Parallel Computing Strategies and Implications

Multi-process vs. multi-threaded



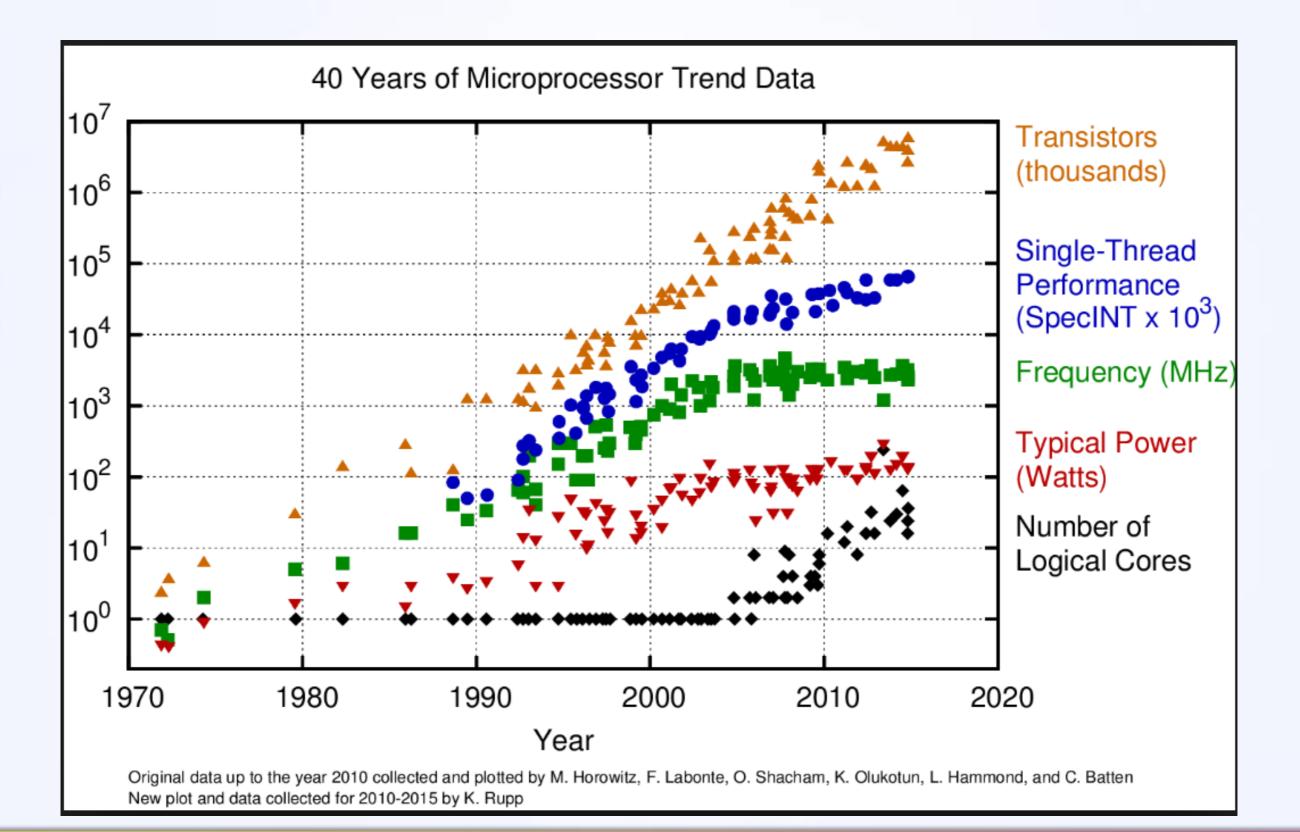
IncrediBuild BEYOND ACCELERATION

Dori Exterman CTO IncrediBuild



Moore's Law Going Multi-Core





Moore's Law Going Multi-Core



Increase in clock speed, allowing software to automatically run faster, no longer oblige to Moore's law

Chip manufacturers keep Moore's law by **increasing the number of CPUs**

Software has to be written to take advantage of all the multi-processors.

Instead of doing everything in a linear fashion, programmers need to **ensure applications are designed for parallel tasking**.

Choosing the Right Parallel Compute Architecture is Important





Product's complexity

Multi-threaded: Pros



Easy and fast to communicate and share data between threads (especially read-only data).

Easy to communicate with parent process.

Beneficial for **large datasets** that can't be divided to sub sets.

