

KTH ROYAL INSTITUTE OF TECHNOLOGY



(KTH)

Middleware

Remote Invocation

Vladimir Vlassov and Johan Montelius



Remote invocation / indirect communication

Socket layer

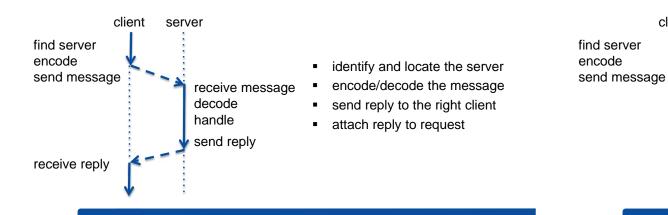
Network layer

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION





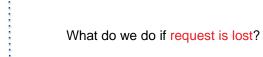
Request / Reply



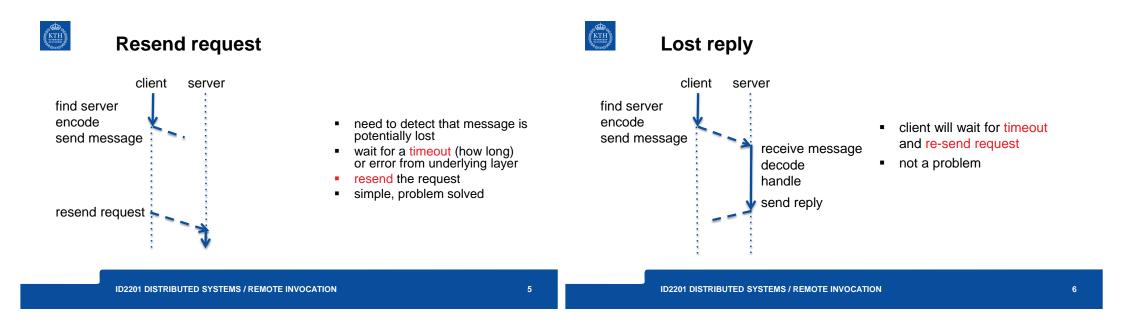
Lost request

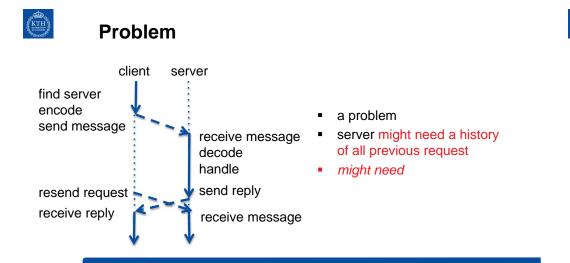
server

client



ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION







Idempotent operations

- add 100 euros to my account
- what is the status of my account
- Sweden scored yet another goal!
- The standing is now 2-1!



History

If operations are not idempotent, the server must make sure that the same request is not executed twice.

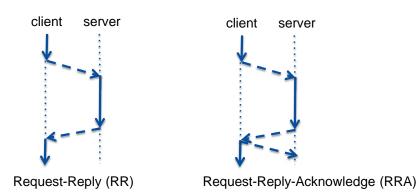
Keep a history of all request and the replies. If a request is resent the same reply can be sent without re-execution.

For how long do you keep the history?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

(KTH)

Request-Reply-Acknowledge



ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

10

(KTH)

At-most-once or At-least-once

How about this:

If an operation succeeds, then..

at-most-once: the request has been executed once.

Implemented using a history or simply not re-sending requests.

at-least-once: the request has been executed at least once.

No need for a history, simply resend requests until a reply is received.



At most or At least

How about errors:

Even if we do resend messages we will have to give up at some time.

If an operation fails/is lost, then..

at-most-once:

at-least-once:



At most or At least

Pros and cons:

- at-most-once without re-sending requests: simple to implement, not fault-tolerant
- *at-most-once with history:* expensive to implement, fault-tolerant
- *at-least-once:* simple to implement, fault-tolerant

Can you live with at-least-once semantics?



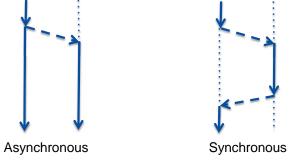
UDP or TCP

Should we implement a request-reply protocol over UDP or TCP?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	13	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	14
--	----	--	----

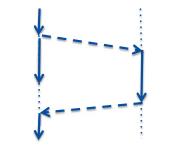
(KTH)

Synchronous or Asynchronous





RR over Asynchronous

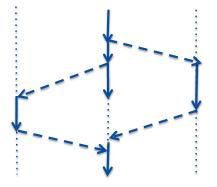


- send request
- continue to execute
- suspend if not arrived
- read reply

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



Hide the latency





HTTP

A request reply protocol, described in RFC 2616.

