

# Scalability Evaluation

TripCom GA Galway

29 Apr 2008

Hans Moritsch, TUW

# Three Dimensions of Quantitative Scalability Evaluation



- work load on the system
  - e.g., the number of users (clients)
- computational resources available
  - e.g., the number of machines (kernels)
- performance metrics
  - e.g., the response time (latency)
- *a scalable system can handle the addition of users and resources without suffering a noticeable loss of performance*
  - B. C. Neuman, Scale in Distributed Systems, 1994

# Scalability Evaluation Procedure



- Specify indicators for
  - load
  - resources
  - performance
- Gather data: capture relationship between load/resources/performance
  - design experiments (specify test cases)
  - develop measurement program
  - generate test data
  - run measurement program
- Analyze results
  - check whether measured data fit to scalability model
    - regression analysis

# Distributed Trivial Tuple Space (DTTS)



- test space for developing scalability evaluation procedure
- the only tuple type supported is `(int, String)`
- built of the following components
  - *host* instances jointly establish a distributed shared memory (space) which stores strings
  - this space is equally distributed onto the host instances
  - *node* instances periodically issue `rd(n, ?String)`
    - with random `n`
- communication via RMI
- hosts artificially create CPU demand of operations
  - e.g. 100msec CPU time/operation

# Scalability Measurements on DTTS

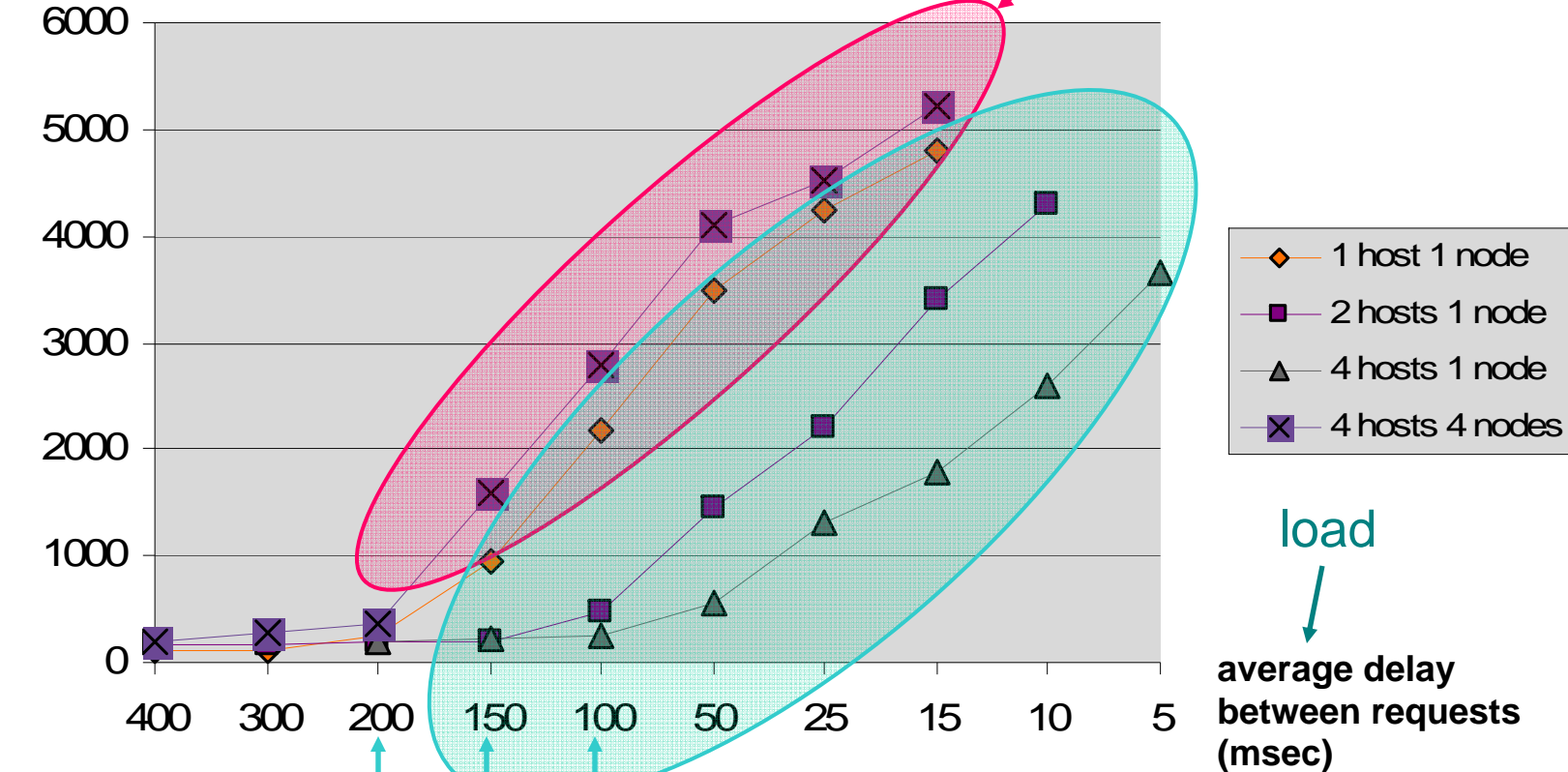


performance

## Scalability of Latency wrt. Request Frequency,

latency (msec)

near linear scalability



load  
average delay between requests (msec)

- some results of the OOPSLA 2003 Workshop on Middleware Benchmarking
  - rough assessment through a few general measurements
    - more specific measurements for exploration of a behavior
    - understanding the behavior is important
  - timestamps are not necessarily sufficient
    - annotate timestamps with data on system utilization and resource consumption
  - determine scalability requirements
    - what must scale ? (number of concurrent user sessions, concurrent requests, concurrent transactions, ...)
    - the application must scale
    - based on critical and representative use case scenarios  
→ application “kernels”

- **numberOfSpaces**
  - the number of spaces managed by a kernel [1..100]
- **numberOfTriplesStored**
  - the number of triples stored in a space [1000..1000000]
- **tripleSizeStored**
  - the average size of triples data stored [100B..1KB]
- **numberOfClients**
  - the number of clients simultaneously issuing requests to a kernel [1..1000]
- **requestFrequency**
  - the frequency of requests (per second, per client) [1..10]
- **numberOfTriplesTransferred**
  - the number of triples transferred between kernel and client [100..100000]
- **tripleSizeTransferred**
  - the average size of triples transferred [100B..1KB]
- **numberOfSpacesRd** *cannot be explicitly prescribed but is reported as a result of rd operation*
  - the number of spaces actually read (without URL) [1..100]

## ■ **numberOfKernels**

- the number of kernels [1..100]

## ■ **numberOfManagingKernels**

- the number of kernels managing a space (assuming the distribution of spaces onto multiple kernels) [1..8]

## ■ **numberOfMachines**

- the number of physical machines on which a kernel is executed (assuming kernel components running on different machines) [1..8]

## ■ **numberOfInstances<component>**

- the number of instances of a component (assuming multiple instances of components) [1..8]

## ■ **responseTime**

- the period of time between calling a TS API function and control returning to the client program
- in case of non blocking operations

## ■ **modificationTime**

- the period of time between calling a TS API function and the effect on persistent storage
- in case of operations which change the contents of a space

## ■ **notificationTime**

- the period of time between changing the contents of a space and the completed receipt of data by the callback
- in case of subscriptions)

## ■ **throughput**

- the number of operations processed in one unit of time (sec)

# Initial Restrictions



- no subspaces
- all operations permitted by security manager
- neither errors nor exceptions occur
- no transactions