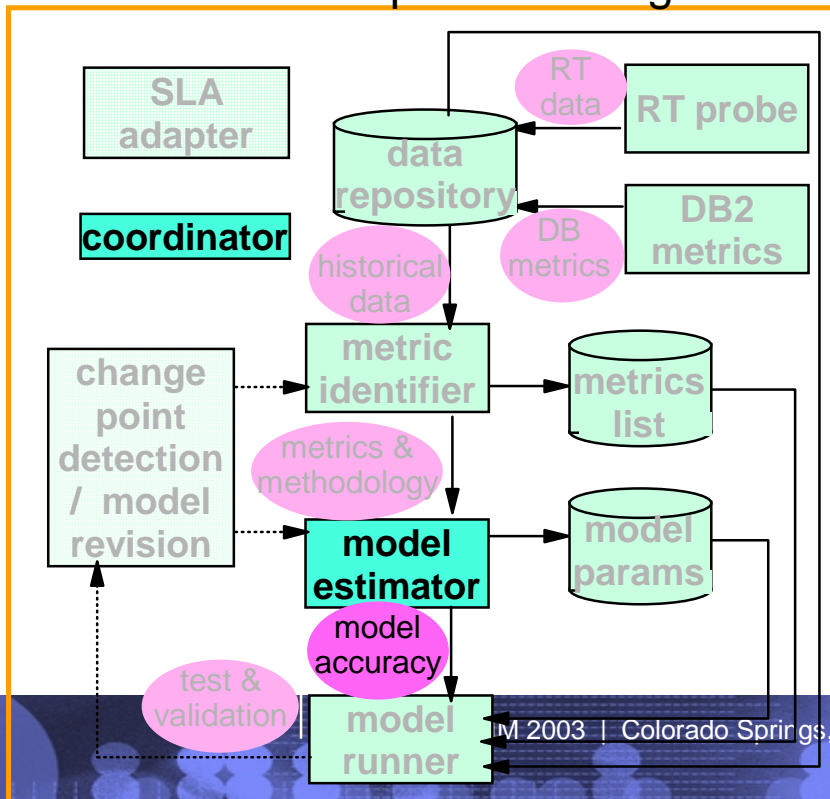
5. Estimation Model for Predicted Response Time

- Linear regression model
 - $RT = b_0 + b_1[\text{DB Snapshot:Appls exec in db mgr currently}] + b_2[\text{Tblspace Snapshot:Tot buffer pool read time:TS_DATA}] + b_3[\text{DB Snapshot:Total buffer pool write time}]$
- Other models possible - e.g. neural net

```

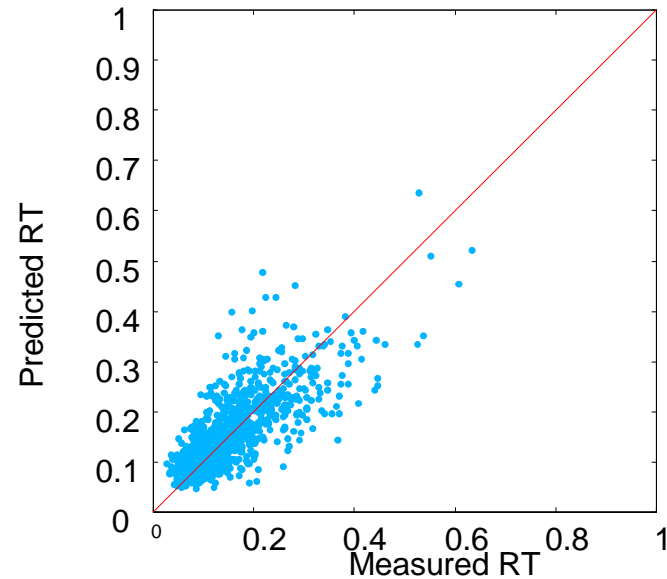
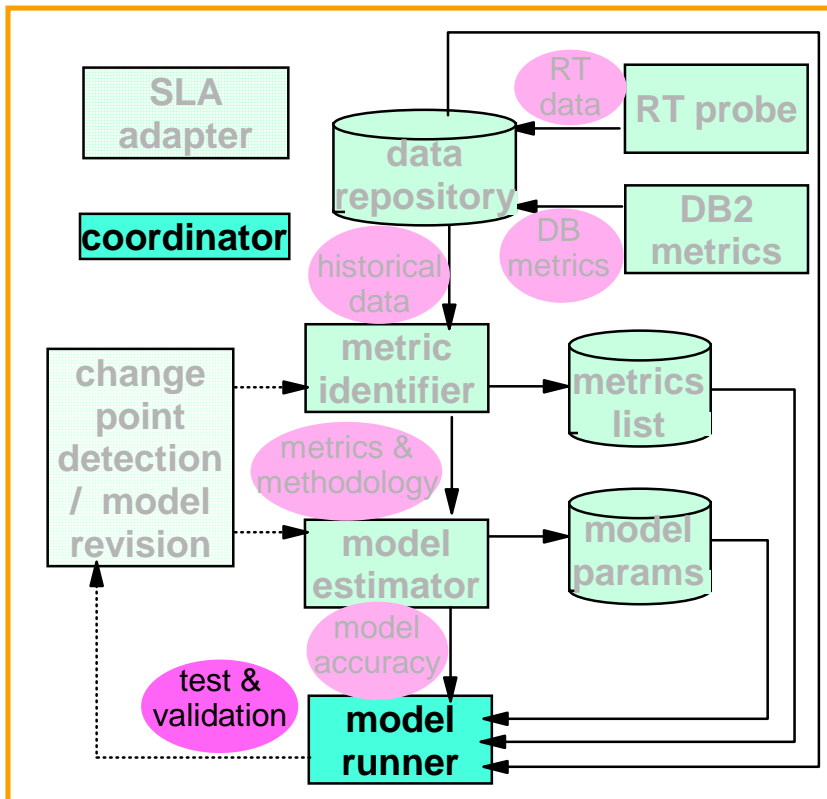
AMcontroller inspector
Bean Edit Data View Options Help

inputBuffer: null
outputBuffer:
  [0]: 2.5278063  2.3409660588003294
  [2]: 10  1.442987269528304
  [4]: 299  8.937219733780748E-5
  [6]: 39  9.69398928715648E-7
    
```