Supported by many libraries.

Multi-threaded: Cons



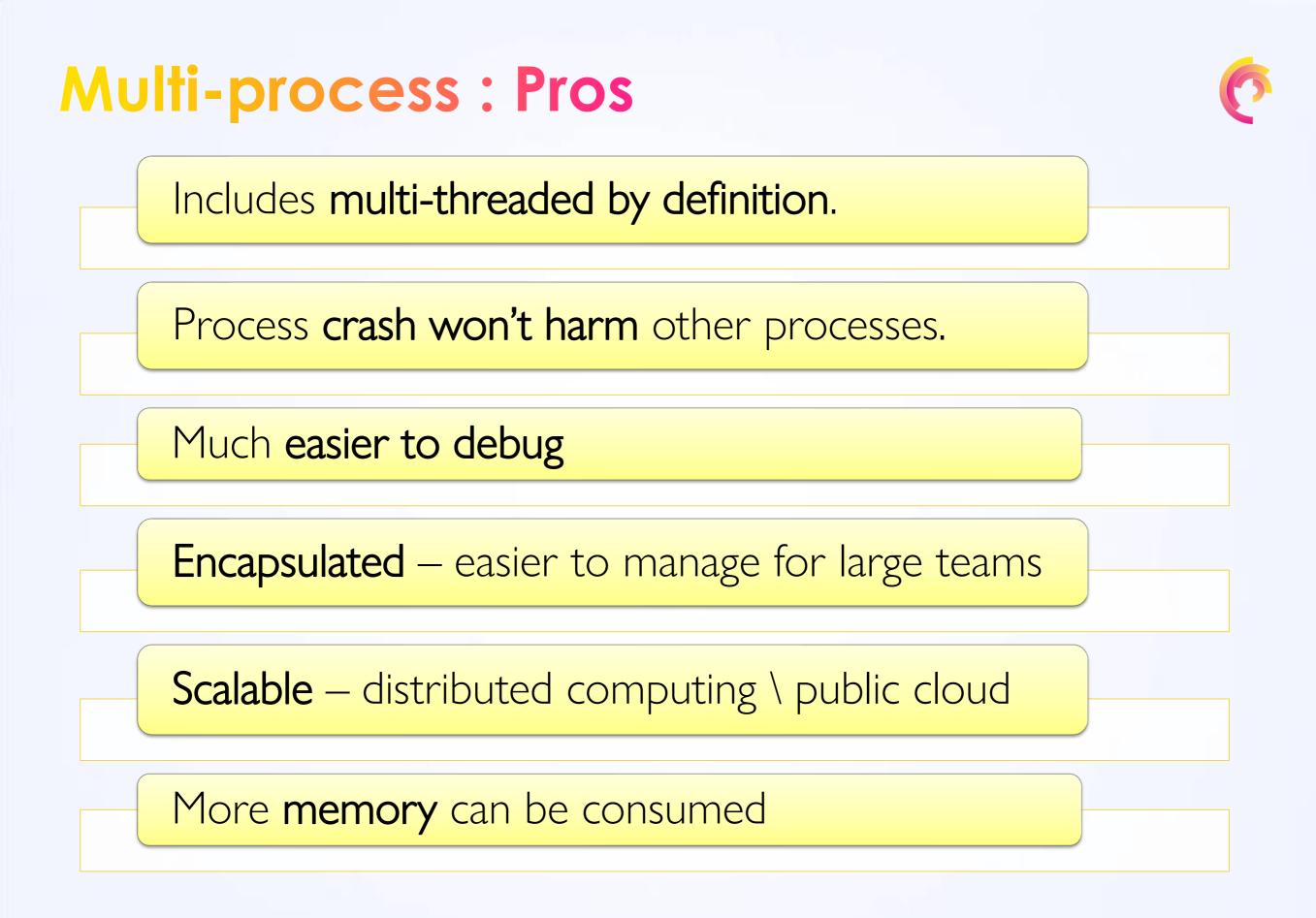
One thread **crashes** \rightarrow the entire process crashes

Multi-threaded apps are hard to debug

Writing to shared data requires (difficult to maintain) locks

Too many threads \rightarrow too much time in **context-switching**.