Request = Request-Line * (header CRLF) CRLF [message-body]

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

GET /index.html HTTP/1.1\r\n foo 42 \r\n\r\nHello

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

17

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



HTTP methods

- GET: request a resource, should be idempotent
- HEAD: request only header information
- POST: upload information to a resource, included in body, status of server could change
- PUT: add or replace a resource, idempotent
- DELETE: add or replace content, idempotent



Wireshark

70 9,473588000	130,237,72,201	130,237,215,140	DNS			response Oxa4	
71 9,473789000	130,237,215,140	130,237,28,40	TCP				00 Len=0 MSS=14
72 9,474175000	130, 237, 28, 40	130, 237, 215, 140	TCP				k=1 Win=8190 Le
73 9,474197000	130,237,215,140	130,237,28,40	TOP				1n=3737600 Len
74 9,474284000	130.237.215.140			699 GET /			
75.9,479642000	130,237,28,40	130, 237, 215, 140	TCP			f a reassemble	d poul
76 9,478672000	130, 237, 215, 140	130,237,28,40	TCP				05 Win=3842048
Ethernet II, Src: Asus Internet Protocol Vers Transmission Control P Hypertext Transfer Pro KGET / HTTP/I.1\r\n Host: www.kth.se\r\n	ion 4, Src: 130.237.2 rotocol, Src Port: 53	15.140 (130.237.215.	140), Dst: 1 t: 80 (80), 1	30.237.28.40 Seq: 1, Ack:	(130.237. 1, Len: 6	28.40)	
Accept-Language: en-U Accept-Encoding: gzig	lication/xhtml+xml.ap S.en:q=0.5\r\n . deflate\r\n	plication/xml;q=0.9,	*/*;q=0.8\r\	n			
Accept: text/html,app Accept: Encoding: gain Accept:Encoding: gain Itruncated]Cookie: (Final Cookie: (Fail: request LUEI: ht [HTTP request 1/1] COSO 66 74 00 ar 71 05 COSO 66 75 75 CO 75 32	Lisation/shtelkaml, op 5, entre0-51(10) 1, deflate(r)n utma=154244322.00010 etrrin tb://www.kth.sm/l dd 0a 555 72 55 72 55 72 60 66 60 27 53 55 62 75 66 74 75 33 55 61 30 54 36 20 75 34	plication/xml;q=0.9, 3708.1409574123.1430 441 G7 G2 kth am. 5 20 30 20 nt Mor 5 20 60 20 nt Mor 5 20 64 cot (K11; U1 2 76 58 34 nu x 80)	*/*:q=0.8\r\ 1294703.14411 1User-Age 1 Ula/5.0 5 untu: Li 64: rvi4	n		∿ll-participat	e#1 ; carftoker
Accept: text/html,app Accept: Encoding: grip Accept:Encoding: grip ItruncatedIcookie: (r/n Full request UAI: ht [HTTP request I/1] 0050 fb 74 08 20 73 80 0060 fb 74 08 20 74 95 0060 fb 75 78 20 77 8 0060 fb 75 78 20 77 8	Lisation/shtel+aml, ap Senge0.5(r/m , deflate\r/m utma=154244322.00016 te\r/m te/r/m te/r/m del de _50 72 60 72 62 73 60 72 60 72 72 73 60 70 60 74 72 33 75 62 77 66 74 72 33 75 63 43 50 70 65 74 72 72 65 60 61 73 73 32 33	plication/xml;q=0.9, 3708.1400574123.1430 4 41 67 65 5 20 30 20 5 20 40 60 (X11) U1 2 26 50 30 0 31 30 30 0 31 30 0 3	*/*:q=0.8\r\ 294703.14411 . User:Age 1 11a/5.0 5 unitu; Li	n		oll-participat	e#1 ; carftoker

20



HTTP GET

GET / HTTP/1.1 Host: www.kth.se User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:40.0) Gecko/20100101 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept.Encoding: gzip, deflate Cookie:

Connection: keep-alive



HTTP Response

HTTP/1.1 200 OK Date: Tue, 08 Sep 2015 10:37:49 GMT Server: Apache/2.2.15 (Red Hat) X-UA-Compatible: IE=edge Set-Cookie: JSESSIONID=CDC76A3;Path=/; Secure; HttpOnly Content-Language: sv-SE Content-Length: 59507 Connection: close Content-Type: text/html;charset=UTF-8 <!DOCTYPE html> <html lang="sv"> <title>KTH | Valkommen till KTH</title>

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

21

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



The web

On the web the resource is often a HTML document that is presented in a browser.

