Scalability evaluation methodology and schedule

Dimensions of Scalability

- Performance Measure *p*
  - high value: „good“ performance
  - response time, latency
  - throughput
- Computational Resource *r*
  - number of kernels managing a space
  - number of physical machines
  - clock speed, memory size, interconnection bandwith, ...
- Computational Workload *l*
  - number of clients
  - number of requests/client
  - volume of data transfered

Definition of Scalability

- functions p(l) decreasing, p(r) increasing
- Scalability „how is p affected when r is increased in order to compensate for larger l“
  - e.g., p(l,r) = p(2l,2r) ... „ideal“ scalability
- Performance model of a system: define sets
  - of resources R
  - of workload parameters L
  - of performance measures P
- Concretize Functions *p(l,r)*
  - analytically, *e.g., response time = c * log(data size)*
  - or via experiments
- Definition - and evaluation - of scalability on the basis of functions *p(l,r)*

Scalability Definition

Steen/Van der Zijden/Sips 1998

Definition. Let $\gamma(\mathbf{a})$ be a function in attribute values $\mathbf{a}$ returning values in the same cost unit as $Cost_A$. Furthermore, let $\delta(\mathbf{a})$ be a function in $\mathbf{a}$ returning values in the same performance measure unit as $Perf_A$ with $\forall \mathbf{a} :: \delta(\mathbf{a}) \geq 0$ and $\delta(\mathbf{a}_{\text{ref}}) = 0$. An application A is scalable in attribute $Attr_i$ for values up to a maximum $a_{\max}$ with respect to γ and δ if the following properties hold:

P1: A can accommodate values $\mathbf{a}_{\text{ref}}[i] < a < a_{\max}$.

P2: $\forall \mathbf{a}_{\text{ref}}[i] < a < a_{\max} \exists \mathbf{r} ::$

$$Perf_A(\mathbf{a}_{\text{ref}}, \mathbf{r}_{\text{ref}}) - Perf_A(\mathbf{a}_{\text{ref}}\langle i : a\rangle, \mathbf{r}) \leq \delta(\mathbf{a}_{\text{ref}}\langle i : a\rangle)$$

P3: $\forall \mathbf{a}_{\text{ref}}[i] < a < a_{\max} ::$

$$Cost_A(\mathbf{a}_{\text{ref}}\langle i : a\rangle, \mathbf{r}_{\min}(\mathbf{a}_{\text{ref}}\langle i : a\rangle)) \leq \gamma(\mathbf{a}_{\text{ref}}\langle i : a\rangle)$$

Example

T(S,n) ... response time of task of size S on n machines

$$scalabilityfactor_S(k) = \frac{T(S,1)}{T(kS,k)}$$

$$Perf_A(\mathbf{a},\mathbf{r}) = -T(S,n) = -b\frac{S}{n}$$

Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

Example (2)

P1: A can accomodate values $S_1 < S \leq S_{\max}$ (there is a maximum task size)

P2: $\forall\, S_1 < S \leq S_{\max}\ \exists k :: -T(S_1,1) + T(kS_1,k) \leq \delta(kS_1)$. Via

$T(kS_1,k) \leq T(S_1,1) + d(kS_1 - S)$ and

$\frac{T(S_1,1)}{T(kS_1,k)} \geq 1 - \frac{d(kS_1-S)}{T(kS_1,k)}$

it turns out that the program is scalable in task size, if

$$scalabilityfactor_{S_1}(k) \geq 1 - \frac{dS_1(k-1)}{T(kS_1,k)}$$

Evaluation methodology

- define component performance models
- review of component performance models
- define system performance model
- review of system performance model
- analysis of interdependences between components and system performance
- build performance functions
  - qualitative/formulas
  - experiments/benchmarks
    - design experiment
      - define benchmark program
      - specify test cases
    - implement benchmark program
    - generate test data
    - run benchmark
- analyse results with respect to scalability definition used, i.e., determine scalability

Test cases

- client load parameters
  - number of concurrent clients
  - number of requests per client
  - number of triples transferred
  - size of triples
  - repeated execution of an operation
- resource parameters
  - numbers of kernels managing a space
  - ...
- operations
  - TS-API operations
    - out, in, rd, (-multiple)
    - subscribe -> notify (number of active subscriptions)

Evaluation of alternative approaches

- RDF Stores
  - ORDI
  - YARS
  - ...
- Spaces
  - JavaSpaces implementations, GigaSpaces, ...
  - Semantic Spaces (if implementation available)
  - P2P Spaces
- DHTs
- Databases
  - Amazon SimpleDB
  - ...

„TripCom Testlab“

- hardware for comparable performance and scalability experiments
- architecture, clockspeed, memory, operating system, interconnection network, ...
- number of computers