Not scalable to other machines / public cloud



Multi-process: Cons



Custom development required in order to communicate between processes

Requires special mechanism to share memory data between processes

Limited "process-safe" libraries

Multi-threaded vs. Multi-core: List of Considerations	C
Parallel parts need to sync data?	
Data sets size	
Ease of development	
Scaling - Compute time and throughput	

Synchronization/State Maintenance



Easier in multi-threaded

Easier to debug in multi-threaded

In multi-process, will require custom development

Many mechanisms exists both for multi-process and multi-threaded





Pre-made libraries for MT (multi-threaded)

MP - Easier debug/maintenance/code understanding / less bugs will be introduced with new commits

MP - Can be developed and maintained by less experienced developers

MP - Requires developer to maintain encapsulation

MP - Bugs are less complicated to solve when MT is not involved

Throughput & scalability - 1/2



How many tasks do I have to execute in parallel – dozens /millions

How many tasks will my product need to support in the future?

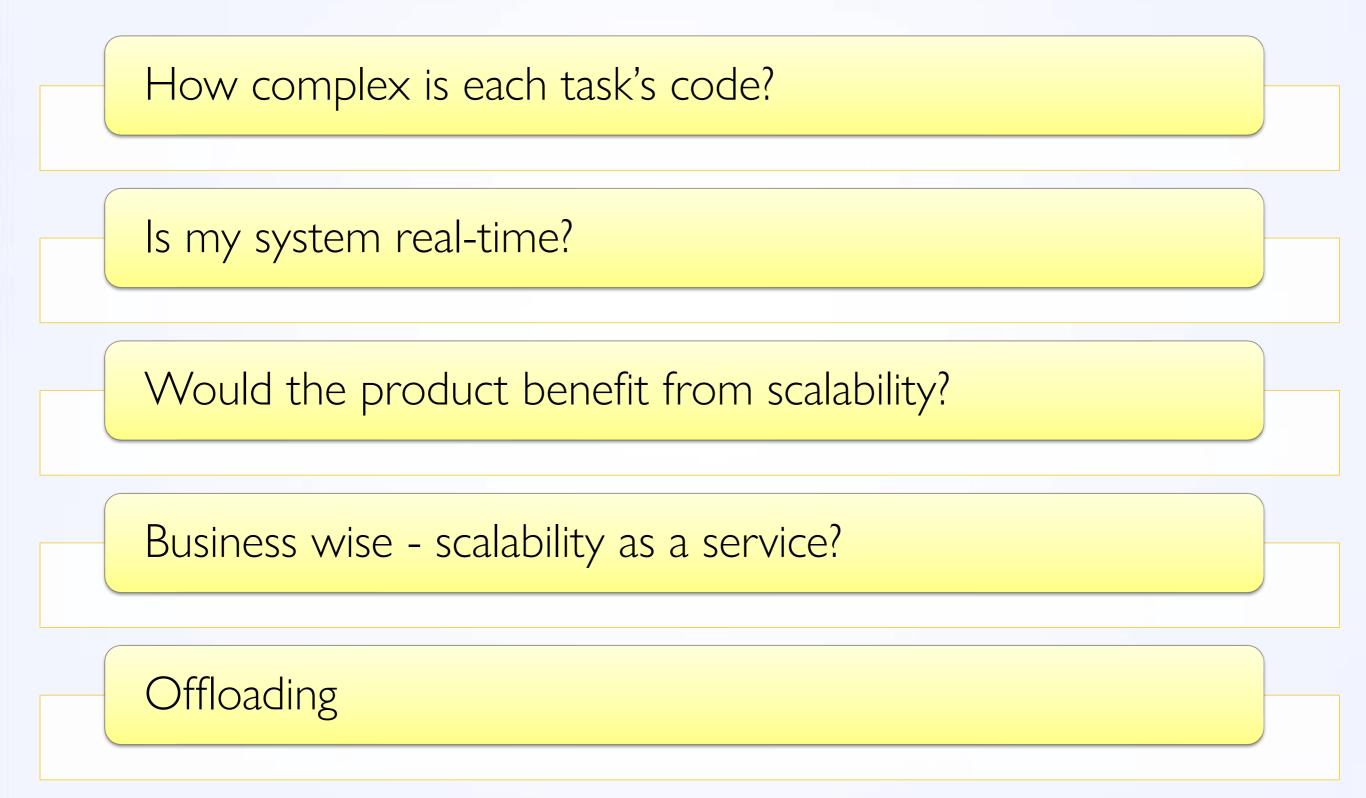
How much time will each task take? Milliseconds or minutes?