HTTP could be used as a general-purpose request-reply protocol.



REST and SOAP

Request-reply protocols for Web-services:

- REST (Representational State Transfer)
 - content described in XML, JSON, ...
 - light weight,
- SOAP (Simple Object Access Protocol)
 - over HTTP, SMTP . . .
 - content described in SOAP/XML
 - standardized, heavy weight

22



HTTP over TCP

HTTP over TCP - a good idea?



Masking a request-reply

Could we use a regular program construct to hide the fact that we do a request-reply?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	25	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	26

(KTH)

Masking a request-reply

Could we use a regular program construct to hide the fact that we do a request-reply?

• RPC: Remote Procedure Call

• **RMI**: Remote Method Invocation



Procedure calls

What is a procedure call:

- find the procedure
- give the procedure access to arguments
- pass control to the procedure
- collect the reply if any
- continue execution

How do we turn this into a tool for distributed programming?



Operational semantics

int x, n;	int x, arr[3];
n = 5;	arr[0] = 5;
proc(n);	proc(arr);
x = n;	x = arr[0];



Call by value/reference

Call by value

- A procedure is given a copy of the datum

Call by reference

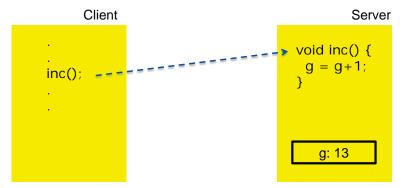
- A procedure is given a reference to the datum

What if the datum is a reference and we pass a copy of the datum? Why is this important?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION 29 ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION 30	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION 29	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	30
---	---	--	----



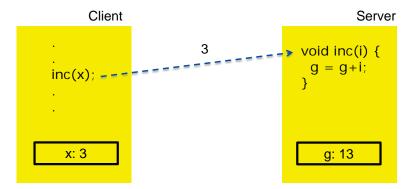
RPC: Remote Procedure Call



KTH

31

RPC: Remote Procedure Call



ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

(KTH)	RPC: Remote Procedure C	Call		KTH	Open Network Computing (ONC) RPC (SunRPC)
	Client ; ; ; ; ; a: {1,2,3,4}	<pre>Server void inc(int[] h) { g = g+h[2]; h[2] = g; } g: 13</pre>			 targeting intranet, file servers etc at-least-once call semantics procedures described in Interface Definition Language (IDL) XDR (eXternal Data Representation) specifies message structure used UDP as transport protocol (TCP also available)
	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION		33		ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



Java RMI (Remote Method Invocation)

- similar to RPC but:
 - we now invoke methods of remote objects
 - at-most-once semantics
- Objects can be passed as arguments, how should this be done?
 - by value
 - by reference



Java RMI

We can do either:

A *remote object* is passed as a reference (*by reference*) i.e. it remains as at the original place where it was created.

A *serializable object* is passed as a copy (*by value*) i.e. the object is duplicated.

34



Finding the procedure/object

How do we locate a remote procedure/object/process?

Network address that specifies the location or..

a known "binder" process that keeps track of registered resources.

Remote invocation design decisions

- failure handling: maybe / at-most-once / at-least-once
- call-by-value / call-by-reference

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

- message specification and encoding
- specification of resource
- procedure binder

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

37

38



Examples

- SunRPC: call-by-value, at-least-once, IDL, XDR, binder
- JavaRMI: call-by-value/reference, at-most-once, interface, JRMP (Java Remote Method Protocol), rmiregistry
- Erlang: message passing, maybe, no, ETF (External Term Format), local registry only
- CORBA (Common Object Request Broker Architecture): call-by-reference, IDL, ORB (Object Request Broker), tnameserv
- Web Services: WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration)



Summary

Implementations of remote invocations: procedures, methods, messages to processes,

have fundamental problems that needs to be solved.

Try to see similarities between different implementations.

When they differ, is it fundamentally different or just implementation details.