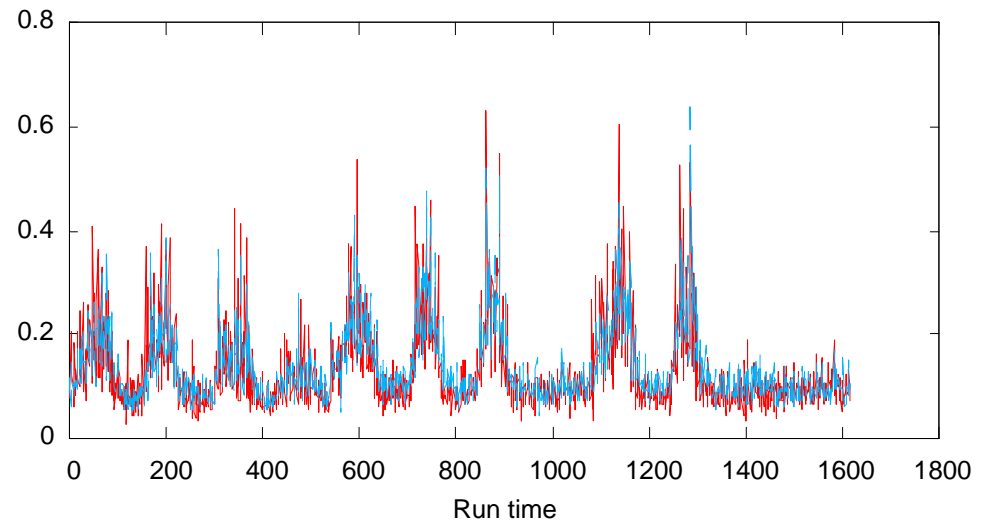


6. Test Estimation Model with New Data

- Inference model test with saved data
 - ▶ Possibility for incremental model improvement
 - e.g. recursive least squares
 - ▶ Significant model revision driven by
 - changes of workload pattern
 - changes to resource deployment



Predicted and actual response time



Conclusions and Outlook

- **Wanted:**
 - ▶ Discovery of functional Relationships between Resource Metrics and SLA Parameters
 - ▶ Sample System: DB2 UDB v8.1 with TPC-W Workload Mix
 - ▶ $\text{ResponseTime} := b_0 + b_1 * \text{LogicalReads} + b_2 * \text{SortTime} \dots$
- **On-line Discovery of Quantitative Models:**
 - ▶ No prior Knowledge of Managed Element Details
 - ▶ Discovery of Metrics through CIM Instrumentation of DB2
 - leverages CIM Inheritance, Type System and Qualifiers (Counters vs. Gauges)
 - ▶ Metrics are used as explanatory variables of quantitative Model
 - Of a total of 500 Resource Metrics, 3 characteristic Metrics are identified:
 - ApplicationsExecutingInDB, BufferPoolReadTime, BufferPoolWriteTime
 - validated with DB2 UDB Toronto Development Team
 - predicts Variability of Response Time with 72% Confidence
 - ...for a very specific Workload Pattern...
- **Current Work:**
 - ▶ Refine Metric Discovery:
 - Decrease training Times (today: 18 hours)
 - Evaluate on more Workloads (today: 15-30 TPC-W Emulated Browsers)
 - ▶ Multi-tiered e-Business Environment, Variety of Components (Apache, WebSphere, DB2)