How much memory will each task require?

How large is the data that each task requires?

Throughput & scalability - 2/2





Other Scalability Questions



Do my tasks require special hardware?

Will my tasks perform better with specific hardware?

Connection to a database

Common scenarios for using multi-threaded



- Simple scenarios
- Real-time performance
- Long initialization time with short computation time
- Communication, synchs and states
- Only a few highly intensive tasks

Common scenarios for using multi-process



- Multiple, encapsulated modules
- Tasks may require much memory
- Limited synchronization required
- Large application \ large team
- Scaling \ High-throughput
- Tasks are complex and prawned to errors



Parallel Computing The **High-Throughput** Challenge

Let's have a look at a real-life scenario

Maven vs Make build tools

- Popular tools
- Many tasks
- Tasks are atomic
- Minimal communication
- Small dataset
- Scalable



Scenarios – Build system approach: Maven



• Multi-threaded code was only added recently:

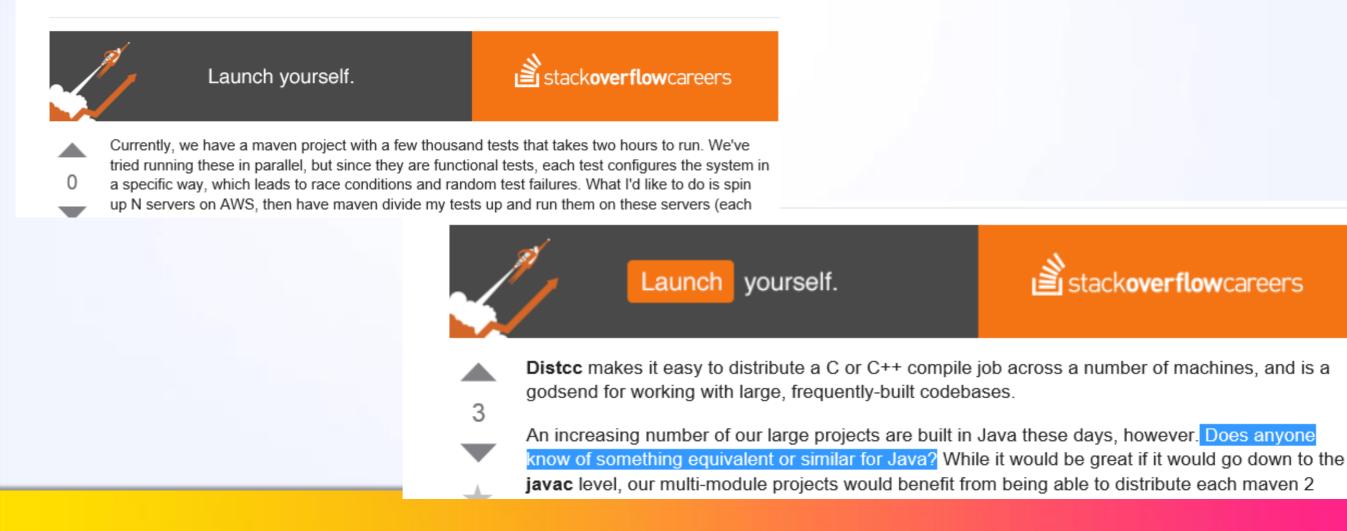
■ Confluence Spaces	
Apache Maven	Pages / Index / Proposals Parallel builds in Maven 3
Pages	I Added by Kristian Rosenvold, last edited by Jason van Zyl on Dec 06, 2013 (view change) show comment
S Blog	Maven 3.x has the capability to perform parallel builds. The command is as follows:
Space tools	mvn -T 4 clean install # Builds with 4 threads mvn -T 1C clean install # 1 thread per cpu core
a Parallel builds in Maven 3	mvn -T 1.5C clean install # 1.5 thread per cpu core This build-mode analyzes your project's dependency graph and schedules modules that can be built

Scenarios – build system approach: Maven



- Not Scalable: Users are asking for it, but it is still missing multi-process invocation.
- Projects are building many 3rd parties.

Distributed build with maven?



Single Machine, Single Core



🖶 Bu	ild Monitor - Build terminated at user requ	est
File	View Dev Help	
		nal', dass 'wrap', initiated by Xoreax Grid Engine Console 5.0 (build 1496)
Comma	andline: xgconsole /command="CommandsBatch	
雷	Agents	· ·
	System Graphs	
		2:55:18.000
a a a a a a a a a a a a a a a a a a a		Distribution
₽		Utilized CPUs 1 CPUs Utilized Processing Power 3.2 GHz
	Vm-retalix	Ready Tasks 1,463 Tasks
	Hex	. StoreSer Stor Contract.ARTS.Utils C StoreServi St St St St St St
?		
	0 CPUs, 0.0 GHz	0:00 2:55:00 3:00:00 3:05:00 3:10:00 3:15:00 3:15:00
	🖄 📑 📴 🖳 😪 🖳	📀 🗏 🔄 🚿 💵 💽 🛠 💁 🕰 💾 💵 🖭

Single Machine – Multi-Core



8 cores, 15:45 minutes

	stem warning(s), 0 error(s), 284 warning(s)					
ew Dev Help	ging level 'minimal', class 'vc14DE', initiated by IncrediB	uild Viewal Studie 2005 (20)	00/2010/2012/2012 Add to 7 1 00 0	uild 1650)		
rom version 7.1 RC (build 1650), log	ging level minimal, class VC14DE, initiated by Incredib	ulid visual Studio 2005/200	08/2010/2012/2013 Add-In 7.1 RC (b	uila 1650)		
Agents						^
System Graphs						
Pc-victor2 (Core #1)	moc_widg	link		moc moc	qdesigner_pch.h	appfontdialog m
Pc-victor2 (Core #2)	moc_stringlisteditor					ass q qdesig
Pc-victor2 (Core #3)	c moc_ta moc_tre					fontpanel
Pc-victor2 (Core #4)	mo moc_tablewidget qrc					main qdesigner_actions
Pc-victor2 (Core #5)	q					mainwindow qttoolbardialo
Pc-victor2 (Core #6)	widget moc_treewidg					new qd qdesigner_work
Pc-victor2 (Core #7)	m moc_textedit_t					pref gdesi saveformaste
Pc-victor2 (Core #8)	moc mo moc					qdesigner q ver
						quesque. qui reini
0 CPUs, 0.0 GHz	:29 15:30 15:31	15:32	15:33 15:34	15:35 15:36	15:37	15:38 15:39

Make



- Multi-process (same as other modern build tools)
- Better memory usage
- Caching add-ons (CCache)
- Scalable 3rd party solutions (DistCC, IceCC, IncrediBuild)

QT on Single Machine – Multi-Core



8 cores, 15:45 minutes

ew Dev Help								
	ging level 'minimal', class 'vc14DE', initiated by IncrediBu	ild Visual Studio 2005/	2008/2010/2012/2013 Add-In 7	7.1 RC (build 1650)				
								e , e
Agents								1
System Graphs								
Pc-victor2 (Core #1)	moc_widg	link			moc moc qdesig	jner_pch.h	appfontdialog m.	
Pc-victor2 (Core #2)	moc_stringlisteditor						ass q qdesig	
Pc-victor2 (Core #3)	c moc_ta moc_tre						fontpanel	
Pc-victor2 (Core #4)	mo moc_tablewidget qrc						main qdesigner_actions	
² c-victor2 (Core #5)	q						mainwindow qttoolb	ardialo
Pc-victor2 (Core #6)	widget moc_treewidg						new qd qdesigne	er_work
Pc-victor2 (Core #7)	m moc_textedit_t						pref gdesi savefo	ormaste
Pc-victor2 (Core #8)	moc mo moc						qdesigner q ver	
0 CPUs, 0.0 GHz	:29 15:30 15:31	15:32	15:33 15:34	15:35	15:36	15:37	15:38 15:39	· >

QT on Single Machine – Multi-Core (predicted)



8 cores, 11 minutes

BI BI	uild Monitor - Projects - 1 build system warn	ing(s), 0 error(s), 284 warning(s)
File	View Dev Help	
Build is	s from version 7.1 RC (build 1650), logging level 'm	inimal', class 'vc14DE', initiated by IncrediBuild Visual Studio 2005/2008/2010/2012/2013 Add-In 7.1 RC (build 1650)
8	Agents	
₽ ₽ ₽	System Graphs	10:09.259 Distribution Utilized Processing Power 0.0 GHz Ready Tasks 128 Tasks Active Tasks 9 Tasks
	Pc-victor2 (Core #1)	nk moc qdesignera q3listvie q3widge m m m moc pathdeform moc_pat lib item item
9	Pc-victor2 (Core #2)	qaxwidgett qdeclar q3ico q3texte qt3su moc mo pho moc_composition formwindow ob q
	Pc-victor2 (Core #3)	gaxwidg moc_q g3m g3t g3wizar moc moc_g3 see pathstroke b f i listwid
	Pc-victor2 (Core #4)	qaxwid m q3istb q3to qdes moc mo videoplay xform
	Pc-victor2 (Core #5)	qaxtypes mo q3is q3ta q3wi mo mo mo moc composition moc_pathstroke designerp designerp
	Pc-victor2 (Core #6)	moc moc_q q3main q3wi m mo m moc plugin co for it ii prope
	Pc-victor2 (Core #7)	moc m q3is q3tabl q3wiz m moc volu gradients moc_xf conn for l o
	Pc-victor2 (Core #8)	qax m q3toolb moc m mo moc_v moc_gradients for m qte
Done	0 CPUs, 0.0 GHz	10:10 10:15 10:20 10:25 10:30 10:35 10:40 V
	- 2 🚔 🔯 🕾 🛸 🖬	

QT on multiple machines, ~200 cores



130 cores, 1:40 minutes

View Dev Help	
from version 7.1 RC (build 1650), loggi	ng level 'minimal', class 'vc14DE', initiated by IncrediBuild Visual Studio 2005/2008/2010/2012/2013 Add-In 7.1 RC (build 1650)
Agents	
System Graphs	1:06.379 Distribution Utilized Processing Power 373.7 GHz
Pc-victor2 (Core #1)	Ready Tasks 139 Tasks Active Tasks 131 Tasks
Pc-victor2 (Core #2)	
Pc-victor2 (Core #3)	
Pc-victor2 (Core #4)	
Pc-victor2 (Core #5)	
Pc-victor2 (Core #6)	
Pc-victor2 (Core #7)	
Pc-victor2 (Core #8)	
Pc-victor (Core #1)	qaxbindable moc domtool qscri formbuilder chip phon main abstra main qde qdeclarative qd
Hermes (Core #1)	qaxfactory main moc_qs mainwindow q3iconview brushmanag main qdeclarati moc_qd
Hermes (Core #2)	qaxmain moc_b abstractvideor moc g3iconvie brushpro qappli qdeclarativeutilmodule
Hermes (Core #3)	qaxserver moc lib_pch.h abs phr spect anim q3listbox buddyeditor qapplic qdedarati moc_qd
Hermes (Core #4)	qaxserverbase custom audiooutput messageeditor boat qaxtypes composition moc qdeclarativeanchors
Hermes (Core #5)	qaxserverdll moc widgetinfo backend mo mo gradients moc qdeclarativeanimatedimage
Hermes (Core #6)	qaxserver moc qau ambientproperties messagee main qaxwidg pathde moc do moc qdedarativeanimation
Hermes (Core #7)	qaxtypes moc_q domb moc_with a g3listview moc qdedarat moc moc moc
Hermes (Core #8)	gglpixelbuffer qaudioo controlinfo phraseview graphic moc seekslid pathstroke moc qdeclarativeb moc_q
Host-12 (Core #1)	tz moc_qglen qim qim docu abs a main moc videop plugin moc qdeclar qme m
Host-12 (Core #2)	d m pa invokeme a qdesign mainwind q3listvi xform moc qdeclarativebinding
0 CPUs, 0.0 GHz	1:00 1:05 1:10 1:15 1:20 1:25 1:30

QT on multiple machines, multi-core



130 cores, 1:40 minutes

View Dev Help							
	ild 1650), logging level 'minimal', class 'vc14Df	, initiated by I	ncrediBuild Visual Studio 200	5/2008/2010/2012/2013 Add-In	7.1 RC (build 1650)		
Agents							
Host-12 (Core #3)			Boo qs	q	q mo	gdec	
Host-12 (Core #4)			Pro qsc		uic m		
Host-12 (Core #5)		p	S	q q			
Host-12 (Core #6)			Sc po	q		qde	
Host-12 (Core #7)			Byteco	q ql	Market	q	
Host-12 (Core #8)		qc	Call qs	qd	gl	qde	
Host-13 (Core #1)	q		CallFr	qlist	q qa g	qd	
Host-13 (Core #2)			CodeBl	q q	qgl		
Host-13 (Core #3)			C ts	q	q	q	
Host-13 (Core #4)	q		S	q	c		
Host-13 (Core #5)			Colle	qli	sp c	q	
Host-13 (Core #6)			Co	qgra	S		
Host-13 (Core #7)	q	q	Com dit				
Host-13 (Core #8)	q		Con q doc	qquerytransformparser		q	
Host-14 (Core #1)	Host-13	qdir	R		qgl ef	qd	
Host-14 (Core #2)	Windows 7 Professional (6.1.7601) Service Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz	Pack 1	Dat	qc	qpaint		
Host-14 (Core #3)	3,392 MHz (8 Cores) 16,270 MB RAM		Da		iod gl d		
Host-14 (Core #4)	1,000 Mbps Network (192. 168. 10. 73)		Dat ht.		Cus me		
Host-14 (Core #5)			Da	qc	qpix me	qd	
Host-14 (Core #6)			Date	qc	de	. qdec	
Host-14 (Core #7)	q	q	Deb	qcom	qg b b		
Host-14 (Core #8)	ql		Pa	qfo	abs d	qde	
Host-15 (Core #1)			Debu	qc q	···		
0 CPUs,	0.0 GHz 0:10	0:	20 0:30	0:40 0:50	1:00 1:10 1:20	1:30	1:40
<							>



Real Life Scenarios



Common Multi-Threaded Scenarios



Large in-memory read \ write meshes:

- CAD
- Al (connections between elements)
- Weather forecasting
- Genetic algorithms

Needs all the data to be in-memory and accessible for all tasks.



Real time performance needs:

- Financial transactions
- Defense systems
- Gaming

Latency and initialization time are very important.



Applications which most of the calculation (business logic) is **Database related** (stored procedures, triggers, OLAP)

CRM, ERP, SAP





Fast sequential processing:

- CnC
- Manufacturing line
- Dependent calculations

Next calculation depends on the previous one.



Common Multi-Process Scenarios



Large **independent dataset**: when we don't need all data in-memory at the same time. Low dependencies between data entities.

- Simulations (transient analysis) Ansys, PSCad
- Financial derivatives Morgan Stanley
- Rendering



Many calculations with small business units. Solution: batch tasks

- Compilations
- Testing (unit tests, API tests, stress tests)
- Code analysis
- Asset builds
- Math calculations (Banking, Insurance, Diamond industry).



GPU – applications that scale to use many GPUs

- CUDA, OpenCL
- Rendering
- Password cracking



Output and computation streaming:

NVIDIA Shield, Onlive Service, Netflix, Waze



The business aspect



APPS

NE VITULE CONNECTION

CELI

DOWNLOAD vieture

HOSTING

ION INTERNET

WED MEDIA was ID

IENT MAIL DATA SHARE MOBILE SOLUTIONINT INSTALL DEVICE BUSINESS VIRTUAL ELECTRONIC COMMUNITY HOST SERVER HOD MOBILITY RESOURCE COMMUNITY HOST SERVER HOD DEVICE SHITTAL WEB

MINERAL/

AEDIA S

FIREWALL MAIL

ERVICE CONNECT

SERVERADAPTERMOBILITY

APPS CALL

TECHNOLOG

- Scalability
- Offloading
- Full cloud solution
- SAAS



Summary

Summary - Writing Scalable Apps:



Architectural considerations:

- Can my execution be broken to many independent tasks?
- Can the work be divided over more than a single host?
- Do I have large datasets I need to work with?
- How many relations are in my dataset?
- How much time would it take my customers to execute a large scenario – do I have large scenarios?

Summary - Writing Scalable app



Technical considerations:

- Do I have any special communication and synchronization requirements?
- Dev Complexity What risks can we have in terms of race-conditions, timing issues, sharing violations – does it justify multi-threading programming?
- Would I like to scale processing performance by using the cloud or cluster?

Distributed Computing



Some of the known infrastructures to using multicore/multi-machine solutions:

- MapReduce (Google dedicated grid)
- Hadoop (Open source dedicated grid)
- Univa Grid Engine(Propriety dedicated grid)
- IncrediBuild (Propriety ad-hoc \ process virtualization grid)







Dori Exterman, IncrediBuild CTO dori@incredibuild.com

IncrediBuild now free in Visual Studio 2017 www.incredibuild.com



Thank you.

